

Элементарная комбинаторика

1 Принцип Дирихле

1.1. Начнем мы данную главу с изложения очень простого принципа, который, однако, является весьма эффективным способом решения многих комбинаторных задач — принципа Дирихле (или в английском варианте — Pigeon-Hole Principle).

1.1.1. В своей простейшей формулировке он гласит следующее:

Утверждение 1.1. *Если в n ящиков положить $k > n$ предметов (в n клеток посадить $k > n$ голубей), то хотя бы в одном ящике будут лежать по крайней мере два предмета (будут сидеть по крайней мере два голубя).*

Доказательство. Несмотря на всю очевидность данного утверждения, все же дадим его формальное доказательство. Предположим, что утверждение неверно, то есть в каждом ящике находится не более одного предмета. Обозначим через m количество ящиков, в котором ничего не лежит. Очевидно, что $m \geq 0$. Тогда ровно по одному предмету лежит в $(n - m)$ ящиках. Это означает, собственно, что общее количество предметов равно $n - m \leq n < k$, что противоречит условию нашего утверждения. \square

1.1.2. На первый взгляд совершенно непонятно, как такой настолько простой принцип может использоваться при решении каких-либо серьезных задач, ведь он кажется совершенно очевидным. Однако оказывается, что в конкретных задачах, использующих в своих решениях данный принцип, зачастую очень нелегко понять, где предметы, где ящики, и почему предметов больше, чем ящиков. Далее, далеко не всегда по формулировке самой задачи можно догадаться, что для ее решения следует воспользоваться данным принципом. И наконец, данный принцип дает нам неконструктивное доказательство какого-то факта — используя его, мы не можем сказать, в каком конкретно ящике находятся два предмета, мы знаем лишь, что такой ящик существует. Попытка же дать конструктивное доказательство, то есть попытка нахождения данного ящика, часто связана с очень большими трудностями. Тем не менее, данный принцип очень полезен, и с его помощью можно доказывать совершенно неочевидные на первый взгляд результаты.

1.1.3. Приведем вначале простейший пример на применение данного принципа.

Пример 1.2. Доказать, что в наборе из любого $(n + 1)$ -го положительного целого числа найдутся по крайней мере два числа, имеющих один и тот же остаток от деления на n .

Доказательство. Действительно, рассмотрим произвольный набор $\{a_1, a_2, \dots, a_n, a_{n+1}\}$ положительных целых чисел. Поделив их с остатком на n , мы получим набор $\{r_1, r_2, \dots, r_n, r_{n+1}\}$ из $(n + 1)$ -го остатка от этого деления — набор неотрицательных целых чисел, каждое из которых меньше n . Но мы знаем, что имеется только лишь n неотрицательных целых чисел, меньших n . Следовательно, по принципу Дирихле, хотя бы два остатка из полученного набора должны совпадать. \square

1.1.4. Обычно в качестве примера применения принципа Дирихле дается чуть более сложная формулировка того же самого утверждения.

Пример 1.3. Доказать, что в последовательности чисел

7, 77, 777 ...

Доказательство. Действительно, поделим исходный прямоугольник на четыре равные части размерами 3×4 сантиметра. Так как пять точек должны находиться либо внутри, либо на границах этих четырех прямоугольников, то, согласно принципу Дирихле, хотя бы две из них должны лежать либо внутри, либо на границах одного и того же прямоугольника. Осталось заметить, что расстояние между любыми такими точками меньше или равно пяти сантиметрам. \square

1.1.7. В заключение данного пункта приведем еще один интересный пример на принцип Дирихле, который предложили венгерские математики Эрдеши и Секереш.

Пример 1.6. Выделим в множестве $[2n]$ первых $2n$ целых положительных чисел подмножество $S \subset [2n]$ мощности $(n + 1)$. Доказать, что в этом подмножестве существуют хотя бы два числа, одно из которых делит другое.

Решение. Для применения принципа Дирихле нам нужно каким-то образом связать голубей с элементами $(n + 1)$ -элементного подмножества S , а клетки — с элементами какого-то n -элементного множества N . Оказывается, что в рассматриваемом примере нам в качестве множества N нужно выбрать n -элементное подмножество всех нечетных чисел.

Действительно, ключевое для решения данной задачи наблюдение состоит в том, что любое число $s \in S$ может быть представлено в виде

$$s = 2^r \cdot q,$$

где q есть нечетное число, $1 \leq q \leq 2n - 1$. Так как таких чисел s ровно $n + 1$ штук, то, согласно принципу Дирихле, у нас обязательно найдутся среди них два числа с одинаковым нечетным делителем q . Как следствие, одно из этих чисел обязательно делит второе.

Данная задача допускает еще несколько решений. Одно из них базируется на построении дерева с корнем в вершине 1 по следующему правилу. Соединим все простые числа ребрами с единицей. Для любого другого числа соединим его ребром с наибольшим делителем этого числа. Заметим, что листьев в таком дереве не более n — это следует из того, что любое число i в диапазоне от 1 до n является делителем по крайней мере одного числа из диапазона от 1 до $2n$, а именно, числа $2i$. Так как нам задано $n + 1$ число, то хотя бы два из них будут принадлежать ветке, имеющей общий лист, а значит, будут делиться одно на другое.

Еще одно решение является некоторой комбинацией двух предыдущих. Именно, разобьем множество чисел на цепочки

$$1, 2, 4, 8, \dots, \quad 3, 6, 12, 24, \dots, \quad 5, 10, 20, 40, \dots, \quad 7, 14, 28, \dots$$

в каждой из которых любое число делится на предыдущее. Заметим, что любая такая цепочка может начинаться только лишь с нечетного числа — любое четное число $2a$ мы можем поделить на два и получить число a , делящее $2a$. Следовательно, количество таких цепочек совпадает с количеством нечетных чисел и равно n . Тогда какие-то два из $(n + 1)$ -го числа обязательно попадут в одну цепочку, а значит, одно из них будет делиться на второе.

1.2. Часто в задачах нужно определить минимальное количество k предметов, гарантированно обеспечивающих наличие хотя бы r предметов в одном из n ящиков.

1.2.1. Очевидно, что в случае $r = 2$ мы должны взять $n + 1$ предмет — действительно, согласно принципу Дирихле, в этом случае хотя бы в одном ящике гарантированно окажется по крайней мере два предмета.

В более общем случае нам достаточно взять

$$k = n \cdot (r - 1) + 1 \quad (1)$$

предметов. Действительно, $n \cdot (r - 1)$ предмет мы еще можем распределить по n ящикам так, чтобы в каждом ящике находилось не более $(r - 1)$ -го предмета, а вот в случае $k = n \cdot (r - 1) + 1$ у нас хотя бы в одном ящике гарантированно окажется хотя бы r предметов.

Последнее утверждение часто называют обобщенным принципом Дирихле и формулируют его так. Пусть у нас имеются k предметов, которые мы должны распределить по $n < k$ ящикам. Тогда существует по крайней мере один ящик, в котором содержится не менее чем

$$\lceil k/n \rceil$$

предметов, где $\lceil k/n \rceil$ — ближайшее целое число, большее или равное k/n .

1.2.2. В качестве примера на использование обобщенного принципа Дирихле рассмотрим следующую задачу.

Пример 1.7 (Эрдеш, Секереш). Доказать, что любая последовательность из $n^2 + 1$ целых чисел содержит либо убывающую, либо возрастающую подпоследовательность, состоящую из не менее чем $(n + 1)$ -го числа.

Для доказательства данного утверждения введем количество s_i чисел в самой длинной возрастающей подпоследовательности, начинающейся с a_i . Например, для последовательности чисел

$$\{a_1, a_2, a_3, a_4, a_5\} = \{1, 5, 3, 2, 4\} \quad s_1 = 3, s_2 = 1, s_3 = 2, s_4 = 2, s_5 = 1.$$

Сразу заметим, что если существует такое i , что $s_i \geq (n + 1)$, то процесс завершен. В нашем случае $s_1 = 3 = (2 + 1)$, и поэтому в нашем случае все в порядке — эта последовательность содержит возрастающую подпоследовательность $\{1, 3, 4\}$ длины три. Для подпоследовательности же

$$\{a_1, a_2, a_3, a_4, a_5\} = \{1, 5, 4, 3, 2\} \quad s_1 = 2, s_2 = 1, s_3 = 1, s_4 = 1, s_5 = 1,$$

и такого i не существует. Иными словами, в этом случае все $(n^2 + 1)$ штук s_i принимают значения из диапазона $[1, n]$. Тогда, согласно обобщенному принципу Дирихле, существует по крайней мере

$$\left\lceil \frac{n^2 + 1}{n} \right\rceil = n + 1$$

чисел s_i , значения которых одинаковы и равны s . Все, что нам осталось понять — это то, что соответствующая этим значениям $s_i = s$ подпоследовательность $\{a_i\}$, состоящая из $(n + 1)$ -го числа, является убывающей.

Действительно, предположим, что в этой подпоследовательности существует такая пара индексов (i, j) , что $j > i$ и $a_j > a_i$. Напомним, что по определению числа s_j , возрастающая подпоследовательность чисел, начинающаяся с a_j имеет длину $s_j = s$. Добавляя a_i в начало этой подпоследовательности, мы получим возрастающую последовательность длины $s + 1$, начинающуюся с a_i . Это, в свою очередь, противоречит тому, что наибольшая числовая возрастающая последовательность, начинающаяся с a_i , имеет длину, равную $s_i = s$.

2 Основные правила перечислительной комбинаторики

2.1. Перейдем теперь к описанию простейших фактов, лежащих в основе перечислительной комбинаторики. Начнем с очень краткого напоминания основных понятий теории множеств.

Определение 2.1. Множеством $X = \{x_1, x_2, \dots, x_n\}$ называется совокупность различных объектов $x_i, i = 1, \dots, n$, объединенных по некоторому признаку.

В качестве характерного примера можно рассмотреть, например, множество X всех студентов, находящихся в данной аудитории. Действительно, все студенты различимы, отличны друг от друга и объединены по признаку “собрались в данной аудитории”.

Определение 2.2. Мощностью $|X|$ множества X называется количество элементов в нем. Как правило, мы будем рассматривать конечные множества, в которых $|X| = n, n \in \mathbb{N}$, и называть их n -множествами.

2.1.1. Основные операции над множествами — это объединение $A \cup B$ (рис.1,а), пересечение $A \cap B$ (рис.1,б), разность $A \setminus B$ (рис.2,а) и симметрическая разность (рис.2,б) двух множеств A и B . В случае, если множество A является подмножеством некоторого более широкого множества X , удобно также рассматривать операцию дополнения $A' := X \setminus A$ множества A (рис.2,с).



Рис. 1

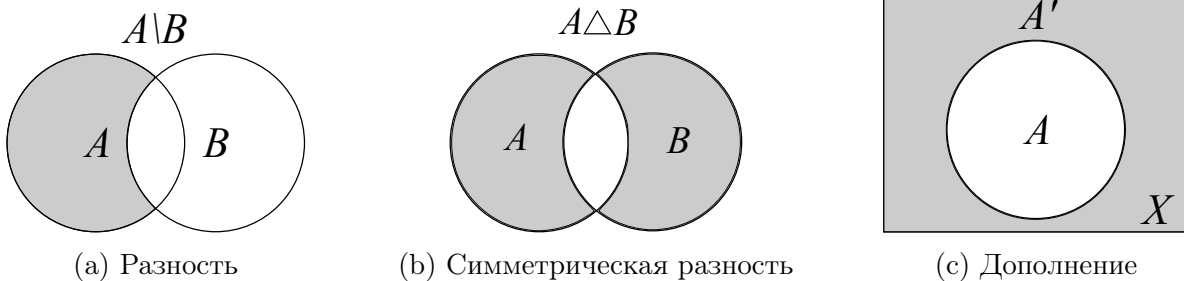


Рис. 2

Свойства операций над множествами удобно изучать графически, с использованием так называемых *диаграмм Эйлера-Венна* (смотри рисунки 1 и 2). Например, с их помощью достаточно просто проиллюстрировать справедливость законов де Моргана

$$A' \cap B' = (A \cup B)', \quad A' \cup B' = (A \cap B)' \quad (2)$$

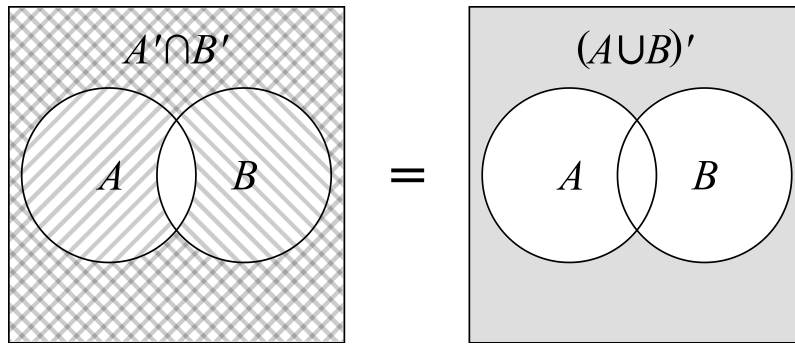


Рис. 3: Графическое доказательство закона де Моргана

(смотри рис.3).

2.1.2. В дальнейшем мы достаточно часто будем использовать понятие покрытия множества X семейством $\{X_1, X_2, \dots, X_k\}$ множеств, а также всевозможные частные случаи этого понятия.

Определение 2.3. Семейство множеств $\{X_1, X_2, \dots, X_k\}$ называется *покрытием* множества X , если их объединение дает нам все множество X :

$$X_1 \cup X_2 \cup \dots \cup X_k = X.$$

Важным частным случаем покрытия является понятие разбиения множества.

Определение 2.4. Говорят, что семейство множеств $\{X_1, X_2, \dots, X_k\}$ образует *разбиение* множества X , если

1. множества $X_i \neq \emptyset$, $i = 1, \dots, n$;
2. $X_i \cap X_j = \emptyset \forall i \neq j$;
3. $X_1 \cup X_2 \cup \dots \cup X_k = X$.

Элементы X_i этого семейства называются *блоками* разбиения.

В качестве характерного примера можно рассмотреть разбиение студентов данного курса на группы. Студенческие группы являются при этом блоками данного разбиения.

Если по каким-то причинам оказывается важным порядок следования блоков, то говорят об *упорядоченном разбиении* (X_1, X_2, \dots, X_k) множества X . Например, если мы выводим группы на сцену для вручения им дипломов, то важен порядок, в котором они туда выходят. Следовательно, в данном случае мы получаем упорядоченное разбиение студентов данного курса.

Наконец, еще одним частным случаем покрытия X семейством множеств $\{X_1, X_2, \dots, X_k\}$ является понятие *разделения* множества X . Разделение есть аналог упорядоченного разбиения, в котором допускаются пустые блоки. Точное определение таково:

Определение 2.5. Разделением множества X называется упорядоченная последовательность (X_1, X_2, \dots, X_k) возможно пустых, попарно непересекающихся множеств, объединение которых дает все множество X .

2.1.3. Еще одной часто используемой в комбинаторике операцией над множествами является операция декартова произведения множеств.

Определение 2.6. Декартовым произведением множеств A и B называется множество

$$A \times B := \{(a, b) \mid a \in A, b \in B\}$$

всех упорядоченных пар (a, b) , таких, что $a \in A, b \in B$.

В качестве простейшего примера декартова произведения множеств обычно приводят шахматную доску. Любая клетка шахматной доски имеет координаты “буква-цифра”, например, $e5$ или $h4$. Иными словами, координаты клеток шахматной доски являются элементами декартова произведения множеств $A = \{a, b, \dots, h\}$ и $B = \{1, 2, \dots, 8\}$.

В частном случае множества A и B могут совпадать. В этом случае декартово произведение $A \times A$ обозначается через $A^{(2)}$.

Определение 2.7. Декартовым произведением k множеств X_1, X_2, \dots, X_k называется множество

$$X_1 \times X_2 \times \dots \times X_k := \{(x_1, x_2, \dots, x_k) \mid x_i \in X_i, \forall i = 1, \dots, k\}$$

всевозможных упорядоченных k -элементных последовательностей вида (x_1, x_2, \dots, x_k) .

В частном случае $X_1 = X_2 = \dots = X_k = X$ имеем декартову степень $X \times X \times \dots \times X =: X^{(k)}$.

2.1.4. Любой элемент $X^{(k)}$ есть упорядоченный набор из k элементов множества X , в котором некоторые элементы могут повторяться. Если же в таком k -множестве порядок элементов не важен, говорят о k -мультимножестве над множеством X . Формальное определение k -мультимножества таково.

Определение 2.8. k -мультимножеством над n -элементным множеством X называется пара (X, φ) , где $\varphi: X \rightarrow \mathbb{Z}_+$ есть функция, сопоставляющая любому элементу $x \in X$ количество $\varphi(x)$ его вхождений в k -мультимножество.

Любую функцию φ такого рода можно определить с помощью множества Ξ упорядоченных пар

$$\Xi := \{(x, \varphi(x)) \mid x \in X\}.$$

Поэтому, например, 3-мультимножество над множеством $X = \{x, y\}$, состоящее из двух элементов x и одного элемента y , можно формально записывать в виде $(X, \{(x, 2), (y, 1)\})$. Однако чаще вместо такой формальной записи используют более наглядную форму записи вида $\{x, x, y\}$.

Самый простой и понятный пример мультимножества — это монеты в кошельке. В этом примере в качестве множества X выступает множество из девяти монет разного достоинства:

$$X = \{1 \text{ копейка}, 5 \text{ копеек}, 10 \text{ копеек}, 50 \text{ копеек}, 1 \text{ рубль}, 2 \text{ рубля}, 5 \text{ рублей}, 10 \text{ рублей}\}.$$

Любой набор из этих монет в количестве k штук образует k -мультимножество над множеством X .

2.1.5. Теория множеств как раздел математики создавалась значительно позже комбинаторики. Поэтому некоторые наиболее важные понятия теории множеств исторически получили в комбинаторике свои, специфические названия. Именно,

1. k -сочетанием без повторений называется любое k -элементное подмножество n -элементного множества;
2. k -сочетанием с повторениями называется любое k -мультимножество над n -множеством;
3. k -перестановкой без повторений называется упорядоченное k -подмножество n -элементного множества;
4. k -перестановкой с повторениями называется любой элемент декартовой степени $X^{(k)}$.

2.2. Теперь перейдем к двум самым простым, но в то же время достаточно важным правилам перечислительной комбинаторики — правилу суммы и правилу произведения.

2.2.1. Начнем с простейшего примера: пусть на одном блюде лежат три яблока, а на втором — две груши; сколькими способами можно выбрать один фрукт? Ответ очевиден: пятью способами.

Обобщающее этот пример простейшее *правило суммы* можно сформулировать так: если некоторый объект из множества A можно выбрать k способами, и, вне зависимости от выбора этого объекта, можно n способами выбрать некоторый элемент множества B , то выбор объекта из множества A или из множества B можно осуществить $k + n$ способами.

Очевидна переформулировка этого правила на языке теории множеств: пусть пересечение двух множеств A и B пусто; тогда

$$|A \cup B| = |A| + |B|.$$

В частности, если $A \subset X$ и A' — дополнение множества A , то

$$|A| + |A'| = |X|. \quad (3)$$

В более общем случае, рассматривая произвольное разбиение множества X на блоки, имеем равенство вида

$$|X| = |X_1| + |X_2| + \dots + |X_k|,$$

которое также называется *правилом суммы* в комбинаторике.

2.2.2. Под *правилом произведения* в комбинаторике понимается равенство

$$|X_1 \times X_2 \times \dots \times X_k| = |X_1| \cdot |X_2| \cdot \dots \cdot |X_k|.$$

Приведем простейший пример на применение этого правила в комбинаторике. Пусть в аудитории находятся 32 студента одной группы, 24 студента другой группы и 17 студентов третьей группы. В этом случае тройку, состоящую из представителей каждой группы, можно выбрать $32 \cdot 24 \cdot 17$ способами.

2.3. Наряду с *правилом суммы*, в элементарной комбинаторике также достаточно часто используется и несложное его обобщение — так называемый *принцип включения-исключения*. Если *правило суммы* связано с разбиением множества X , то *принцип включения-исключения* связан с некоторым произвольным покрытием этого n -множества семейством $\{X_1, X_2, \dots, X_k\}$. Сформулируем его для самого простого случая двух множеств.

2.3.1. Рассмотрим два конечных множества A и B , пересечение которых может быть и непусто. Тогда количество элементов в объединении этих множеств, очевидно, равно

$$|A \cup B| = |A| + |B| - |A \cap B|. \quad (4)$$

Действительно, когда мы считаем количество $|A|$ элементов в множестве A и складываем его с количеством $|B|$ элементов в множестве B , мы любой элемент, принадлежащий как множеству A , так и множеству B , считаем дважды. Чтобы этот избыток убрать, нам нужно один раз вычесть количество элементов, содержащихся в пересечении этих двух множеств.

Равенство (4) можно называть обобщенным правилом суммы — оно обобщает правило суммы на случай, когда пересечение двух множеств не пусто.

2.3.2. Предположим теперь, что A и B являются подмножествами некоторого более широкого множества X . В этом случае у множества $A \subset X$ и множества $B \subset X$ имеются дополнения к ним — множества A' и B' , причем $A \cup A' = B \cup B' = X$.

Рассмотрим теперь пересечение $A' \cap B'$ дополнений множеств A и B . Согласно одной из теорем де Моргана (2), $A' \cap B' = (A \cup B)'$. Следовательно, количество элементов в этом пересечении с учетом равенства (3) и обобщенного правила суммы (4) можно сосчитать так:

$$|A' \cap B'| = |(A \cup B)'| = |X| - |A \cup B| = |X| - |A| - |B| + |A \cap B|.$$

Равенство

$$|A' \cap B'| = |X| - |A| - |B| + |A \cap B|, \quad (5)$$

и называется в комбинаторике принципом включения-исключения.

2.3.3. Приведем простейший пример его использования. Пусть в аудитории находятся 30 человек, 20 человек из которых знают английский, 12 — французский, а 6 человек знают оба языка. Сколько человек не знает ни один из этих иностранных языков? Ответ, согласно принципу включения-исключения (5), следующий:

$$N = 30 - 20 - 12 + 6 = 4.$$

2.3.4. Несложно обобщить равенства (4) и (5) на случай большего количества множеств. Например, для трех множеств A , B , C соответствующие формулы выглядят так:

а) обобщенное правило суммы:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|; \quad (6)$$

б) принцип включения-исключения:

$$|A' \cap B' \cap C'| = |X| - |A| - |B| - |C| + |A \cap B| + |A \cap C| + |B \cap C| - |A \cap B \cap C|. \quad (7)$$

Действительно, рассмотрим, к примеру, левую часть равенства (6). Она подсчитывает количество элементов, принадлежащих объединению трех множеств (смотри рис.4). Если элемент x_1 , принадлежащий этому объединению, содержится в множестве A , но не содержится в множествах B и C , то он один раз подсчитывается в правой части равенства (6) (слагаемое $|A|$, отвечающее зеленой подобласти на рис.4). Если элемент x_2 принадлежит множествам A и B , но не принадлежит C (красная подобласть на рис.4), то в правой части (6) этот элемент входит в слагаемые $|A|$, $|B|$ и $-|A \cap B|$, то есть также подсчитывается ровно один раз. Наконец, если x_3 принадлежит пересечению всех трех множеств (зеленая подобласть на рис.4), то за этот элемент отвечают все слагаемые в правой части (6). Так как четыре из них входят со знаком плюс, а три — со знаком минус, то этот элемент также считается в правой части (6) лишь однажды.

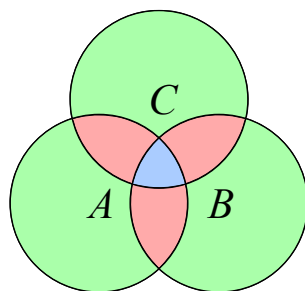


Рис. 4: Диаграмма Эйлера-Венна для трёх множеств

3 Подсчет k -сочетаний из n элементов. Биномиальные коэффициенты.

3.1. Основная задача данного параграфа состоит в подсчете количества всех k -сочетаний из n элементов. Начнем мы с подсчета количества k -сочетаний из n элементов без повторений.

3.1.1. Число k -сочетаний без повторений известно в литературе под названием биномиальных коэффициентов. Ранее в советской литературе такие числа обозначались через C_n^k . В настоящее время для этих коэффициентов используется обозначение $\binom{n}{k}$ (читается “из n по k ”).

3.1.2. Обычно на вопрос, чему равны биномиальные коэффициенты, вспоминают формулу

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Эта формула не очень удачна с двух точек зрения — с вычислительной и с идейной. С вычислительной точки зрения ее затруднительно использовать для достаточно больших значений n и k . Например, при $n = 38$ и $k = 19$ числитель и знаменатель могут просто не уместиться в диапазон изменения целых чисел для того или иного языка программирования. С идейной же точки зрения эта формула неудобна потому, что она не позволяет обобщить понятие биномиальных коэффициентов на случай целых, вещественных или комплексных значений n . В следующем параграфе мы получим более удобное выражение для этих коэффициентов, допускающее такое обобщение. Здесь же мы с помощью правила суммы выведем рекуррентное соотношение для биномиальных коэффициентов, позволяющее эти биномиальные коэффициенты эффективно вычислять для достаточно больших значений n и k .

3.1.3. Для получения рекуррентного соотношения введем множество Σ_k всех k -элементных подмножеств n -элементного множества X . Например, для $X = \{x_1, x_2, x_3\}$ множество $\Sigma_2 = \{\{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}\}$. Заметим, что биномиальный коэффициент $\binom{n}{k}$ как раз и описывает мощность такого множества. Разобьем теперь множество Σ_k на два блока — блок $\Sigma_k^{(1)}$, k -элементные подмножества которого содержат элемент x_1 , и блок $\Sigma_k^{(2)}$, подмножества которого этот элемент не содержат. Понятно, что это — непустые, непересекающиеся подмножества, объединение которых дает нам все множество Σ_k . Поэтому по правилу суммы мы получаем равенство вида

$$\binom{n}{k} = |\Sigma_k| = |\Sigma_k^{(1)}| + |\Sigma_k^{(2)}|.$$

Осталось выразить через биномиальные коэффициенты количество элементов в каждом из блоков $\Sigma_k^{(1)}$, $\Sigma_k^{(2)}$. А это делается довольно легко.

3.1.4. Действительно, во всех подмножествах первого блока элемент x_1 уже выбран, и нам из оставшегося $(n - 1)$ -элементного множества $X \setminus x_1$ остается выбрать $(k - 1)$ -элементные подмножества. Сделать это можно $\binom{n-1}{k-1}$ способами. Во втором блоке содержатся k -элементные подмножества множества $X \setminus x_1$, состоящего из $(n - 1)$ -го элемента. Их количество, очевидно, равно $\binom{n-1}{k}$. Таким образом окончательно имеем следующее рекуррентное соотношение:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \quad k \geq 1, \quad n \geq 1. \quad (8)$$

3.1.5. Соотношение (8) следует дополнить начальными и граничными условиями. Так как k -элементных подмножеств n -элементного множества в случае $k > n$ не существует, то

$$\binom{n}{k} = 0 \quad \text{при} \quad k > n.$$

Далее, пустое подмножество можно выбрать всегда и только одним способом; поэтому

$$\binom{n}{0} = 1 \quad \forall n \geq 0.$$

Используя эти условия, можно шаг за шагом вычислить коэффициенты $\binom{n}{k}$. Часто их записывают в виде так называемого треугольника Паскаля:

$$\begin{array}{cccccc} & & & & & & 1 & & & & \\ & & & & & & & & & & \\ & & & & & & 1 & & 1 & & \\ & & & & & & 1 & & 2 & & 1 \\ & & & & & & 1 & & 3 & & 3 & & 1 \\ & & & & & & 1 & & 4 & & 6 & & 4 & & 1 \\ & & & & & & & & & & \dots & & & & \end{array}$$

3.1.6. Как видно, треугольник Паскаля симметричен, т.е. $\binom{n}{k} = \binom{n}{n-k}$. Комбинаторное доказательство этого факта очевидно. Действительно, выбирая любое k -элементное множество, мы тем самым однозначно выбираем и дополнение к нему, т.е. $(n - k)$ -элементное множество. Следовательно, количество k -элементных и $(n - k)$ -элементных подмножеств совпадает.

3.2. Наряду с треугольником Паскаля биномиальные коэффициенты допускают и еще одно очень удобное графическое представление — представление на координатной плоскости (n, k)

3.2.1. Данное представление связано со следующей довольно интересной комбинаторной задачей. Рассмотрим плоскость (n, k) и нарисуем на этой плоскости пути, исходящие из начала координат, приходящие в точку с координатами (n, k) , $n \geq 0, k = 0, \dots, n$, и состоящие из диагональных и вертикальных отрезков (рис.5). В самих точках (n, k) отметим количество таких путей, приходящих в эту точку из начала координат. Заметим теперь, что попасть в точку с координатами (n, k) мы можем, пройдя только лишь через точку с координатами $(n - 1, k)$ или через точку с координатами $(n - 1, k - 1)$. С комбинаторной точки зрения это означает, что количество путей $\binom{n}{k}$, приходящих в точку с координатами (n, k) , равняется количеству $\binom{n-1}{k}$ путей, приходящих в точку с координатами $(n - 1, k)$, плюс количество путей $\binom{n-1}{k-1}$, приходящих в точку с координатами $(n - 1, k - 1)$. Но равенство

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

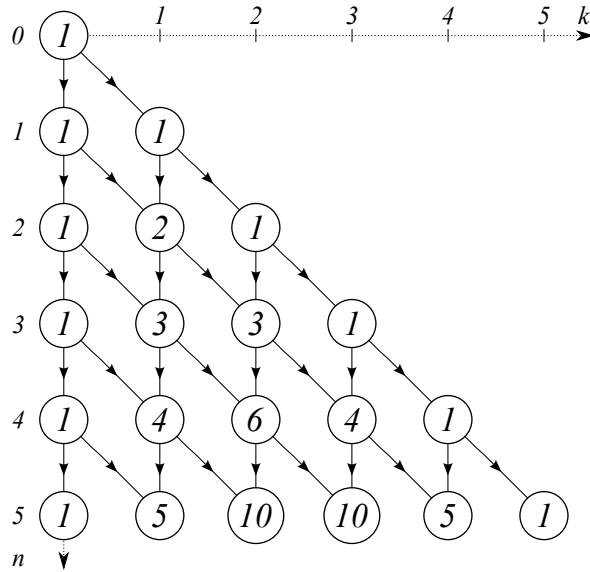


Рис. 5: Графическое представление чисел $\binom{n}{k}$ на координатной плоскости (n, k)

есть основное тождество для биномиальных коэффициентов (8). А это означает, что мы получили новую комбинаторную интерпретацию таких коэффициентов. Именно, числа $\binom{n}{k}$ описывают количество путей, состоящих из диагональных $(1, 1)$ и вертикальных $(1, 0)$ отрезков, выходящих из начала координат — точки $(0, 0)$, и оканчивающихся в точке с координатами (n, k) .

3.2.2. Графическое представление чисел $\binom{n}{k}$ на плоскости (n, k) очень удобно для получения разного рода тождеств с биномиальными коэффициентами. В качестве первого примера зафиксируем какое-то конкретное значение параметра k и просуммируем биномиальные коэффициенты $\binom{m}{k}$ по m от k до некоторого фиксированного значения n . Например, выберем $k = 1, n = 4$. Складывая числа $1, 2, 3, 4$, мы получим число 10 , то есть биномиальный коэффициент, стоящий правее и ниже рассмотренной цепочки биномиальных коэффициентов. Есть подозрение, что данный факт, а именно, равенство вида

$$\binom{k}{k} + \binom{k+1}{k} + \dots + \binom{n}{k} = \binom{n+1}{k+1} \quad (9)$$

выполняется для любых значений параметров n и k .

3.2.3. Для формального доказательства тождества

$$\sum_{m=0}^n \binom{m}{k} = \underbrace{\binom{0}{k} + \binom{1}{k} + \dots + \binom{k-1}{k}}_{=0} + \binom{k}{k} + \binom{k+1}{k} + \dots + \binom{n}{k} = \sum_{m=k}^n \binom{m}{k} = \binom{n+1}{k+1},$$

называемого формулой суммирования биномиальных коэффициентов по верхнему индексу, применим рекуррентное соотношение (8) к коэффициенту $\binom{m+1}{k+1}$:

$$\binom{m+1}{k+1} = \binom{m}{k} + \binom{m}{k+1} \quad \Rightarrow \quad \binom{m}{k} = \binom{m+1}{k+1} - \binom{m}{k+1}.$$

Просуммируем теперь полученное равенство по m от k до n :

$$\begin{aligned} \sum_{m=k}^n \binom{m}{k} &= \binom{n+1}{k+1} + \sum_{m=k}^{n-1} \binom{m+1}{k+1} - \sum_{m=k}^n \binom{m}{k+1} = \\ &= \binom{n+1}{k+1} + \sum_{m=k}^{n-1} \binom{m+1}{k+1} - \sum_{m=k+1}^n \binom{m}{k+1} = \\ &= \binom{n+1}{k+1} + \sum_{m'=k+1}^n \binom{m'}{k+1} - \sum_{m=k+1}^n \binom{m}{k+1} = \binom{n+1}{k+1}. \end{aligned}$$

3.2.4. Комбинаторное доказательство тождества (9) основано на следующем общем подходе: мы разбиваем множество Σ_{k+1} всех $(k+1)$ -элементных подмножеств $(n+1)$ -элементного множества X на блоки, подсчитываем количество элементов в каждом блоке, а затем пользуемся правилом суммы для подсчета числа $|\Sigma_{k+1}| = \binom{n+1}{k+1}$.

Для реализации этого подхода возьмем $(n+1)$ -элементное множество

$$X = \{x_1, x_2, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_{n-1}, x_n, x_{n+1}\}$$

и будем разбивать множество всех $(k+1)$ -элементных подмножеств множества X следующим образом. В первый блок поместим все $(k+1)$ -элементные подмножества, содержащие элемент x_{n+1} . Такие подмножества элемент x_{n+1} гарантированно содержат. Нам из оставшихся элементов x_1, \dots, x_n множества $X \setminus x_{n+1}$ нужно выбрать недостающие k элементов. По определению биномиального коэффициента, сделать мы это можем $\binom{n}{k}$ способами.

Теперь мы рассмотрим все $(k+1)$ -элементные подмножества, которые не содержат элемент x_{n+1} , но обязательно содержат элемент x_n . Для того, чтобы любое такое подмножество сформировать, нам нужно из множества элементов $\{x_1, \dots, x_{n-1}\}$ выбрать недостающие k элементов. Это можно сделать $\binom{n-1}{k}$ количеством способов.

Затем рассмотрим все $(k+1)$ -элементные подмножества, которые содержат в обязательном порядке элемент x_{n-1} и не содержат элементов x_{n+1} и x_n . Эти подмножества получаются выбором недостающих k элементов из множества элементов $\{x_1, \dots, x_{n-2}\}$, и выбрать такие подмножества можно $\binom{n-2}{k}$ способами.

Продолжая далее, мы дойдем когда-то до ситуации, в которой нам нужно выбрать все $(k+1)$ -элементные подмножества, которые содержат элемент x_{k+1} и не содержат элементы со старшими индексами. Так как элемент x_{k+1} у нас уже выбран, то нам остается из множества $\{x_1, \dots, x_k\}$ выбрать k -элементное подмножество. Сделать мы это можем, очевидно, $\binom{k}{k} = 1$ способом.

Складывая теперь количество элементов в каждом блоке, мы и получаем тождество (9).

Пример 3.1. Пусть $n = 4$, $X = \{x_1, x_2, x_3, x_4, x_5\}$, $k = 2$, $k+1 = 3$. Приведем список всех трехэлементных подмножеств этого множества:

$$\begin{aligned} \{x_1, x_2, x_3\}, & \quad \{x_1, x_2, x_4\}, & \quad \{x_1, x_2, x_5\}, & \quad \{x_1, x_3, x_4\}, & \quad \{x_1, x_3, x_5\}, \\ \{x_1, x_4, x_5\}, & \quad \{x_2, x_3, x_4\}, & \quad \{x_2, x_3, x_5\}, & \quad \{x_2, x_4, x_5\}, & \quad \{x_3, x_4, x_5\}. \end{aligned}$$

В первый блок разбиения этого множества подмножеств включим подмножества, содержащие элемент x_5 ; таковых имеется $\binom{4}{2} = 6$ штук. Из *оставшегося* списка выберем все подмножества, содержащие x_4 ; их $\binom{3}{2} = 3$ штуки. Наконец, у нас остается единственное подмножество элементов, не содержащих ни x_4 , ни x_5 , т.е. подмножество $\{x_1, x_2, x_3\}$.

3.2.5. В заключение данного пункта докажем комбинаторно еще одно важное тождество для биномиальных коэффициентов — так называемое тождество Вандермонда

$$\binom{n+m}{k} = \sum_{i=0}^k \binom{n}{i} \cdot \binom{m}{k-i}.$$

Рассмотрим для этого группу, состоящую из n мужчин и m женщин. По определению, количество способов выбрать из нее команду, состоящую из k человек, равно биномиальному коэффициенту $\binom{n+m}{k}$. С другой стороны, мы можем выбирать эту команду так, чтобы в ней было ровно i мужчин и $(k-i)$ женщин. При фиксированном i количество способов подбора такой команды, согласно правилу произведения, равно $\binom{n}{i} \cdot \binom{m}{k-i}$. Меняя теперь i от нуля до k , мы вновь получим общее количество способов образовать требуемую команду.

Замечание 3.2. Приведенные выше рассуждения очень часто используются в комбинаторике. Они даже имеют специальное название — *принцип double counting* или правило подсчета двумя способами. Основная идея такого рода рассуждений состоит в следующем: если две формулы подсчитывают количество одних и тех же элементов, то эти формулы равны.

3.3. Название “биномиальные коэффициенты” связано с тем, что они, помимо всего прочего, встречаются в формуле бинома Ньютона

$$(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}. \quad (10)$$

3.3.1. Комбинаторное доказательство этой формулы довольно элементарно: нужно просто расписать $(x+y)^n$ в виде произведения n сомножителей

$$(x+y)^n = \underbrace{(x+y)}_1 \cdot \underbrace{(x+y)}_2 \cdot \dots \cdot \underbrace{(x+y)}_n$$

и пометить каждый из таких сомножителей числом в диапазоне от единицы до n . В результате мы имеем множество X , состоящее из n различных экземпляров сомножителей вида $(x+y)$.

После перемножения этих n скобок получается определенный набор слагаемых вида $x^k y^{n-k}$, $k = 0, 1, \dots, n$. Для подсчета количества этих слагаемых при фиксированном значении параметра k заметим, что любое слагаемое $x^k y^{n-k}$ можно получить так: выбрать в n -элементном множестве X k -элементное подмножество, взять в этом подмножестве в качестве сомножителей переменные x , а в оставшемся $(n-k)$ -элементном подмножестве выбрать в качестве сомножителей переменные y . Как следствие, количество слагаемых $x^k y^{n-k}$ совпадает с количеством способов выбрать k -элементное подмножество n -элементного множества X и равно $\binom{n}{k}$.

3.3.2. Формула (10) оказывается чрезвычайно полезной для вывода разного рода соотношений, связанных с биномиальными коэффициентами. Например, полагая в ней $x = y = 1$, получаем тождество

$$\sum_{k=0}^n \binom{n}{k} = 2^n. \quad (11)$$

Заметим теперь, что, суммируя биномиальные коэффициенты $\binom{n}{k}$ по k , мы подсчитываем все подмножества n -элементного множества. Иными словами, мы формально доказали тот факт, что количество всех подмножеств данного n -множества равно 2^n .

3.3.3. Для комбинаторного доказательства равенства (11) воспользуемся чрезвычайно полезным и часто используемым в комбинаторике *принципом биекции*. Формально этот принцип можно сформулировать следующим образом: пусть X, Y — пара конечных множеств, и пусть существует биекция $f: X \rightarrow Y$, т.е. такое отображение, что

$$\forall y \in Y \quad \exists! x \in X : \quad y = f(x).$$

Тогда количество элементов в множествах X и Y совпадают: $|X| = |Y| = n$.

Неформально использование принципа биекции можно проиллюстрировать на следующем примере. Предположим, что вы устраиваете вечеринку и приглашаете на нее довольно много друзей. Как гостеприимный хозяин, вы встречаете всех своих друзей на входе в дом, но запоминаете только пришедших к вам девушек. В какой-то момент вы решаете подсчитать, сколько парней пришло к вам на вечеринку. Вы знаете количество пришедших к вам девушек, и вам кажется, что количество девушек и парней одинаково. Как вам быстро проверить это предположение? Ответ достаточно очевиден: попросить каждую девушку взять ровно одного парня за руку. Если в результате этой процедуры все множество гостей разбилось на пары, то ваше предположение окажется верным. Тем самым вы сильно упростили себе жизнь — вам не пришлось проделывать довольно утомительную работу по пересчету пришедших к вам парней, вы просто воспользовались для их подсчета результатом уже проделанной работы по пересчету пришедших к вам девушек.

Вернемся теперь к равенству (11). Заметим, что любое подмножество A множества X мы можем закодировать бинарной строкой $f(A)$ длины n , то есть строкой над алфавитом $\{0, 1\}$. Единице на i -м месте строки будет при этом отвечать ситуация, при которой $x_i \in A$, а нулю — вариант, при котором $x_i \notin A$. Так, если множество X имеет вид $\{x_1, x_2, x_3, x_4\}$, а подмножество A записывается в виде $A = \{x_2, x_4\}$, то соответствующая этому подмножеству строка длины 4 равна

$$f(A) = (0, 1, 0, 1).$$

Очевидно, что построенное отображение f взаимно-однозначно. Следовательно, количество подмножеств данного n -множества X совпадает с количеством бинарных строк длины n . А таких строк у нас имеется ровно 2^n штук — действительно, на любое из n мест мы двумя способами можем поставить либо ноль, либо единицу.

3.3.4. Полагая в (10) $x = -1, y = 1$, мы получаем еще одно важное тождество

$$\sum_{k=0}^n (-1)^k \binom{n}{k} = 0, \quad n > 0. \quad (12)$$

Для того, чтобы понять комбинаторный смысл равенства (12), перепишем его в следующем виде:

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \binom{n}{3} + \dots = 0 \quad \iff \quad \binom{n}{0} + \binom{n}{2} + \dots = \binom{n}{1} + \binom{n}{3} + \dots$$

В левой части последнего равенства стоит количество всех подмножеств, в которых содержится четное количество элементов, а в правой — число подмножеств, содержащих нечетное количество элементов. Иными словами, следствием равенства (12) является тот факт, что количество четных подмножеств любого множества, отличного от \emptyset , равняется количеству его нечетных подмножеств.

Наконец, из (12) мы также можем заключить, что количество, например, всех нечетных подмножеств ровно в два раза меньше общего количества всех подмножеств. С учетом (11) это означает, что количество всех нечетных подмножеств равно 2^{n-1} .

3.3.5. Наконец, продифференцируем (10) по x :

$$n(x+y)^{n-1} = \sum_{k=1}^n k \binom{n}{k} x^{k-1} y^{n-k}. \quad (13)$$

Подставляя в это равенство $x = y = 1$, получим еще одно полезное равенство:

$$\sum_{k=1}^n k \binom{n}{k} = n 2^{n-1}. \quad (14)$$

3.4. Перейдем теперь к задачам, связанным с подсчетом k -сочетаний с повторениями.

3.4.1. Начнем с примера. Пусть множество X состоит из двух чисел 1 и 2. Выпишем все 3-сочетания с повторениями из 2-множества X :

$$\{1, 1, 1\}, \quad \{1, 1, 2\}, \quad \{1, 2, 2\}, \quad \{2, 2, 2\}.$$

Как видно, таковых оказалось 4 штуки. Для подсчета количества $\binom{n}{k}$ таких мультимножеств в общем случае нам будет удобнее вначале конкретизировать n -множество X , а именно, взять в качестве X множество $[n] := \{1, 2, \dots, n\}$ первых n натуральных чисел. Любое k -мультимножество такого множества можно записать, очевидно, в следующем виде:

$$1 \leq a_1 \leq a_2 \leq \dots \leq a_k \leq n.$$

Например, 3-мультимножество $\{1, 1, 2\}$ над 2-множеством $X = \{1, 2\}$ можно записать так:

$$1 \leq (a_1 = 1) \leq (a_2 = 1) \leq (a_3 = 2) \leq (n = 2).$$

Теперь превратим в этой цепочке все нестрогие неравенства в строгие. Для этого мы к a_2 прибавим единицу, к a_3 — двойку, к a_4 — тройку, и так далее. К последнему числу a_n мы, таким образом, добавим число $(k-1)$. В результате получим цепочку строгих неравенств вида

$$1 \leq a_1 < a_2 + 1 < a_3 + 2 < a_4 + 3 < \dots < a_k + (k-1) \leq n + (k-1).$$

В нашем примере

$$1 \leq (a_1 = 1) < (a_2 + 1 = 2) < (a_3 + 2 = 4) \leq (n + (3-1) = 4).$$

Заметим, что в результате этой операции мы получили некоторое k -элементное подмножество *различных* чисел вида $a_i + (i-1)$ множества $\tilde{X} = [n+k-1]$ всех чисел от единицы до $n+k-1$. Иными словами, мы сопоставили любому k -мультимножеству над множеством $X = [n]$ вполне определенное k -подмножество множества $\tilde{X} = [n+k-1]$. Очевидно, что это сопоставление взаимно-однозначно. Но количество всех k -подмножеств данного множества мы знаем — оно равно $\binom{n+k-1}{k}$. Следовательно, этому числу равно, по принципу биекции, и количество $\binom{n}{k}$ всех k -мультимножеств над множеством $X = [n]$:

$$\binom{n}{k} = \binom{n+k-1}{k}.$$

Справедливость этого равенства для произвольного n -элементного множества X вновь следует из принципа биекции.

4 k -перестановки из n элементов. Урновые схемы и схемы раскладки предметов по ящикам.

4.1. Перейдем теперь к подсчету количества k -перестановок из n элементов.

4.1.1. Напомним, что k -перестановкой из n элементов называется *упорядоченный* набор

$$(a_1, a_2, \dots, a_k)$$

элементов, в котором все a_i принадлежат одному и тому же n -элементному множеству X .

Элементы a_i в наборе (a_1, a_2, \dots, a_k) могут как повторяться, так и не повторяться. В первом случае говорят о k -перестановках с повторениями, во втором — о k -перестановках без повторений.

Номер паспорта — это типичный пример k -перестановки с повторениями над множеством из десяти цифр $X = \{0, 1, \dots, 9\}$. Классическим примером 3-перестановки без повторений является упорядоченный список спортсменов, занявших призовые места в любых спортивных соревнованиях.

4.1.2. В литературе встречается довольно много альтернативных названий для данного объекта. Именно, упорядоченный набор (a_1, a_2, \dots, a_k) , $a_i \in X$, также иногда называется

- k -размещением из n элементов;
- кортежем из k элементов множества X ;
- упорядоченной k -выборкой из n элементов;
- k -мерным вектором над множеством X ;
- k -элементным словом над n -элементным алфавитом.

4.1.3. Сосчитаем количество k -перестановок с повторениями.

Утверждение 4.1. *Количество k -перестановок с повторениями из n элементов равно n^k .*

Для доказательства можно либо просто сослаться на правило произведения, либо рассмотреть (a_1, a_2, \dots, a_k) , $a_i \in X$ как слово из k элементов над алфавитом из $n = |X|$ букв. На первое место в слове мы можем поставить любую из n букв, на второе — также любую из n букв и так далее. Всего же получаем n^k вариантов записать данное слово.

4.1.4. Перейдем теперь к подсчету количества перестановок без повторений.

Утверждение 4.2. *Количество $P(n, k)$ k -перестановок из n элементов без повторений равно*

$$P(n, k) = n \cdot (n - 1) \cdot \dots \cdot (n - k + 1) =: (n)_k.$$

Доказательство очевидно — на первое место в строке длины k я могу поставить любой из n элементов, на второе — любой из оставшихся $(n - 1)$ элементов и так далее.

4.1.5. В частном случае $k = n$ k -перестановки из n элементов без повторений называются просто перестановками n -элементного множества X . Их количество равно

$$P_n \equiv P(n) = n!, \quad P(0) = 0! = 1.$$

4.1.6. Любую k -перестановку из n элементов без повторений можно рассматривать и как упорядоченное k -подмножество n -множества. Количество таких подмножеств мы можем сосчитать следующим образом: мы можем $\binom{n}{k}$ способами выбрать k -подмножество n -элементного множества, а затем $k!$ способами его упорядочить. Действительно, на первое место мы можем поставить любой из k элементов подмножества, на второе — любой из оставшихся $k - 1$ элементов и так далее. Таким образом, мы получаем некоторое новое выражение для количества всех упорядоченных k -элементных подмножеств n -множества, то есть количества k -перестановок из n элементов без повторений, равно

$$(n)_k = k! \cdot \binom{n}{k} \implies \binom{n}{k} = \frac{(n)_k}{k!}.$$

Последняя формула часто используется как некобинаторное определение биномиальных коэффициентов $\binom{n}{k}$ в случае, когда $k \in \mathbb{Z}$, а n принадлежит \mathbb{Z} , \mathbb{R} или даже \mathbb{C} . Именно, по определению,

$$\binom{q}{k} := \begin{cases} \frac{q(q-1) \dots (q-k+1)}{k!} =: \frac{(q)_k}{k!}, & \text{если } k \in \mathbb{N}, \\ 1, & \text{если } k = 0, \\ 0, & \text{если } k < 0, \end{cases} \quad \forall q \in \mathbb{C}.$$

Например,

$$\binom{-1}{3} = \frac{(-1) \cdot (-2) \cdot (-3)}{3 \cdot 2 \cdot 1} = -1.$$

4.1.7. Функцию $(q)_k$ часто также обозначают через $q^{\bar{k}}$ и называют *убывающей факториальной степенью* [?]. Наряду с убывающей можно ввести и так называемую *возрастающую факториальную степень*

$$q^{(k)} \equiv q^{\bar{k}} := q \cdot (q+1) \cdot \dots \cdot (q+k-1).$$

В частности, с ее помощью получается удобное для вычислений выражение для количества $\binom{n}{k}$ сочетаний с повторениями:

$$\binom{\binom{n}{k}}{k} = \binom{n+k-1}{k} = \frac{n^{(k)}}{k!}. \tag{15}$$

4.2. Итак, мы получили простые соотношения для подсчета количества четырех основных объектов элементарной комбинаторики — k -сочетаний и k -перестановок из n элементов с повторениями и без повторений. Эти объекты встречаются в огромном количестве внешне не очень похожих друг на друга задач элементарной комбинаторики. Оказывается, однако, что большинство этих задач можно свести к одной из двух простейших схем — либо к так называемой урновой схеме, либо к схеме раскладки предметов по ящикам.

4.2.1. В урновой схеме имеется урна, в которой находятся n *различимых* предметов. Из урны последовательно вытаскивается k предметов. Задача состоит в подсчете количества различных способов выбора этих предметов, или, как еще говорят, в подсчете различных k -элементных выборок из n предметов, находящихся в урне.

На практике наиболее часто встречаются четыре модификации этой задачи, различающиеся способами формирования k -элементной выборки. Прежде всего, мы можем возвращать или не возвращать вытаскиваемые предметы обратно в урну. В первом случае говорят о *выборке с повторениями*, во втором — о *выборке без повторений*. Далее, в некоторых задачах нам важен порядок вытаскиваемых предметов. В этом случае имеем так называемые *упорядоченные* выборки. В противном случае выборки называются *неупорядоченными*.

Нетрудно понять, что задачи о подсчете k -элементных выборок представляют собой, по сути, те же самые задачи о подсчете k -перестановок или k -сочетаний из n элементов. Действительно, любая неупорядоченная k -элементная выборка представляет собой либо k -элементное подмножество n -множества, либо k -мультимножество над n -элементным множеством в зависимости от того, возвращаем мы вытаскиваемые предметы обратно в урну или не возвращаем. Следовательно, количество таких неупорядоченных выборок совпадает с коэффициентами $\binom{n}{k}$ или $\left(\!\!\binom{n}{k}\!\!\right)$. Очевидно также, что любая упорядоченная k -элементная выборка есть просто некоторая k -перестановка n -элементного множества. Поэтому количество таких выборок равно n^k или $(n)_k$ в зависимости от того, говорим ли мы о выборке с повторениями или без повторений.

В результате получаем следующую таблицу решений задач, связанных с урновыми схемами:

Предметы на выходе	с возвращением	без возвращения
упорядоченные	n^k	$(n)_k$
неупорядоченные	$\left(\!\!\binom{n}{k}\!\!\right)$	$\binom{n}{k}$

4.2.2. Приведем несколько характерных примеров, достаточно естественно сводящихся к одной из описанных выше урновых схем.

Пример 4.3. Предположим, что у нас имеется некоторое общество, состоящее из двадцати членов. Сколькими способами можно выбрать президента, вице-президента, секретаря и казначея этого общества?

Решение. Очевидно, что любой способ выбора представляет собой упорядоченное 4-элементное подмножество 20-элементного множества. Следовательно, существует ровно $(20)_4$ различных способов выбора членов общества на эти должности.

Пример 4.4. Для того, чтобы открыть сейф, нужно набрать код из пяти символов с помощью вращающихся дисков. На каждом из этих дисков нанесено 12 символов, одинаковых для каждого из дисков. Сколько вариантов различных кодов существует?

Решение. Любой код представляет собой упорядоченную 5-элементную выборку с повторениями или, иначе, строку из пяти символов над алфавитом из 12 букв. Следовательно, имеется 12^5 вариантов различных кодов.

Пример 4.5. На почте продаются открытки десяти различных видов. Сколькими способами можно купить восемь открыток? А восемь открыток разных видов?

Решение. Понятно, что в первом случае любой набор из восьми открыток представляет собой неупорядоченную выборку с повторениями, а во втором — выборку без повторений из 10 элементов. Следовательно, в первом случае имеем $\left(\!\!\binom{10}{8}\!\!\right)$, а во втором — $\binom{10}{8}$ способов покупки восьми открыток.

4.2.3. Второй, не менее популярной в элементарной комбинаторике схемой, связанной с подсчетом количества k -перестановок и k -сочетаний, является схема раскладки предметов по ящикам. В этой схеме имеется n *различимых* ящиков, по которым нужно разложить k *различимых* или *неразличимых* предметов. При этом мы можем накладывать определенные ограничения на количество предметов в каждом ящике.

Рассмотрим, к примеру, задачу о подсчете количества способов раскладки k различимых предметов по n различным ящикам при условии, что в любой ящик можно класть любое количество предметов. Количество способов совершить эти действия равно, очевидно, n^k . Действительно, любой предмет мы можем положить в любой из n ящиков вне зависимости от того, куда мы положили оставшиеся предметы. Поэтому, согласно правилу произведения, это количество равно n^k . Иными словами, данная задача представляет собой переформулировку задачи о подсчете количества k -перестановок из n элементов с повторениями.

Теперь предположим, что в той же схеме мы не имеем права класть более одного предмета в один ящик. Тогда первый предмет мы можем поместить в любой из n ящиков, второй — в любой из оставшихся свободными $(n - 1)$ ящиков и так далее. Всего же получаем $(n)_k$ способов раскладки. Следовательно, данная схема соответствует подсчету k -перестановок без повторений.

4.2.4. Пусть теперь у нас имеются n различимых ящиков и k неразличимых предметов. Тогда подсчет количества различных способов раскладки этих предметов по ящикам сводится к задаче о подсчете количества k -сочетаний из n элементов.

Действительно, в данной схеме в качестве n -элементного множества выступает множество, состоящее из n различимых ящиков. В случае, когда в каждый ящик можно класть не более одного предмета, мы, раскладывая предметы по ящикам, выделяем в этом множестве некоторое k -элементное подмножество. Следовательно, количество таких раскладок совпадает с количеством различных k -элементных подмножеств n -множества и равно $\binom{n}{k}$.

В случае же, когда никаких ограничений на количество предметов в ящике не накладывается, мы, раскладывая по ящикам k неразличимых предметов, задаем тем самым некоторое k -мультимножество n -множества X . Поэтому количество различных способов такой раскладки равно количеству $\binom{n}{k}$ k -сочетаний из n элементов с повторениями.

Подводя итоги, построим таблицу рассмотренных схем раскладок k предметов по n ящикам:

Предметы	Ящики	Произвольное количество предметов в ящике	Не более одного предмета в ящике
различимые	различимые	n^k	$(n)_k$
неразличимые	различимые	$\binom{n}{k}$	$\binom{n}{k}$

4.2.5. Проиллюстрируем некоторые характерные примеры задач, которые довольно естественно сводятся к схеме схем раскладки предметов по ящикам.

Пример 4.6. Сколькими способами можно разложить по двум *различимым* карманам (например, левому и правому) девять монет *различного* достоинства?

Решение. Рассматриваемый пример является типичной задачей, которая естественным образом сводится к схеме раскладки предметов по ящикам. Действительно, в роли ящиков здесь выступают левый и правый карман, а в роли предметов — девять различных монет. Поэтому ответ в этой задаче — 2^9 способов.

Замечание 4.7. Рассмотренная задача, однако, не всегда решается верно: в качестве ответа иногда выдают 9^2 способов. Путаница, как правило, происходит потому, что эту задачу пытаются свести к урновой схеме, считая, что имеются урна, в которой расположены 9 различных предметов, а также 2 различные позиции на выходе.

Для того, чтобы этой путаницы избежать, полезно сформулировать следующие основные отличия схемы раскладки по ящикам от соответствующей ей урновой схемы. Во-первых, в схеме раскладки предметов по ящикам предметы обратно не возвращаются, они остаются в ящике. Во-вторых, в этой схеме в любой ящик можно класть любое количество предметов. В аналогичной урновой схеме на любую позицию помещается ровно один предмет.

Приведем теперь два характерных примера, связанных с раскладкой неразличимых предметов по различимым ящикам.

Пример 4.8. У отца имеется 5 (неразличимых) апельсинов, которые он может раздать восьми своим сыновьям. Если его задача состоит в том, чтобы раздать их максимальному количеству сыновей, то он должен поставить дополнительное условие — любой из его сыновей не должен получить более одного апельсина. В этом случае количество способов, которыми он может раздать своим сыновьям эти пять апельсинов, равно, очевидно, $\binom{8}{5}$. Если же он раздает их по каким-то заслугам, и может, таким образом, любому сыну отдать любое количество апельсинов, то количество способов это сделать равно $\binom{8}{5} = \binom{12}{5}$.

Пример 4.9. В физике встречаются задачи, в которых имеются n различных уровней энергии и k неразличимых элементарных частиц. Если эти частицы — фермионы, то для них действует так называемый принцип запрета Паули, согласно которому на любом энергетическом уровне может находиться не более одной элементарной частицы. Как следствие, количество различных распределений k фермионов по n энергетическим уровням равно $\binom{n}{k}$. Наряду с фермионами существуют и частицы иного сорта — бозоны, для которых не существует ограничений на количество частиц, занимающих один и тот же уровень энергии. Для бозонов количество таких распределений равно, очевидно, $\binom{n+k-1}{k}$.

4.2.6. К задачам раскладки неразличимых предметов по различимым ящикам, связанным с подсчетом количества k -сочетаний, сводятся также задачи о так называемом *разбиении* натурального числа k на n слагаемых. Данная задача формулируется следующим образом: сколькими способами можно представить натуральное число k в виде суммы n слагаемых вида

$$a_1 + a_2 + \dots + a_n = k$$

при условии, что порядок слагаемых важен, то есть при условии, что разбиения вида

$$1 + 3 + 3 + 3 = 10 \quad \text{и} \quad 3 + 1 + 3 + 3 = 10$$

считаются различными?

Если на числа a_i накладывается единственное условие вида $a_i \geq 0$, то количество разбиений равно количеству $\binom{n+k-1}{k}$ k -мультимножеств над n -элементным множеством. Действительно, в упорядоченной сумме $a_1 + a_2 + \dots + a_n$ любой индекс i слагаемого a_i можно рассматривать как i -й

ящик, в который мы складываем a_i единиц. Следовательно, эту задачу можно трактовать как задачу о раскладке k “неразличимых” единиц по n различным ящикам.

Пример 4.10. Подсчитать количество разбиений числа $k = 4$ на два слагаемых:

$$a_1 + a_2 = 4, \quad a_1, a_2 \geq 0.$$

Ответ: $\binom{2}{4} = \binom{5}{4} = 5$ разбиений: $0 + 4 = 1 + 3 = 2 + 2 = 3 + 1 = 4 + 0 = 4$.

К подсчету числа k -сочетаний из n элементов без повторений задача о разбиении числа k сводится в случае, когда на числа a_i накладываются следующие условия:

$$a_i = 0 \quad \text{или} \quad a_i = 1.$$

В этом случае индекс i также можно трактовать как i -й ящик; его можно выбрать (положив $a_i = 1$) или не выбрать (положив $a_i = 0$). Всего же нужно выбрать k таких ящиков. Это можно сделать $\binom{n}{k}$ способами.

Пример 4.11. Подсчитать количество разбиений числа 2 на три слагаемых при условии, что любое слагаемое может принимать значения 0 или 1.

Ответ: $\binom{3}{2} = 3$ разбиения: $1 + 1 + 0 = 1 + 0 + 1 = 0 + 1 + 1 = 2$.

4.2.7. Достаточно часто на практике встречаются ситуации, когда одну и ту же задачу можно свести и к урновой схеме, и к схеме раскладки предметов по ящикам.

Пример 4.12. В кондитерском магазине продаются пирожные трех разных видов. Сколькими различными способами можно купить семь пирожных?

Решение. Ответ в этой задаче, очевидно, равен $\binom{3}{7} = \binom{9}{7}$. Этот ответ можно, например, получить, представляя себе коробку с тремя отделениями, в каждое из которых кладется пирожное только одного вида; в этом случае мы сводим задачу к подсчету количества раскладок семи неразличимых предметов по трем различным ящикам. Другой способ получить тот же ответ — это представлять себе урну, в которой находятся три разных пирожных, и считать количество способов выбора из урны семи пирожных с возвращениями любого выбранного пирожного обратно в урну. Наконец, можно вообще забыть о любых схемах, если понимать, что любые семь купленных пирожных трех различных видов представляют собой 7-мультимножество над 3-элементным множеством различных видов пирожных.

5 Подсчет количества отображений конечных множеств. Числа Стирлинга второго рода

5.1. Оказывается, задачи о раскладке различных предметов по различным же ящикам имеют и еще одну, чрезвычайно важную комбинаторную интерпретацию — они эквивалентны задачам о подсчете количества отображений конечных множеств.

5.1.1. Напомним определение произвольного отображения $f: X \rightarrow Y$.

Определение 5.1. Пусть X, Y — пара конечных множеств. Отображением f из X в Y называется правило, согласно которому любому элементу $x \in X$ ставится в соответствие единственный элемент $y \in Y$:

$$\forall x \in X \quad \exists! y \in Y: \quad y = f(x).$$

С комбинаторной точки зрения любое отображение f из n -элементного множества X в k -элементное множество Y можно рассматривать как некоторый вариант раскладки n различных предметов по k различным ящикам при отсутствии каких-либо ограничений на количество предметов в каждом ящике.

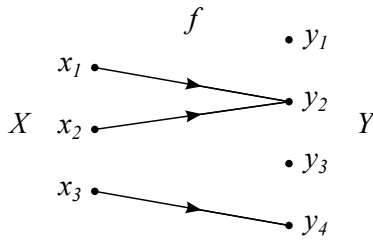


Рис. 6

Пример 5.2. Рассмотрим отображение f из трехэлементного множества X в четырехэлементное множество Y вида

$$f(x_1) = y_2, \quad f(x_2) = y_2, \quad f(x_3) = y_4$$

(смотри рис.6). Этому отображению отвечает раскладка трех различных предметов по четырем различным ящикам, при которой первые два предмета размещаются во втором ящике, а третий предмет — в четвертом ящике.

Как следствие, общее количество всех отображений n -элементного множества X в k -элементное множество Y равно k^n .

5.1.2. Напомним теперь определение *инъективного* отображения.

Определение 5.3. Отображение $f: X \rightarrow Y$ называется инъективным, если из условия

$$f(x_1) = f(x_2) \quad \implies \quad x_1 = x_2.$$

Иными словами, отображение называется инъективным, если у любого элемента $y \in Y$ имеется не более одного прообраза, т.е. элемента $x \in X$, такого, что $y = f(x)$.

Понятно, что любому инъективному отображению $f: X \rightarrow Y$ отвечает такая раскладка n элементов множества X , при которой в каждом из k ящиков находится не более одного элемента. Как следствие, количество всевозможных инъективных отображений равно $(k)_n$.

5.1.3. Наконец, рассмотрим случай биективного и сюръективного отображений.

Определение 5.4. Отображение $f: X \rightarrow Y$ называется *биективным*, если

$$\forall y \in Y \quad \exists! x \in X: \quad y = f(x).$$

Количество таких отображений равно, очевидно, $n!$, где $n = |X| = |Y|$.

Определение 5.5. Отображение $f: X \rightarrow Y$ называется сюръективным, если

$$\forall y \in Y \quad \exists x \in X: \quad y = f(x).$$

Другими словами, отображение f сюръективно, если у любого элемента $y \in Y$ найдется хотя бы один прообраз $x \in X$, то есть такой x , что $f(x) = y$.

Комбинаторная интерпретация сюръективного отображения такова: это есть некоторая раскладка n различных предметов по k различным ящикам при условии, что в каждом ящике находится хотя бы один предмет. Количество таких раскладок при $k > n$ равно, очевидно, нулю. Задача следующего пункта данного параграфа — сосчитать количество этих раскладок для случая $0 \leq k \leq n$.

5.2. Обозначим через $\widehat{S}(n, k)$ количество всех сюръективных отображений n -элементного множества X в k -элементное множество Y . Сосчитаем $\widehat{S}(n, k)$ для случая $n \geq 0$, $0 \leq k \leq n$.

5.2.1. Рассмотрим множество *всех* отображений из n -множества X в k -множество Y . Как мы знаем, количество таких отображений равно k^n . Наша задача состоит в том, чтобы подсчитать это количество по-другому, выразив k^n через числа $\widehat{S}(n, k)$.

5.2.2. Заметим, что *любое* отображение $f: X \rightarrow Y$ можно рассматривать как *сюръективное* отображение множества X на множество

$$\text{Im}(f) = \{y \in Y \mid \exists x: y = f(x)\},$$

являющееся образом множества X при отображении f .

Так, для отображения f из примера 5.2 образ $\text{Im}(f) = \{y_2, y_4\}$, а отображение $f: X \rightarrow Y$ является сюръективным отображением множества X на подмножество $\text{Im}(f) \subset Y$.

5.2.3. Разобьем теперь все множество отображений $f: X \rightarrow Y$ на блоки, включив в i -й блок все отображения, образ $\text{Im}(f)$ которых содержит ровно i элементов: $|\text{Im}(f)| = i$, $i = 1, \dots, k$. Все, что нам остается — это сосчитать количество элементов в каждом блоке, а затем воспользоваться правилом суммы для того, чтобы получить общее количество k^n всех отображений.

5.2.4. Заметим, что существует $\binom{k}{i}$ способов выбрать i -элементное подмножество k -множества Y . Для каждого из этих подмножеств имеется $\widehat{S}(n, i)$ различных сюръективных отображений из n -элементного множества X в выбранное i -элементное подмножество множества Y . Таким образом, по правилу произведения, общее количество элементов в i -м блоке равно

$$\binom{k}{i} \cdot \widehat{S}(n, i).$$

Тогда по правилу суммы можем записать, что

$$k^n = \sum_{i=0}^k \binom{k}{i} \cdot \widehat{S}(n, i). \quad (16)$$

При этом мы суммируем не от 1 до k , а от 0 до k , учитывая, что $\widehat{S}(n, 0) = 0$ для всех $n > 0$.

Замечание 5.6. Формулу (16) полезно иногда записывать в виде

$$k^n = \sum_{i=0}^n \binom{k}{i} \cdot \widehat{S}(n, i). \quad (17)$$

Несложно убедиться в том, что эта формула непосредственно следует из (16), а также в том, что она оказывается справедливой как для случая $n \geq k$, так и для случая $n < k$.

5.2.5. Мы выразили количество всех отображений n -элементного множества X в k -элементное множество Y через количество $\widehat{S}(n, i)$ сюръективных отображений. Нам же нужна обратная формула, выражающая количество $\widehat{S}(n, k)$ сюръективных отображений через число i^n всех отображений. Для ее получения воспользуемся так называемыми *формулами обращения*.

Утверждение 5.7. Пусть (f_0, f_1, f_2, \dots) и (g_0, g_1, g_2, \dots) — две числовые последовательности, и пусть одна из них выражается через вторую по формулам

$$f_k = \sum_{i=0}^k \binom{k}{i} g_i, \quad k \geq 0. \quad (18)$$

Тогда справедлива следующая формула обращения:

$$g_k = \sum_{i=0}^k (-1)^{k-i} \binom{k}{i} f_i, \quad k \geq 0. \quad (19)$$

С учетом этих формул обращения можно, считая n параметром, из соотношения (16) получить следующую явную формулу для вычисления чисел $\widehat{S}(n, k)$:

$$\widehat{S}(n, k) = \sum_{i=0}^k (-1)^{k-i} \binom{k}{i} \cdot i^n.$$

5.3. Задачи подсчета количества отображений n -элементного множества X в k -элементное множество Y имеют еще одну важную комбинаторную интерпретацию.

5.3.1. Начнем с простого примера.

Пример 5.8. Для трехэлементного множества $X = \{x_1, x_2, x_3\}$ и двухэлементного множества $Y = \{y_1, y_2\}$ имеется, как мы знаем, $2^3 = 8$ различных отображений множества X в множество Y . Запишем все эти отображения как упорядоченные пары подмножеств множества X :

$$\begin{aligned} &(\{x_1, x_2, x_3\}, \emptyset), & (\{x_1, x_2\}, \{x_3\}), & (\{x_1, x_3\}, \{x_2\}), & (\{x_2, x_3\}, \{x_1\}), \\ &(\{x_1\}, \{x_2, x_3\}), & (\{x_2\}, \{x_1, x_3\}), & (\{x_3\}, \{x_1, x_2\}), & (\emptyset, \{x_1, x_2, x_3\}). \end{aligned}$$

Видно, что записанное в таком виде решение представляет собой не что иное, как список всех возможных *разделений* множества X , то есть упорядоченных разбиений X на два блока, один из которых может быть и пустым.

5.3.2. Очевидно, что данный результат справедлив и в общем случае. Именно, любое отображение $f: X \rightarrow Y$ задает нам некоторое разделение множества X , то есть разбиение этого множества на k упорядоченных блоков, часть из которых могут быть пустыми. Как следствие, количество таких разделений совпадает с количеством всех отображений f и равно k^n .

Аналогичные рассуждения показывают, что любое сюръективное отображение $f: X \rightarrow Y$ задает нам некоторое *упорядоченное разбиение* множества X на блоки. Поэтому количество всех упорядоченных разбиений n -элементного множества X на k блоков равно числу $\widehat{S}(n, k)$.

5.3.3. Рассмотрим теперь некоторый специальный вид k -разделений множества X , а именно, такие k -разделения, в которых в первом блоке содержится a_1 элемент, во втором блоке — a_2 элемента, в k -м блоке — a_k элементов. Очевидно, что при этом общая сумма всех элементов должна быть равна мощности $|X| = n$:

$$a_1 + a_2 + \dots + a_k = n, \quad a_i \geq 0.$$

Утверждение 5.9. *Количество всех таких k -разделений n -множества X равно*

$$P(n; a_1, a_2, \dots, a_k) := \binom{n}{a_1} \cdot \binom{n-a_1}{a_2} \cdot \dots \cdot \binom{n-a_1-a_2-\dots-a_{k-1}}{a_k} = \frac{n!}{a_1! \cdot a_2! \cdot \dots \cdot a_k!}. \quad (20)$$

Доказательство. Действительно, из любого n -элементного множества X мы $\binom{n}{a_1}$ способами можем выбрать a_1 элементов и положить их в первый ящик (отнести к первому блоку разбиения). Затем для каждого такого выбора мы $\binom{n-a_1}{a_2}$ способами можем из оставшегося $(n - a_1)$ -элементного множества выбрать a_2 элементов и положить их во второй ящик (отнести ко второму блоку разбиения), и так далее. Формула (20), описывающая общее количество способов совершить все эти действия, следует теперь из правила произведения. \square

Следствие 5.10. *Общее количество k^n всех k -разделений n -множества X выражается через числа $P(n; a_1, a_2, \dots, a_k)$ по формуле*

$$k^n = \sum_{\substack{a_1+a_2+\dots+a_k=n \\ a_i \geq 0}} P(n; a_1, a_2, \dots, a_k) = \sum_{\substack{a_1+a_2+\dots+a_k=n \\ a_i \geq 0}} \frac{n!}{a_1! \cdot a_2! \cdot \dots \cdot a_k!}. \quad (21)$$

Замечание 5.11. Если в условии рассматриваемой задачи заменить нестрогие неравенства $a_i \geq 0$ на строгие, то есть на неравенства $a_i > 0$, то вместо разделения мы получим упорядоченное разбиение специального вида. Количество таких упорядоченных разбиений также описывается формулой (20), а вместо формулы (21) получается не менее полезное соотношение вида

$$\widehat{S}(n, k) = \sum_{\substack{a_1+a_2+\dots+a_k=n \\ a_i > 0}} \frac{n!}{a_1! \cdot a_2! \cdot \dots \cdot a_k!}. \quad (22)$$

5.4. Числа $P(n; a_1, a_2, \dots, a_k)$ имеют и еще один важный комбинаторный смысл. Именно, они перечисляют так называемые *перестановки n -множества X с повторениями*.

5.4.1. Рассмотрим в качестве элементарного примера следующую задачу: на полке имеются 15 различных книг по математике, 16 по информатике и 12 по физике; каково количество способов перестановки этих книг на полке? Ответ очевиден: $(15 + 16 + 12)! = 43!$ способов.

Предположим теперь, что мы перестали различать книги, посвященные одному и тому же предмету. В этом случае количество различных способов перестановки таких книг уменьшится. Обозначим это количество через λ_n . Так как существует $15!$ способов упорядочить книги по математике, $16!$ — по информатике и $12!$ — по физике, то по правилу произведения мы можем записать, что

$$43! = \lambda_n \cdot 16! \cdot 15! \cdot 12! \quad \implies \quad \lambda_n = \frac{43!}{16! \cdot 15! \cdot 12!} = P(43; 15, 16, 12).$$

5.4.2. Аналогичные рассуждения справедливы и в общем случае. Именно, пусть среди n переставляемых предметов имеется a_1 неразличимых предметов первого сорта, a_2 неразличимых

предметов второго сорта и так далее, причем $a_1 + a_2 + \dots + a_k = n$. Тогда для количества перестановок таких предметов с повторениями получаем уже знакомую нам формулу

$$\frac{n!}{a_1! \cdot a_2! \cdot \dots \cdot a_k!} = P(n; a_1, a_2, \dots, a_k).$$

5.4.3. В частном случае перестановки n предметов двух различных сортов получаем

$$P(n; k, n - k) = \frac{n!}{k! \cdot (n - k)!} = \binom{n}{k}.$$

Иными словами, количество таких перестановок совпадает с количеством различных k -элементных подмножеств n -элементного множества. Для комбинаторного доказательства данного факта можно воспользоваться рассуждениями, которые мы проводили при подсчете количества всех подмножеств данного множества. Напомним, что там мы любое подмножество кодировали упорядоченной битовой строкой длины n , состоящей из k единиц и $(n - k)$ нулей. Осталось заметить, что любая такая строка представляет собой некоторую перестановку k элементов первого сорта (единиц) и $(n - k)$ элементов второго сорта (нулей).

5.4.4. В заключение отметим еще одну полезную биекцию, позволяющую несколько по-другому сосчитать количество k -мультимножеств n -элементного множества. Мы знаем, что любому k -мультимножеству над n -множеством отвечает некоторая раскладка k неразличимых предметов по n различным ящикам. В свою очередь, любую такую раскладку можно рассматривать как упорядоченный набор, состоящий из k неразличимых предметов одного сорта (например, k неразличимых шаров), и $(n - 1)$ -го предмета второго сорта ($(n - 1)$ -й неразличимой перегородки между этими шарами). Как следствие, количество всех k -мультимножеств

$$\binom{\binom{n}{k}}{k} = P(k + n - 1; k, n - 1) = \frac{(n + k - 1)!}{(n - 1)! \cdot k!} = \binom{n + k - 1}{k}.$$

5.5. Вернемся к числам $\widehat{S}(n, k)$, описывающим, в частности, количество всех упорядоченных разбиений n -множества на k блоков. Обозначим теперь через $S(n, k)$ количество обычных, неупорядоченных разбиений n -множества на k блоков. Так как для любого неупорядоченного разбиения существует $k!$ способов упорядочить k его блоков, то

$$\widehat{S}(n, k) = k! S(n, k).$$

Отсюда с учетом (16) и (22) получаем следующие две явные формулы, позволяющие вычислять числа $S(n, k)$:

$$S(n, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k - i)^n = \frac{1}{k!} \sum_{\substack{a_1 + a_2 + \dots + a_k = n \\ a_i > 0}} \frac{n!}{a_1! \cdot a_2! \cdot \dots \cdot a_k!}. \quad (23)$$

Числа $S(n, k)$ называются *числами Стирлинга второго рода* и встречаются в большом количестве комбинаторных приложений. Исследуем эти числа поподробнее.

5.5.1. Для практического расчета чисел $S(n, k)$ удобно использовать рекуррентные соотношения.

Утверждение 5.12. Числа Стирлинга второго рода удовлетворяют следующим рекуррентным соотношениям:

$$S(n, k) = S(n - 1, k - 1) + k \cdot S(n - 1, k), \quad k = 1, \dots, n; \quad (24)$$

$$S(0, 0) = 1; \quad S(n, 0) = 0 \quad \forall n > 0; \quad S(n, k) = 0 \quad \forall k > n.$$

Доказательство. Граничные условия $S(n, 0) = 0$ для всех $n > 0$ и $S(n, k) = 0$ для всех $k > n$ очевидны — n -элементное множество в случае $n > 0$ нельзя разбить на 0 блоков, а также на k блоков в случае, когда $k > n$. Равенство $S(0, 0) = 1$ введено просто для удобства.

Докажем теперь соотношение (24). Разобьем для этого множество всех k -разбиений на два блока. К первому блоку Σ_1 отнесем все разбиения, содержащие одноэлементное подмножество $\{x_1\}$. Ко второму блоку Σ_2 отнесем все оставшиеся k -разбиения, т.е. разбиения, в которых элемент x_1 входит в подмножества, содержащие как минимум два элемента.

Рассмотрим, к примеру, все 2-разбиения множества $X = \{x_1, x_2, x_3, x_4\}$:

$$\begin{aligned} & \{\{x_1, x_2, x_3\}, \{x_4\}\}, \quad \{\{x_1, x_2, x_4\}, \{x_3\}\}, \quad \{\{x_1, x_3, x_4\}, \{x_2\}\}, \quad \{\{x_2, x_3, x_4\}, \{x_1\}\}, \\ & \{\{x_1, x_2\}, \{x_3, x_4\}\}, \quad \{\{x_1, x_3\}, \{x_2, x_4\}\}, \quad \{\{x_1, x_4\}, \{x_2, x_3\}\}. \end{aligned}$$

Для этого примера к первому блоку относится единственное разбиение вида $\{\{x_2, x_3, x_4\}, \{x_1\}\}$. Остальные шесть разбиений относятся в данном случае ко второму блоку.

Довольно очевидно, что количество элементов в первом блоке равно количеству $S(n - 1, k - 1)$ всех $(k - 1)$ -разбиений оставшегося $(n - 1)$ -элементного множества. В примере это число равно $S(3, 1) = 1$ — любое множество можно только одним способом разбить на один блок.

Для подсчета количества элементов во втором блоке заметим, что, по сути дела, это число равно количеству всевозможных разбиений $(n - 1)$ -элементного множества на k непустых подмножеств с поочередным добавлением элемента x_1 в каждое из этих подмножеств. Действительно, для разобранный выше примера имеется три разбиения трехэлементного множества $\{x_2, x_3, x_4\}$ на два непустых подмножества, а именно, разбиения вида

$$\{\{x_2\}, \{x_3, x_4\}\}, \quad \{\{x_3\}, \{x_2, x_4\}\}, \quad \{\{x_4\}, \{x_2, x_3\}\}.$$

Добавляя к каждому из этих подмножеств элемент x_1 , получим $2 \cdot 3 = 6$ выписанных выше разбиений четырехэлементного множества X на блоки требуемого вида. Количество таких разбиений в общем случае равно, очевидно, $k \cdot S(n - 1, k)$. \square

5.5.2. Как и биномиальные коэффициенты, числа Стирлинга второго рода удобно представлять в виде треугольного массива на плоскости (n, k) . На рис.7 показаны первые несколько строк такого массива, вычисленных с помощью рекуррентных соотношений (24). Такой рисунок дает нам еще одну комбинаторную интерпретацию чисел $S(n, k)$ — это есть количество *взвешенных* путей, идущих из начала координат в точку с координатами (n, k) .

5.5.3. Числа Стирлинга второго рода встречается в очень большом количестве самых разнообразных задач. В качестве примера вернемся к формуле (17) и перепишем ее через числа Стирлинга второго рода:

$$k^n = \sum_{i=0}^n \binom{k}{i} \cdot i! \cdot S(n, i) = \sum_{i=0}^n (k)_i \cdot S(n, i). \quad (25)$$

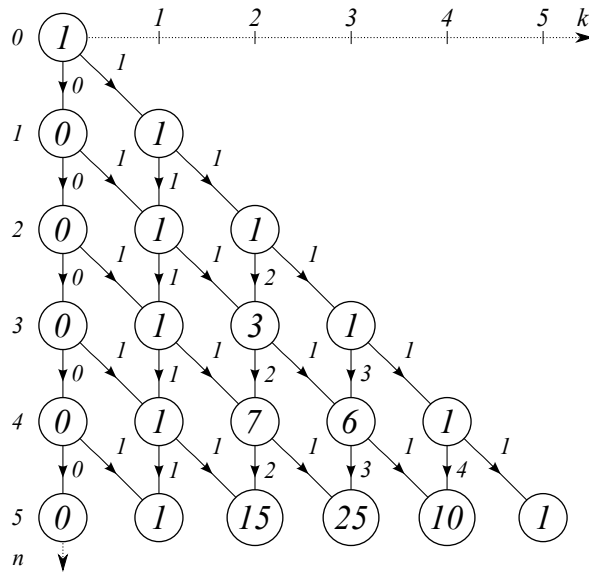


Рис. 7: Графическое представление чисел $S(n, k)$

Оказывается, эта формула справедлива для любых вещественных и даже комплексных значений k . Именно, справедливо равенство

$$x^n = \sum_{i=0}^n (x)_i \cdot S(n, i).$$

Эта формула активно используется в теории конечных операторов — она позволяет перейти от базиса x^n к базису $(x)_n$.

5.5.4. С точки зрения схемы раскладки предметов по ящикам числа Стирлинга второго рода описывают количество способов разложить n различных предметов по k неразличимым ящикам так, чтобы в любом ящике содержался хотя бы один предмет. Сосчитаем, зная $S(n, k)$, количество $B(n, k)$ различных раскладок n различных предметов по k неразличимым ящикам в случае, когда ограничения на количество предметов в любом ящике отсутствуют.

Для этого вновь разобьем множество всех таких раскладок на блоки, поместив в i -й блок все раскладки, в которых занято ровно i ящиков. В случае различных ящиков в процессе аналогичного разбиения множества на блоки мы в i -м блоке имели $\binom{k}{i} \cdot \widehat{S}(n, i)$ элементов. В случае неразличимых ящиков мы i из k ящиков выбираем ровно одним способом, а затем $S(n, i)$ способами заполняем эти ящики n предметами. Используя правило суммы, получаем отсюда следующее выражение для чисел $B(n, k)$:

$$B(n, k) = \sum_{i=1}^k S(n, i).$$

5.5.5. Числа $B(n, k)$ в случае $n = k$ называются числами Белла $B(n)$. Эти числа перечисляют количество *всех* возможных разбиений n -элементного множества X .

Заметим, что любое разбиение множества X можно получить, введя на этом множестве некоторое отношение эквивалентности. Как следствие, количество всех возможных отношений эквивалентности на n -элементном множестве X описывается числами Белла B_n .

5.5.6. Покажем, что для чисел Белла $B(n)$ справедливо следующее рекуррентное соотношение:

$$B_{n+1} = \sum_{k=0}^n \binom{n}{k} B_k. \quad (26)$$

Действительно, рассмотрим блок разбиения $(n+1)$ -элементного множества, содержащий число $n+1$. Все возможные разбиения мы можем разбить на $n+1$ группу (блок) в зависимости от того, сколько чисел содержится вместе с $n+1$. Предположим, что вместе с $n+1$ находятся i чисел, $i = 0, \dots, n$. Эти числа мы можем выбрать $\binom{n}{i} = \binom{n-i}{n-i}$ способами. Оставшиеся $k = n-i$ чисел, $k = 0, \dots, n$, мы можем $B(k)$ способами разбить на блоки. Суммируя по всем k , мы и получим формулу (26).

5.6. Подведем итоги, построив таблицу различных схем раскладок n предметов по k ящикам:

Элементы множества X (предметы)	Элементы множества Y (ящики)	Произвольное количество предметов в ящике	Не более 1 предмета в ящике	Как минимум 1 предмет в ящике
различимые	различимые	k^n	$(k)_n$	$\widehat{S}(n, k)$
неразличимые	различимые	$\binom{k}{n}$	$\binom{k}{n}$	$\binom{n-1}{k-1}$
различимые	неразличимые	$B(n, k)$	$0, \quad n > k$ $1, \quad n \leq k$	$S(n, k)$

6 Рекуррентные соотношения

6.1. Мы уже несколько раз получали решения комбинаторных задач, записанные в виде тех или иных рекуррентных соотношений. Настало время поговорить о них немного подробнее.

6.1.1. Начнем с простого примера [?].

Пример 6.1. Популяция лягушек в озере увеличивается в четыре раза каждый год. В последний день каждого года 100 лягушек отлавливают и переправляют на другие озера. Предполагая, что в начале первого года в озере было 50 лягушек, найти количество лягушек в начале любого последующего года.

Решение. Обозначим через a_n количество лягушек в начале $(n+1)$ -го года. По условию задачи, $a_0 = 50$. Тогда, очевидно,

$$a_1 = 4 \cdot 50 - 100 = 100, \quad a_2 = 4 \cdot 100 - 100 = 300,$$

а в общем случае

$$a_{n+1} = 4 \cdot a_n - 100, \quad n = 0, 1, 2, \dots$$

Полученное равенство является простейшим примером *рекуррентного соотношения*.

6.1.2. Перейдем теперь к формальным определениям.

Определение 6.2. Пусть a_0, a_1, a_2, \dots — произвольная числовая последовательность. Если для любого $n \geq 0$ число a_{n+m} является некоторой функцией от m предыдущих членов последовательности, т.е.

$$a_{n+m} = f_n(a_n, a_{n+1}, \dots, a_{n+m-1}), \quad (27)$$

то такая последовательность называется рекуррентной последовательностью, а соотношение (27) — рекуррентным соотношением m -го порядка.

В частном случае линейной функции f имеем так называемое *линейное рекуррентное соотношение*

$$a_{n+m} = b_1(n) a_{n+m-1} + b_2(n) a_{n+m-2} + \dots + b_{m-1}(n) a_{n+1} + b_m(n) a_n + u(n). \quad (28)$$

В случае $u(n) = 0$ оно называется *однородным*, в противном случае — *неоднородным*.

Самый простой случай рекуррентного соотношения (28) — это *линейное однородное рекуррентное соотношение с постоянными коэффициентами*

$$a_{n+m} = b_1 a_{n+m-1} + b_2 a_{n+m-2} + \dots + b_{m-1} a_{n+1} + b_m a_n. \quad (29)$$

Очевидно, что для однозначного определения всех a_n необходимо наряду с самим рекуррентным соотношением (27) задать и первые m членов a_0, a_1, \dots, a_{m-1} данной последовательности, то есть, как говорят, *задать начальные условия* для рекуррентного соотношения.

6.1.3. Итак, наличие рекуррентного соотношения и начальных условий позволяет нам последовательно, шаг за шагом, определить любое наперед заданное количество n членов рекуррентной числовой последовательности. Зачастую это все, что нам требуется от задачи. Иными словами, получение рекуррентного соотношения для искомой числовой последовательности a_n является вполне приемлемым, а зачастую — и наиболее удобным с вычислительной точки зрения ответом для поставленной комбинаторной задачи.

Однако иногда нам полезно иметь явную формулу, которая для любого n позволяет вычислять значение a_n . Такую формулу называют *решением* рассматриваемого рекуррентного соотношения. Мы в данном параграфе покажем, как строить такие решения для случая линейного однородного рекуррентного соотношения с постоянными коэффициентами (29).

6.2. Прежде чем рассматривать общий случай уравнения (29), рассмотрим наиболее простой его вариант — линейное однородное рекуррентное соотношение первого порядка

$$a_{n+1} = b_1 \cdot a_n, \quad n = 0, 1, 2, \dots; \quad a_0 — \text{заданное число}. \quad (30)$$

6.2.1. Решение этого уравнения построить легко. Действительно,

$$a_1 = b_1 \cdot a_0; \quad a_2 = b_1 \cdot a_1 = b_1^2 \cdot a_0; \quad \dots \quad a_n = b_1^n \cdot a_0.$$

6.2.2. Предположим теперь, что нам кто-то сразу подсказал вид решения, а именно, что решение нашего уравнения (30) степенным образом зависит от n :

$$a_n = r^n \quad \text{для некоторого } r.$$

Воспользовавшись этой подсказкой, подставим это выражение в исходное рекуррентное соотношение (30). В результате получается равенство вида

$$r^{n+1} = b_1 \cdot r^n,$$

из которого сразу же следует, что $r = b_1$; при этом a_n оказывается равным b_1^n . Это означает, что при $n = 0$ число $a_0 = b_1^0 = 1$. Иными словами, $a_n = b_1^n$ есть решение исходной задачи (30) в частном случае $a_0 = 1$. Поэтому такое решение называется *частным решением* уравнения (30).

6.2.3. Теперь попытаемся, зная частное решение, построить общее решение задачи (30). Для этого заметим, что в силу однородности уравнения (30) любое его частное решение, умноженное на произвольную постоянную c , по-прежнему этому уравнению удовлетворяет:

$$c \cdot b_1^{n+1} \equiv c \cdot b_1 \cdot b_1^n.$$

Решение же вида $c \cdot b_1^n$ позволяет удовлетворить любому начальному условию. Действительно, подставляя его в начальное условие для уравнения (30), получим:

$$c \cdot b_1^0 = a_0 \quad \implies \quad c = a_0 \quad \implies \quad a_n = a_0 \cdot b_1^n.$$

По этой причине решение вида $c \cdot b_1^n$ называется *общим решением* уравнения (30).

6.3. Подведем предварительные итоги. Так как исходное уравнение (30) было очень простым, то нам удалось сразу построить его решение и выяснить, что оно степенным образом зависит от параметра n . Затем мы заметили, что если бы нам кто-то заранее подсказал степенной характер решения уравнения (30), то нам хватило бы этой информации для построения как общего, так и частного решения нашего рекуррентного уравнения.

Возникает вопрос: зачем же нам нужен для столь простого уравнения столь сложный алгоритм построения его решения? Оказывается, что этот алгоритм практически без изменений работает и для построения решения линейного однородного уравнения произвольного порядка.

6.3.1. Рассмотрим в качестве примера линейное однородное рекуррентное соотношение второго порядка

$$a_{n+2} = b_1 \cdot a_{n+1} + b_2 \cdot a_n, \quad n = 0, 1, 2, \dots, \quad a_0, a_1 \text{ — заданные числа}, \quad (31)$$

и попытаемся применить к этому уравнению алгоритм, описанный в конце предыдущего пункта.

Для этого предположим, что частное решение уравнения (31) по-прежнему степенным образом зависит от параметра n , то есть предположим, что существуют такие значения параметров a_0 и a_1 , при которых $a_n = r^n$. Подставляя это выражение в рекуррентное соотношение, получим

$$r^{n+2} = b_1 \cdot r^{n+1} + b_2 \cdot r^n \quad \implies \quad r^2 - b_1 r - b_2 = 0, \quad (32)$$

т.е. квадратное уравнение на r . Любое его решение r_0 дает нам некоторое частное решение уравнения (31). По этой причине данное уравнение называется *характеристическим уравнением* для рекуррентного соотношения (31).

6.3.2. Предположим, что характеристическое уравнение (32) имеет два различных вещественных корня r_1 и r_2 . Покажем, что в таком случае выражение

$$a_n = c_1 r_1^n + c_2 r_2^n, \quad (33)$$

где c_1 и c_2 — произвольные постоянные, является *общим решением* соотношения (31) в том смысле, что любое решение (31) с заданными начальными условиями в нем содержится.

Действительно, покажем вначале, что выражение (33) действительно удовлетворяет рекуррентному соотношению (31):

$$c_1 r_1^{n+2} + c_2 r_2^{n+2} = b_1 c_1 r_1^{n+1} + b_1 c_2 r_2^{n+1} + b_2 c_1 r_1^n + b_2 c_2 r_2^n \quad \iff$$

$$\iff c_1 r_1^n (r_1^2 - b_1 r_1 - b_2) + c_2 r_2^n (r_2^2 - b_1 r_2 - b_2) \equiv 0.$$

Покажем теперь, что это — действительно общее решение, т.е. что мы всегда можем подобрать константы c_1 и c_2 так, чтобы решение вида (33) удовлетворяло любым заданным начальным условиям. Для этого рассмотрим это выражение при $n = 0$ и $n = 1$:

$$\begin{aligned} a_0 &= c_1 r_1^0 + c_2 r_2^0 = c_1 + c_2, \\ a_1 &= c_1 r_1^1 + c_2 r_2^1 = c_1 r_1 + c_2 r_2. \end{aligned}$$

Эти выражения следует рассматривать как систему линейных уравнений для определения неизвестных постоянных c_1 и c_2 . Определитель этой системы

$$\begin{vmatrix} 1 & 1 \\ r_1 & r_2 \end{vmatrix} = r_2 - r_1 \neq 0,$$

поэтому система всегда имеет единственное решение.

6.3.3. Пожалуй, самым известным и важным примером рекуррентного соотношения (31) является соотношение вида

$$F_{n+2} = F_{n+1} + F_n, \quad n = 0, 1, 2, \dots, \quad F_0 = 0, \quad F_1 = 1,$$

определяющее так называемые числа Фибоначчи F_n . Характеристическое уравнение для этого рекуррентного уравнения имеет вид

$$r^2 = r + 1 \quad \implies \quad r_{1,2} = \frac{1 \pm \sqrt{5}}{2} \quad \implies \quad F_n = c_1 \left(\frac{1 + \sqrt{5}}{2} \right)^n + c_2 \left(\frac{1 - \sqrt{5}}{2} \right)^n.$$

Константы c_1 и c_2 определим из начальных условий:

$$\begin{aligned} F_0 = 0 &= c_1 + c_2 \\ F_1 = 1 &= c_1 \frac{1 + \sqrt{5}}{2} + c_2 \frac{1 - \sqrt{5}}{2} = c_1 \left[\frac{1 + \sqrt{5} - 1 + \sqrt{5}}{2} \right] = c_1 \sqrt{5} \quad \implies \\ &\implies \quad c_1 = +\frac{1}{\sqrt{5}}, \quad c_2 = -\frac{1}{\sqrt{5}}. \end{aligned}$$

Таким образом, окончательно получаем следующее явное выражение для чисел Фибоначчи:

$$F_n = \frac{1}{\sqrt{5}} \left(\frac{1 + \sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1 - \sqrt{5}}{2} \right)^n. \quad (34)$$

6.3.4. Предположим теперь, что характеристическое уравнение (32) имеет ровно один кратный корень $r_1 = r_2 =: \rho$. Покажем, что в этом случае общее решение уравнения (31) имеет вид

$$a_n = c_1 \rho^n + c_2 n \rho^n.$$

Мы уже показали, что частное решение вида ρ^n удовлетворяет уравнению (31). С учетом линейности и однородности этого уравнения нам осталось показать, что этому уравнению удовлетворяет и частное решение вида $n \rho^n$. Подставляя в уравнение (31) выражение $n \rho^n$, получим

$$(n + 2) \rho^{n+2} = b_1 (n + 1) \rho^{n+1} + b_2 n \rho^n \quad \implies \quad n (\rho^2 - b_1 \rho - b_2) + \rho (2 \rho - b_1) = 0.$$

Первое слагаемое в левой части этого выражения равно нулю в силу характеристического уравнения. Для того, чтобы понять, что равно нулю и последнее слагаемое, заметим, что в случае совпадающих корней $r_1 = r_2 = \rho$ имеем

$$\rho = \frac{b_1 \pm 0}{2} \quad \implies \quad 2\rho - b_1 = 0.$$

Осталось показать, что такой вид решения может удовлетворить любым начальным условиям. Подставив начальные условия в выражение для a_n , получим следующую систему двух линейных алгебраических уравнений относительно параметров c_1 и c_2 :

$$\begin{aligned} c_1 &= a_0, \\ \rho(c_1 + c_2) &= a_1. \end{aligned}$$

Понятно, что такая система имеет решение при любых a_0 , a_1 и $\rho \neq 0$. Случай же $\rho = 0$ тривиален — в этом случае $b_1 = b_2 = 0$, и потому $a_n = 0$ для любых n .

6.3.5. Наконец, давайте рассмотрим случай, когда уравнение (32) имеет два комплексно сопряжённых корня

$$r_1 = x + iy = \rho e^{i\vartheta}, \quad r_2 = x - iy = \rho e^{-i\vartheta}, \quad \rho, \vartheta \neq 0.$$

Покажем, что в таком случае общее решение линейного однородного рекуррентного соотношения с постоянными коэффициентами второго порядка (31) можно записать так:

$$a_n = \tilde{c}_1 \rho^n \cos(n\vartheta) + \tilde{c}_2 \rho^n \sin(n\vartheta). \quad (35)$$

Действительно, рассуждения, аналогичные проведенным для случая двух различных вещественных корней, показывают, что линейная комбинация частных решений вида

$$a_n = c_1 r_1^n + c_2 r_2^n$$

удовлетворяет рекуррентному соотношению (31) при любых c_1 и c_2 . Рассмотрим теперь систему линейных уравнений

$$\begin{aligned} a_0 &= c_1 + c_2, \\ a_1 &= c_1 \rho e^{i\vartheta} + c_2 \rho e^{-i\vartheta} = \rho(c_1 + c_2) \cos \vartheta + \rho i(c_1 - c_2) \sin \vartheta. \end{aligned} \quad (36)$$

Определитель этой системы по-прежнему отличен от нуля, так что мы с помощью общего решения по-прежнему можем удовлетворить любым начальным условиям. Из (36) также следует, что коэффициенты $\tilde{c}_1 = c_1 + c_2$ и $\tilde{c}_2 = i(c_1 - c_2)$ являются вещественными числами. Действительно, первое уравнение (36) гарантирует нам, что сумма $c_1 + c_2$ является вещественным числом. Но тогда вещественным числом обязано быть и выражение $i(c_1 - c_2)$.

Заметим теперь, что линейную комбинацию частных решений мы в рассматриваемом случае можем переписать так:

$$\begin{aligned} a_n &= c_1 r_1^n + c_2 r_2^n = c_1 \rho^n e^{i\vartheta n} + c_2 \rho^n e^{-i\vartheta n} = \\ &= (c_1 + c_2) \rho^n \cos(n\vartheta) + i(c_1 - c_2) \rho^n \sin(n\vartheta). \end{aligned}$$

Обозначив через $\tilde{c}_1 = c_1 + c_2$ и $\tilde{c}_2 = i(c_1 - c_2)$, мы получаем для a_n формулу (35).

6.3.6. Описанная выше техника решения линейных однородных рекуррентных соотношений с постоянными коэффициентами обобщается и на случай соотношений более высокого порядка. Именно, рассмотрим линейное однородное рекуррентное соотношение m -го порядка (29)

$$a_{n+m} = b_1 \cdot a_{n+m-1} + b_2 \cdot a_{n+m-2} + \dots + b_m \cdot a_n, \quad a_0, \dots, a_{m-1} \text{ — заданы.}$$

Предположим, что частное решение этого соотношения имеет вид $a_n = r^n$. Подставляя это выражение в исходное рекуррентное соотношение, получаем для параметра r следующее характеристическое уравнение:

$$r^m = b_1 \cdot r^{m-1} + b_2 \cdot r^{m-2} + \dots + b_m \cdot r.$$

Пусть r_1, \dots, r_k есть корни этого уравнения. Каждый такой корень дает свой вклад в общее решение рассматриваемого рекуррентного соотношения. В частности, если r_j есть корень кратности q_j , то вклад этого корня в общее решение задается выражением вида

$$c_{1,j} \cdot r_j^n + c_{2,j} \cdot n \cdot r_j^n + \dots + c_{q_j,j} \cdot n^{q_j-1} \cdot r_j^n.$$

Так, если характеристическое уравнение для некоторого рекуррентного соотношения шестого порядка имеет вид

$$(r - 1)^3 \cdot (r - 2)^2 \cdot (r - 3) = 0,$$

то общее решение такого соотношения записывается так:

$$a_n = c_1 + c_2 \cdot n + c_3 \cdot n^2 + c_4 \cdot 2^n + c_5 \cdot n \cdot 2^n + c_6 \cdot 3^n.$$

6.4. Вернемся к примеру о лягушках, рассмотренному вначале параграфа. В отличие от разобранных выше примеров, рекуррентное соотношение, полученное для этой задачи, было неоднородным. Постараемся понять, как нам решать такие соотношения.

6.4.1. Рекуррентное соотношение

$$a_{n+1} = 4 \cdot a_n - 100, \quad a_0 = 50$$

представляет собой частный случай линейного неоднородного рекуррентного соотношения с постоянными коэффициентами первого порядка. В общем случае такое соотношение имеет следующий вид:

$$a_{n+1} = b_1 \cdot a_n + u, \quad a_0 \text{ — заданное число.} \quad (37)$$

Как и в случае однородного рекуррентного соотношения (30), решение рекуррентного соотношения (37) довольно легко построить, последовательно выражая a_n через a_0 и u . Именно,

$$\begin{aligned} a_1 &= b_1 \cdot a_0 + u, \\ a_2 &= b_1 \cdot a_1 + u = b_1 \cdot (b_1 \cdot a_0 + u) + u = b_1^2 \cdot a_0 + b_1 \cdot u + u, \\ a_3 &= b_1 \cdot a_2 + u = b_1 \cdot (a_0 \cdot b_1^2 + b_1 \cdot u + u) + u = b_1^3 \cdot a_0 + b_1^2 \cdot u + b_1 \cdot u + u, \\ &\dots \\ a_n &= b_1^n \cdot a_0 + (b_1^{n-1} + b_1^{n-2} + \dots + b_1 + 1) \cdot u. \end{aligned}$$

В зависимости от значений параметра b_1 мы имеем два варианта развития событий. В случае $b_1 \neq 1$ мы выражение для a_n можем переписать так:

$$a_n = b_1^n \cdot a_0 + \frac{b_1^n - 1}{b_1 - 1} \cdot u = \left(a_0 - \frac{u}{1 - b_1} \right) \cdot b_1^n + \frac{u}{1 - b_1}.$$

В случае $b_1 = 1$ мы получаем

$$a_n = a_0 + n \cdot u.$$

6.4.2. Из приведенных выше рассуждений мы можем сделать два важных наблюдения. Первое из них заключается в том, что в обоих случаях решение a_n неоднородного рекуррентного соотношения может быть записано в виде

$$a_n = c_1 \cdot r^n + q_n,$$

где $c_1 \cdot r^n$ есть общее решение соответствующего (37) однородного рекуррентного соотношения

$$a_{n+1}^{(0)} = b_1 \cdot a_n^{(0)},$$

а q_n есть некоторое частное решение неоднородного рекуррентного соотношения (37). Оказывается, что этот факт справедлив и в общем случае. Именно, справедлива следующая

Теорема 6.3. Пусть q_n есть частное решение рекуррентного соотношения

$$a_{n+m} = b_1 a_{n+m-1} + b_2 a_{n+m-2} + \dots + b_{m-1} a_{n+1} + b_m a_n + u(n).$$

Тогда любое решение этого рекуррентного соотношения имеет вид $a_n = p_n + q_n$, где p_n есть общее решение соответствующего однородного рекуррентного соотношения

$$a_{n+m}^{(0)} = b_1 a_{n+m-1}^{(0)} + b_2 a_{n+m-2}^{(0)} + \dots + b_{m-1} a_{n+1}^{(0)} + b_m a_n^{(0)}.$$

Второе важное наблюдение состоит в том, что мы легко можем построить решение рекуррентного соотношения в случае, если нам кто-то подскажет конкретный вид как общего решения однородного соотношения, так и частного решения неоднородного соотношения. Предположим, например, что нам кто-то подсказал, что в случае $b_1 \neq 1$ частное решение (37) следует искать в виде $q(n) = c_0 \cdot u$, а общее решение соответствующего (37) однородного соотношения — в виде $a_n = c_1 \cdot r^n$. Подставляя в (37) вместо q_n выражение $c_0 \cdot u$, получаем

$$c_0 \cdot u = b_1 \cdot c_0 \cdot u + u \quad \implies \quad c_0 = \frac{1}{1 - b_1}.$$

Далее, подставляя вместо $a_n^{(0)}$ в соответствующее (37) однородное рекуррентное соотношение

$$a_{n+1}^{(0)} = b_1 \cdot a_n^{(0)}$$

выражение $c_1 \cdot r^n$, мы находим, что $r = b_1$. Осталось определить константу c_1 . Для этого воспользуемся начальным условием:

$$c_1 \cdot b_1^0 + \frac{u}{1 - b_1} = a_0 \quad \implies \quad c_1 = a_0 - \frac{u}{1 - b_1}.$$

Аналогичный алгоритм проходит и в случае $b_1 = 1$.

6.4.3. Мы уже знаем, в каком виде следует искать общее решение однородного рекуррентного соотношения. Осталось понять, как определять вид частного решения неоднородного рекуррентного соотношения. Ранее мы отмечали, что общее решение соответствующего однородного рекуррентного соотношения представляет собой сумму функций $f_j(n)$ вида

$$f_j(n) = (c_{1,j} + c_{2,j} \cdot n + c_{3,j} \cdot n^2 + \dots + c_{q_j,j} \cdot n^{q_j-1}) \cdot r_j^n.$$

Предположим, что неоднородность $u_n \equiv u(n)$ имеет вид

$$u(n) = (d_0 + d_1 \cdot n + \dots + d_l \cdot n^l) \cdot r^n =: P_l(n) \cdot r^n.$$

Тогда частное решение исходного неоднородного уравнения следует искать в виде

$$(c_0 + c_1 \cdot n + \dots + c_l \cdot n^l) \cdot n^q \cdot r^n,$$

где $q = 0$ в случае, если r отличен от корней r_j характеристического уравнения, и равно кратности корня r_j в случае, если r совпадает с одним из корней r_j . В случае, когда неоднородность представляет собой сумму описанных выше функций $u_t(n)$, частное решение рекуррентного соотношения можно искать в виде суммы частных решений неоднородных рекуррентных соотношений вида

$$a_{n+m} = b_1 \cdot a_{n+m-1} + b_2 \cdot a_{n+m-2} + \dots + b_m \cdot a_n + u_t(n).$$

6.4.4. Разберем несколько примеров решения неоднородных рекуррентных соотношений.

Пример 6.4. Построить общее решение неоднородного рекуррентного соотношения второго порядка

$$a_{n+2} = 4a_{n+1} - 4a_n + 3 \cdot 2^n.$$

Решение. Характеристическое уравнение для соответствующего однородного рекуррентного соотношения имеет следующий вид:

$$r^2 - 4r + 4 = 0 \quad \iff \quad (r - 2)^2 = 0.$$

Как следствие, общее решение соответствующего однородного рекуррентного соотношения имеет вид

$$c_1 \cdot 2^n + c_2 \cdot n \cdot 2^n.$$

Так как $r = 2$ есть корень кратности 2 характеристического уравнения, то частное решение исходного неоднородного рекуррентного соотношения следует искать в виде

$$c_0 \cdot n^2 \cdot 2^n.$$

Подставим это выражение в рекуррентное соотношение:

$$\begin{aligned} c_0 \cdot (n+2)^2 \cdot 2^{n+2} &= 4c_0 \cdot (n+1)^2 \cdot 2^{n+1} - 4c_0 \cdot n^2 \cdot 2^n + 3 \cdot 2^n && \iff \\ \iff & 4c_0(n^2 + 4n + 4) = 8c_0(n^2 + 2n + 1) - 4cn^2 + 3 && \iff 8c_0 = 3. \end{aligned}$$

Следовательно, $c_0 = 3/8$, так что частное решение заданного рекуррентного соотношения имеет вид

$$3 \cdot n^2 \cdot 2^{n-3}.$$

Пример 6.5. Построить общее решение следующего линейного неоднородного рекуррентного соотношения второго порядка:

$$a_{n+2} = 6a_{n+1} - 9a_n + 3^n \cdot n + 3^n \cdot \cos(\pi n/2).$$

Решение. Для построения общего решения соответствующего однородного рекуррентного соотношения запишем его характеристическое уравнение:

$$r^2 = 6r - 9 \quad \iff \quad (r - 3)^2 = 0.$$

Как видно, это уравнение имеет корень $r = 3$ кратности два. Следовательно, общее решение однородного рекуррентного соотношения записывается в виде

$$(d_1 + d_2 \cdot n) \cdot 3^n.$$

Неоднородная часть рекуррентного соотношения состоит из двух слагаемых:

$$u_1(n) = 3^n \cdot n \quad \text{и} \quad u_2(n) = 3^n \cdot \cos(\pi n/2).$$

Так как

$$\cos(\pi n/2) = \frac{e^{i\pi n/2} + e^{-i\pi n/2}}{2},$$

то выражение для $u_2(n)$ можно переписать так:

$$u_2(n) = \frac{1}{2} (r_1^n + r_2^n), \quad \text{где} \quad r_1 = 3 \cdot e^{i\pi/2}, \quad r_2 = 3 \cdot e^{-i\pi/2}.$$

Иными словами, неоднородная часть рекуррентного соотношения состоит из трех слагаемых вида $r^n \cdot P_i(n)$, и только в первом из них показатель совпадает с корнем характеристического уравнения. Как следствие, мы можем искать отдельно частное решение, связанное с первым слагаемым, и частное решение, связанное с оставшимися слагаемыми.

Рассмотрим вначале рекуррентное соотношение

$$a_{n+2} = 6a_{n+1} - 9a_n + 3^n \cdot n.$$

Частное его решение следует искать в виде $(c_1 + c_2 \cdot n) \cdot n^2 \cdot 3^n$. Подставляя это выражение в рекуррентное соотношение, получим равенство

$$54n \cdot c_2 + 54 \cdot c_2 + 18 \cdot c_1 = n.$$

Приравнявая коэффициенты при одинаковых степенях n , получаем значения констант $c_2 = 1/54$, $c_1 = -1/18$.

Теперь запишем рекуррентное соотношение

$$a_{n+2} = 6a_{n+1} - 9a_n + 3^n \cdot \cos(\pi n/2) \quad \iff \quad a_{n+2} = 6a_{n+1} - 9a_n + 3^n \cdot e^{i\pi n/2}/2 + 3^n \cdot e^{-i\pi n/2}/2.$$

Частное его решение можно искать в виде умноженной на 3^n линейной комбинации экспонент с показателями $\pm i\pi n/2$, либо, что более удобно, в виде линейной комбинации синуса и косинуса, умноженной на 3^n :

$$c_3 \cdot 3^n \cdot \cos(\pi n/2) + c_4 \cdot 3^n \cdot \sin(\pi n/2).$$

Подставляя это выражение в записанное выше рекуррентное соотношение, получаем равенство

$$18 \cdot c_3 \cdot \sin(\pi n/2) - (18 \cdot c_4 + 1) \cdot \cos(\pi n/2) = 0,$$

из которого следует, что $c_3 = 0$, $c_4 = -1/18$.

Окончательно получаем следующее общее решение исходного рекуррентного соотношения:

$$(d_1 + d_2 \cdot n) \cdot 3^n + (n - 3) \cdot n^2 \cdot 3^{n-3} / 2 - 3^{n-2} \cdot \sin(\pi n / 2) / 2.$$

6.5. Читатели, хорошо знакомые с математическим анализом, уже, видимо, заметили, насколько похожи методы решения линейных рекуррентных соотношений и методы решения линейных обыкновенных дифференциальных уравнений с постоянными коэффициентами.

6.5.1. Действительно, рассмотрим линейное однородное дифференциальное уравнение первого порядка вида

$$\frac{dy}{dx} = b_1 \cdot y(x), \quad y(x_0) = y_0.$$

Такое уравнение легко решается с помощью разделения переменных:

$$\begin{aligned} \frac{dy}{y} = b_1 \cdot dx &\iff d \ln(y) = b_1 \cdot dx &\iff \int_{y_0}^y d \ln(y) = b_1 \cdot \int_{x_0}^x dx &\iff \\ &\iff \ln(y) - \ln(y_0) = \ln \frac{y}{y_0} = b_1 \cdot (x - x_0) &\iff y = y_0 e^{b_1 \cdot (x - x_0)}. \end{aligned}$$

Как и в случае рекуррентных соотношений, мы могли бы решить это уравнение и без использования метода разделения переменных — для его решения достаточно знать характер решения такого дифференциального уравнения. Именно, предположим, что решение $y(x)$ экспоненциально зависит от x :

$$y(x) = e^{\lambda \cdot x}.$$

Подставим это выражение в исходное дифференциальное уравнение:

$$y'(x) = \lambda \cdot e^{\lambda \cdot x} = b_1 \cdot e^{\lambda \cdot x} \iff \lambda = b_1.$$

Заметим теперь, что решение $y(x) = e^{b_1 \cdot x}$ удовлетворяет начальным условиям $y(x_0) = e^{b_1 \cdot x_0}$. Для того, чтобы удовлетворить произвольным начальным условиям, достаточно воспользоваться однородностью данного уравнения и рассмотреть вместо частного общее решение вида $y(x) = c \cdot e^{b_1 \cdot x}$. Константа c легко находится из начальных условий:

$$y(x_0) = y_0 = c \cdot e^{b_1 \cdot x_0} \implies c = y_0 \cdot e^{-b_1 \cdot x_0} \implies y = y_0 \cdot e^{b_1(x - x_0)}.$$

6.5.2. Теперь рассмотрим линейное однородное дифференциальное уравнение второго порядка с постоянными коэффициентами

$$y''(x) = b_1 \cdot y'(x) + b_2 \cdot y(x), \quad y(0) = x_0, \quad y'(0) = x_1.$$

Предполагая, что характер решения этого уравнения остался тем же, что и в предыдущем случае, а именно, полагая, что $y(x) = e^{\lambda \cdot x}$, мы в результате подстановки этого выражения в исходное дифференциальное уравнение получаем абсолютно то же самое характеристическое уравнение на параметр λ , что и в случае линейного однородного рекуррентного соотношения второго порядка с постоянными коэффициентами:

$$y''(x) = \lambda^2 \cdot e^{\lambda \cdot x} = b_1 \cdot \lambda \cdot e^{\lambda \cdot x} + b_2 \cdot e^{\lambda \cdot x} \implies \lambda^2 - b_1 \cdot \lambda - b_2 = 0.$$

Если теперь это уравнение имеет пару различных вещественных корней λ_1 и λ_2 , то общее решение обыкновенного дифференциального уравнения в силу его линейности и однородности записывается в виде

$$y(x) = c_1 \cdot e^{\lambda_1 \cdot x} + c_2 \cdot e^{\lambda_2 \cdot x}.$$

В случае единственного вещественного корня λ характеристического уравнения решение по-прежнему ищется в виде

$$y(x) = c_1 \cdot e^{\lambda \cdot x} + c_2 \cdot x \cdot e^{\lambda \cdot x}.$$

Наконец, в случае комплексных корней $\lambda_{1,2} = \alpha \pm i \cdot \beta$ решение обыкновенного дифференциального уравнения можно искать в виде

$$y(x) = c_1 \cdot e^{\alpha \cdot x} \cos(\beta \cdot x) + c_2 \cdot e^{\alpha \cdot x} \sin(\beta \cdot x).$$

6.5.3. Отмеченная аналогия наводит нас на мысль о существовании взаимно-однозначного отображения, позволяющего сопоставить линейному однородному рекуррентному соотношению с постоянными коэффициентами аналогичное линейное однородное дифференциальное уравнение с постоянными коэффициентами. Оказывается, что такое отображение построить достаточно просто. Рассмотрим для определенности линейное однородное рекуррентное соотношение с постоянными коэффициентами второго порядка:

$$a_{n+2} = b_1 \cdot a_{n+1} + b_2 \cdot a_n, \quad a_0, a_1 \text{ — заданные числа.}$$

Умножим это уравнение на $x^n/n!$ и просуммируем по n от 0 до $+\infty$:

$$\sum_{n=0}^{+\infty} a_{n+2} \frac{x^n}{n!} = b_1 \sum_{n=0}^{+\infty} a_{n+1} \frac{x^n}{n!} + b_2 \sum_{n=0}^{+\infty} a_n \frac{x^n}{n!}.$$

Предположим теперь, что ряд

$$a_0 + a_1 \frac{x^1}{1!} + a_2 \frac{x^2}{2!} + \dots + a_n \frac{x^n}{n!} + \dots = \sum_{n=0}^{+\infty} a_n \frac{x^n}{n!}$$

представляет собой ряд Тейлора некоторой функции $y(x)$, построенный в окрестности точки $x = 0$:

$$y(x) = \sum_{n=0}^{+\infty} a_n \frac{x^n}{n!}, \quad a_n = y^{(n)}(0).$$

Тогда

$$y'(x) = a_1 + a_2 \frac{x^1}{1!} + a_3 \frac{x^2}{2!} + \dots + a_n \frac{x^{n-1}}{(n-1)!} + a_{n+1} \frac{x^n}{n!} + \dots = \sum_{n=0}^{+\infty} a_{n+1} \frac{x^n}{n!},$$

$$y''(x) = a_2 + a_3 \frac{x^1}{1!} + a_4 \frac{x^2}{2!} + \dots + a_n \frac{x^{n-2}}{(n-2)!} + a_{n+1} \frac{x^{n-1}}{(n-1)!} + a_{n+2} \frac{x^n}{n!} + \dots = \sum_{n=0}^{+\infty} a_{n+2} \frac{x^n}{n!},$$

так что исходное рекуррентное соотношение превращается в обыкновенное дифференциальное уравнение вида

$$y''(x) = b_1 \cdot y'(x) + b_2 \cdot y(x)$$

с начальными условиями $y(0) = a_0$, $y'(0) = a_1$. Обратное, раскладывая функцию $y(x)$ в ряд Тейлора в окрестности точки $x = 0$, подставляя это разложение в обыкновенное дифференциальное уравнение и приравнивая коэффициенты при одинаковых степенях $x^n/n!$, мы из исходного обыкновенного дифференциального уравнения получаем рекуррентное соотношение на коэффициенты $a_n = y^{(n)}(0)$.

6.5.4. Полученное нами взаимно-однозначное преобразование полезно как для решения обыкновенных дифференциальных уравнений, так и для решения рекуррентных соотношений. Так, с вычислительной точки зрения иногда имеет смысл переходить от обыкновенного дифференциального уравнения к соответствующему рекуррентному соотношению. Быстрое вычисление чисел a_n позволяет эффективно вычислять коэффициенты в разложении искомой функции в ряд Тейлора, а следовательно, и значение этой функции в произвольной точке x . Напротив, методы, разработанные для решения обыкновенных дифференциальных уравнений, часто оказываются полезными и для решения аналогичных рекуррентных соотношений.

Элементы теории графов

7 Основные понятия и определения теории графов

7.1. Начнем данный параграф с определения неориентированных и ориентированных графов.

7.1.1. Формальное и достаточно общее определение неориентированного графа таково.

Определение 7.1. Неориентированным графом G называется тройка

$$G = (V, E, I),$$

состоящая из

(1) (конечного) множества вершин $V = V(G)$, например,

$$V = \{1, 2, 3, 4\},$$

(2) (конечного) множества ребер $E = E(G)$, например,

$$E = \{a, b, c, d, e, f, g, h\},$$

(3) а также отображения $I: E \rightarrow V_2$, сопоставляющего любому ребру $e \in E$ неупорядоченную пару вершин $\{x, y\} \in V_2$, которую это ребро соединяет.

Вершины x и y называются *концевыми вершинами* ребра e . При этом говорят, что ребро e *инцидентно* своим концевым вершинам. Про любую из двух вершин x, y , в свою очередь, говорят, что она инцидентна ребру e .

В принципе, возможен случай $x = y$. Ребро $e \in E$, соответствующее паре $\{x, x\}$, называется обычно *петлей*. Кроме того, в общем случае у нас могут быть несколько различных ребер, соединяющих одну и ту же пару вершин $\{x, y\}$. Такие ребра называют кратными ребрами и говорят, что они образуют *мультиребро* графа G . Если пара вершин $\{x, y\}$ соединена между собой единственным ребром, то такое ребро иногда называют простым ребром графа.

Пример 7.2. Зададим отображение I в виде следующей таблицы:

E	a	b	c	d	e	f	g	h
V_2	$\{1, 3\}$	$\{2, 4\}$	$\{1, 3\}$	$\{3, 4\}$	$\{3, 4\}$	$\{1, 2\}$	$\{3, 4\}$	$\{2, 2\}$

Ей соответствует граф G , изображенный на рис.8. В этом графе ребра f и b являются простыми, кратные ребра a и c образуют мультиребро, соединяющее вершины 1 и 3, а ребро h представляет собой петлю.

7.1.2. С понятием инцидентности тесно связано важное понятие степени вершины графа G .

Определение 7.3. В неориентированном графе G *степенью* $\deg(x)$ или *валентностью* вершины x называется количество ребер, инцидентных x . При этом считается, что петля дает вклад, равный двум, в степень любой вершины.

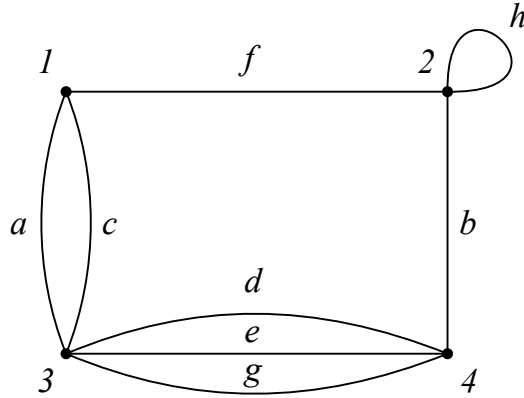


Рис. 8: Пример графа на четырех вершинах

Так, вершина 1 на рис.8 имеет степень, равную трем, а вершина 2 — степень, равную четырем. Следующее утверждение часто называют первой теоремой теории графов.

Теорема 7.4. *В неориентированном графе G сумма степеней всех вершин равна удвоенному количеству всех ребер графа:*

$$\sum_{x \in V(G)} \deg(x) = 2|E(G)|. \quad (38)$$

Доказательство практически очевидно — любое ребро дает вклад, равный двум, в стоящую слева сумму.

Следствие 7.5. *Количество вершин в графе G , имеющих нечетную степень, четно.*

7.1.3. Несмотря на достаточную очевидность доказательства теоремы 7.4, оно использует чрезвычайно важный прием, очень часто встречающийся и в теории графов, и в комбинаторике — так называемый двойной подсчет (double counting).

Неформально его можно себе представлять следующим образом. Предположим, что у нас имеется какое-то множество C элементарных объектов C_i , любой из которых мы можем рассматривать как составную часть двух различных, более сложно устроенных объектов A_j и B_k . Тогда количество $c = |C|$ всех элементарных объектов C_i мы всегда сможем сосчитать двумя способами. Именно, для любого сложного объекта первого типа A_j мы можем сосчитать количество a_j элементарных объектов, из которых состоит A_j , а затем просуммировать полученные числа a_j по j от 1 до n , где n есть количество всех сложных объектов первого типа. С другой стороны, мы можем для любого B_k сосчитать количество b_k элементарных объектов, из которых состоит сложный объект второго типа B_k , а затем просуммировать b_k по всем k от 1 до m , где m — количество объектов второго типа. Так как общее количество c всех элементарных объектов C_i , от способа подсчета не зависит, то мы в итоге получаем равенство вида

$$\sum_{j=1}^n a_j = c = \sum_{k=1}^m b_k, \quad (39)$$

которое и отражает некоторый конечный результат процедуры double counting.

При доказательстве первой теоремы теории графов мы в качестве элементарного объекта C_i можем взять так называемое *полуребро* графа G (semiedge или dart), то есть половинку любого

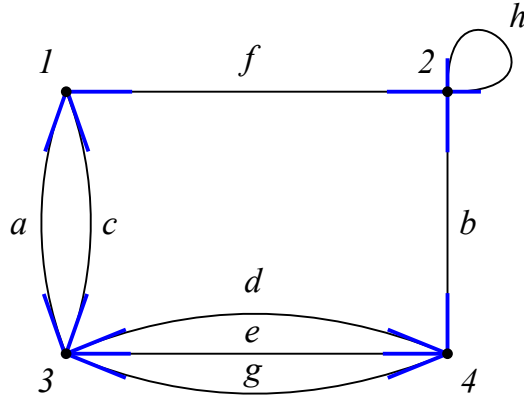


Рис. 9

ребра, инцидентного вершине x (см.рис.9, на котором полуребра изображены синим цветом). Каждое ребро состоит ровно из двух полуребер, поэтому количество s всех полуребер равно удвоенному количеству $2m$, $m = |E(G)|$, всех ребер. С другой стороны, любую вершину x_j мы можем представить в виде “ежа”, то есть вершины вместе с инцидентными ей полуребрами. Количество a_j таких полуребер совпадает со степенью $\deg(x_j)$ вершины x_j . Суммируя эти числа a_j по всем $j = 1, \dots, n$, где n — количество всех вершин в графе, мы вновь получаем общее количество полуребер s . Формула (38) при этом есть не что иное, как соотношение (39), переписанное для рассматриваемого частного случая.

7.1.4. Описанную выше процедуру double counting можно формализовать с помощью так называемой матрицы инцидентности M_i . Именно, рассмотрим матрицу M_i размерами $n \times m$, строки которой отвечают объектам A_j , $j = 1, \dots, n$ первого типа, а столбцы — объектам B_k , $k = 1, \dots, m$ второго типа. Назовем объекты A_j и B_k *инцидентными* друг другу, если у нас существует хотя бы один элементарный объект, являющийся частью как A_j , так и B_k . Любой элемент $m_{j,k}$ матрицы M_i равен количеству элементарных объектов, входящих как в A_j , так и в B_k . При таком подходе левая часть равенства (39) подсчитывает количество элементарных объектов суммированием по строкам матрицы M_i , а правая часть (39) — суммированием по столбцам.

В случае неориентированного графа G строки матрицы M_i инцидентности графа G отвечают вершинам этого графа, а столбцы — его ребрам. Элемент $m_{j,k}$ матрицы M_i равен нулю, если вершина x_j и ребро e_k не инцидентны друг другу, $m_{j,k} = 2$ в случае, если e_k есть инцидентная вершине x_j петля, и $m_{j,k} = 1$ в остальных случаях.

Рассмотрим, к примеру, граф G , показанный на рисунках 8 и 9. Для такого графа матрица инцидентности имеет следующий вид:

$$M_i = \left(\begin{array}{c|cccccccc} V \setminus E & a & b & c & d & e & f & g & h \\ \hline 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 2 \\ 3 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 4 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{array} \right).$$

Видно, что сумма элементов в любом столбце этой матрицы равна двум, а сумма элементов в любой строке совпадает со степенью вершины x этого графа. Суммирование по строкам и суммирование по столбцам снова приводит нас к равенству (38).

7.2. Чаще всего на практике встречаются так называемые простые графы, к описанию которых

мы и перейдем.

7.2.1. Начнем с определения простого графа.

Определение 7.6. Граф G называется *простым*, если он не содержит петель и мультиребер. Граф, не являющийся простым, часто называют *мультиграфом*.



Рис. 10: Примеры простых графов

В качестве примера рассмотрим изображенный на рис.10,а простой граф G , построенный на четырех вершинах и имеющий два ребра. Видно, что в таком графе мы можем не вводить какое-то дополнительное специальное обозначение для ребер — любое ребро однозначно задается парой вершин, которые оно соединяет. Как следствие, для описания простого графа G нам достаточно задать множество V его вершин, а также множество E его ребер в виде некоторого подмножества множества всевозможных неупорядоченных пар множества вершин. Так, для графа G на рис.10,а мы имеем

$$V = \{1, 2, 3, 4\}, \quad E = \{\{1, 2\}, \{1, 3\}\}.$$

Итак, с формальной точки зрения любой простой неориентированный граф G , построенный на n вершинах, можно рассматривать как некоторое подмножество множества $V^{(2)}$ всех двухэлементных подмножеств множества $V(G)$ его вершин:

$$G \subseteq V^{(2)}.$$

Если же говорить менее формально, то простой граф — это граф, построенный на n вершинах, любая пара которых может быть соединена или не соединена ребром.

Сразу заметим, что матрица M_i инцидентности простого графа состоит только из нулей и единиц.

7.2.2. Разберем некоторые важные примеры простых графов. В качестве первого примера рассмотрим так называемый *полный* граф K_n (см.рис.11,а, на котором показан полный граф K_5), в котором любая вершина соединена со всеми оставшимися вершинами графа. Такой граф отвечает всему множеству $V^{(2)}$ двухэлементных подмножеств множества n вершин. Пустому подмножеству множества $V^{(2)}$ отвечает так называемый *пустой* граф, то есть граф, состоящий из n изолированных вершин (рис.11,б), то есть вершин, степени которых равны нулю.

Пустой граф является дополнением к полному графу K_n (и обозначается он как \bar{K}_n) в смысле следующего определения.

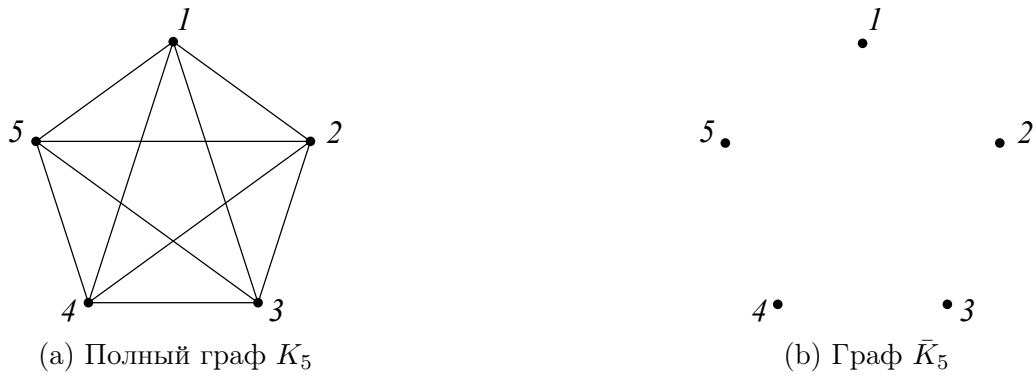


Рис. 11: Полный граф и пустой граф

Определение 7.7. Граф \bar{G} называется *дополнением* к графу G , если множества вершин этих двух графов совпадают, а множество ребер графа \bar{G} дополняет множество ребер $E(G)$ исходного графа G до множества ребер полного графа K_n .

На рис.10 в качестве примера приведены два графа, построенные на четырех вершинах — граф G и граф \bar{G} . Граф \bar{G} из графа G можно получить, например, так: взять полный граф K_4 , построенный на том же количестве вершин, что и графы G и \bar{G} , и удалить из него ребра, принадлежащие графу G . Полученный в результате этой операции граф \bar{G} и будет являться дополнением к графу G в смысле данного выше определения.

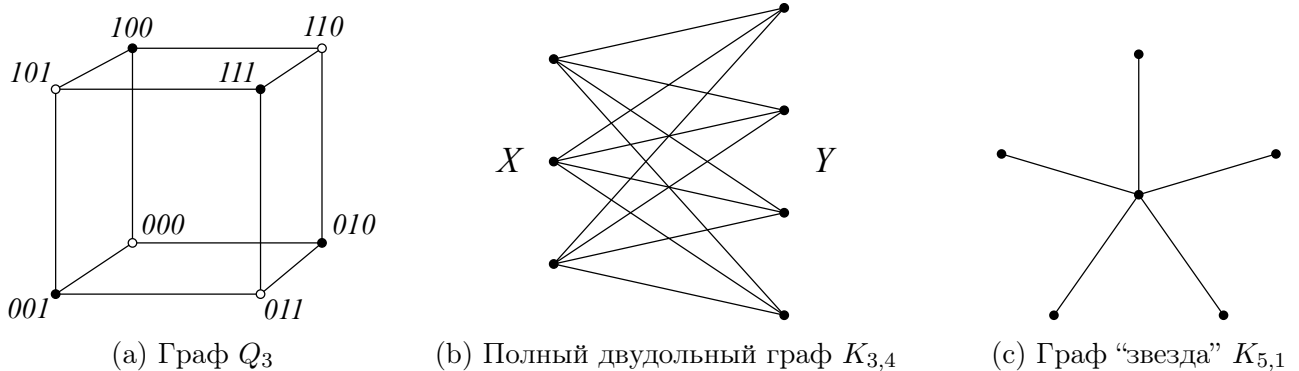


Рис. 12: Двудольные графы

7.2.3. При изучении теории графов нам часто будут встречаться еще несколько важных подклассов простых графов.

Определение 7.8. Граф G называется *двудольным* (рис.12), если множество $V(G)$ его вершин можно разбить на два блока X и Y так, что концы x и y любого ребра $e = \{x, y\} \in E(G)$ лежат в разных блоках этого разбиения. Обозначается двудольный граф с разбиением (X, Y) через $G[X, Y]$.

Простой граф $G[X, Y]$, в котором любая вершина из X соединена ребром с каждой вершиной из блока Y и наоборот, называется *полным двудольным графом* (рис.12,b). Часто такой граф обозначается через $K_{n,m}$, где $n = |X|$, а $m = |Y|$. Граф $K_{n,1}$ называется *звездой* (рис.12,c).

Еще два важных подкласса простого графа, называемые путем P_n и циклом C_n , показаны на рис.13,a и рис.13,b. Количество ребер в графах P_n и C_n характеризует длину таких графов.

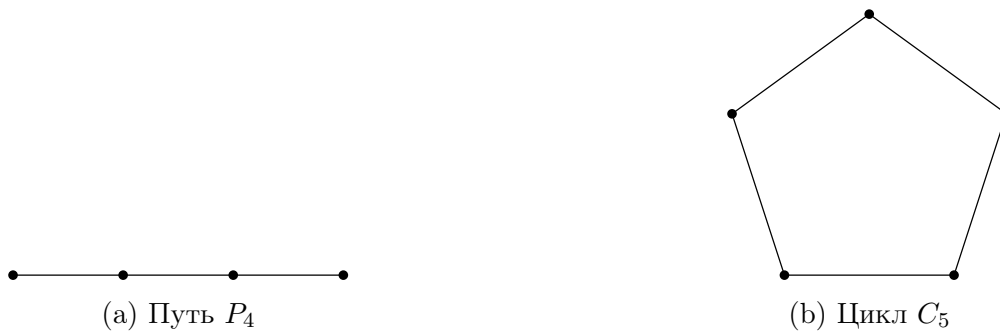


Рис. 13: Путь и цикл

Также часто нам будет встречаться граф W_n , называемый колесом (см.рис.14,а). Индекс n в обозначении W_n такого графа обозначает либо количество вершин в его внешнем цикле, либо количество всех вершин. Мы будем считать, что n — это количество вершин во внешнем цикле графа W_n .



Рис. 14: Колесо и граф Петерсена

Важный класс простых подграфов представляют собой так называемые *регулярные графы*, то есть графы, у которых все вершины имеют одну и ту же степень k . Также их называют k -регулярными графами или регулярными графами степени k . В случае $k = 3$ такие графы называются кубическими. На рис.14,б показан пример кубического графа, который достаточно часто будет встречаться у нас в приложениях — так называемый граф Петерсена (Petersen graph). Графы C_n и K_n также являются регулярными графами степени 2 и $n - 1$ соответственно.

Наконец, имеется важный класс графов, являющихся одновременно k -регулярными и двудольными — так называемые k -кубы Q_k . Вершины графа Q_k можно пометить бинарными строками длины k (см.рис.12,а). Ребра же в таком графе проводятся только между теми вершинами, бинарные последовательности которых отличаются только в одной из k позиций. Пример графа Q_3 показан на рис.12,а.

7.3. Наряду с неориентированными, в теории графов также изучаются и так называемые ориентированные графы (или орграфы).

7.3.1. Начнем с мы вновь с достаточно формального и общего определения ориентированного мультиграфа D .

Определение 7.9. Если в тройке

$$D = (V, E, I),$$

в которой V есть множество вершин, а E — множество ребер, отображение I ставит в соответствие любому ребру e упорядоченную пару вершин $(x, y) \in V \times V$, то такая тройка называется *ориентированным графом* (или *орграфом*). В таком случае говорят, что ребро e *выходит* из вершины x и *входит* в вершину y . На рисунке такое ребро помечается стрелкой, указывающей направление данного ребра.

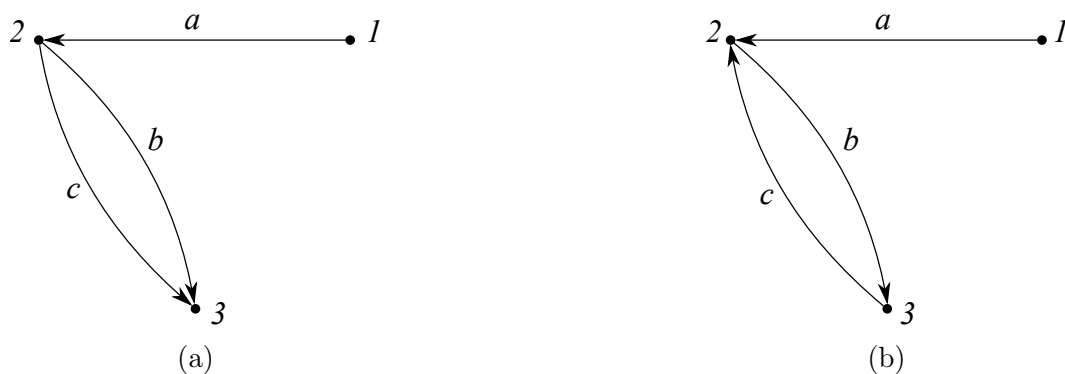


Рис. 15: Примеры ориентированных графов

В качестве примера на рис.15,а показан ориентированный граф D , построенный на множестве вершин $V = \{1, 2, 3\}$ и имеющий три ребра. Ребро a выходит из вершины 1 и входит в вершину 2 (то есть отвечает упорядоченной паре $(1, 2)$), а ребра b и c исходят из вершины 2 и приходят в вершину 3 (то есть им соответствует упорядоченная пара $(2, 3)$ вершин графа D). Таким образом, орграф D на рис. задается тройкой (V, E, I) , в которой множество V вершин и множество E ребер имеют вид

$$V = \{1, 2, 3\}, \quad E = \{a, b, c\},$$

а отображение $I: E \rightarrow V \times V$ задается таблицей вида

E	a	b	c
$V \times V$	$(1, 2)$	$(2, 3)$	$(2, 3)$

7.3.2. Для беспетлевых орграфов можно ввести понятие матрицы инцидентности $M_i(D)$. Каждая строка такой матрицы отвечает некоторой вершине орграфа, каждый столбец — некоторому ребру. В случае, если из вершины x_i исходит ребро e_j , элемент $m_{i,j}$ матрицы инцидентности считается равным $+1$, а в случае, если в вершину x_i входит ребро e_j , то $m_{i,j}$ полагают равным -1 .

Так, для изображенного на рис.15 орграфа D матрица инцидентности $M_i(D)$ имеет следующий вид:

$$M_i = \left(\begin{array}{c|ccc} V \setminus E & a & b & c \\ \hline 1 & 1 & 0 & 0 \\ 2 & -1 & 1 & 1 \\ 3 & 0 & -1 & -1 \end{array} \right).$$

7.3.3. В орграфе различают исходящую ($\text{outdeg}(x)$) и входящую ($\text{indeg}(x)$) степень любой вершины $x \in V(D)$. Так, в графе, изображенном на рис.15,а, вершина 2 имеет входящую степень, равную единице, и исходящую степень, равную двойке.

Так как любое ориентированное ребро в орграфе D вносит вклад, равный единице, в сумму всех исходящих степеней вершин орграфа D , а также вклад, равный единице, в сумму всех входящих степеней вершин орграфа D , то для орграфа справедливо равенство вида

$$\sum_{x \in V(D)} \text{indeg}(x) = |E(D)| = \sum_{x \in V(D)} \text{outdeg}(x), \tag{40}$$

являющееся аналогом первой теоремы теории графов (7.4).

7.3.4. Как и для неориентированного графа, важным частным случаем орграфа является простой орграф.

Определение 7.10. Орграф D называется *простым*, если он не содержит петель, а также кратных *упорядоченных* ребер, то есть ребер, отвечающих одинаковым упорядоченным парам вершин.

Упорядоченность вершин в этом определении важна. Так, граф из рис.15,а простым не является — в нем ребра b и c отвечают одной и той же упорядоченной паре $(2, 3)$ вершин. Изображенный же на рис.15,б граф D' является простым. Несмотря на то, что в этом графе по-прежнему имеются два ребра, соединяющих вершины 2 и 3, направлены эти ребра в разные стороны. Иными словами, эти ребра отвечают различным упорядоченным парам вершин — ребро b отвечает упорядоченной паре $(2, 3)$, а ребро c — упорядоченной паре $(3, 2)$.

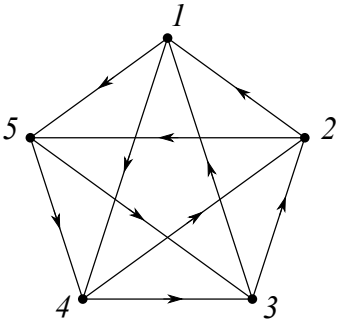


Рис. 16

7.3.5. Важным частным случаем простого орграфа, довольно часто встречающимся на практике, является турнир T — орграф, полученный ориентацией ребер полного графа K_n (рис.16).

Определение 7.11. Ориентацией графа G называется орграф D , полученный из G ориентацией каждого из ребер G , то есть заменой любого ребра $\{x, y\} \in E(G)$ неориентированного графа либо на ребро $(x, y) \in E(D)$, либо на ребро $(y, x) \in E(D)$.

Определение 7.12. Турниром T называется орграф, полученный ориентацией полного графа K_n .

Любой орграф T моделирует результат так называемого кругового турнира (round-robin tournament), в котором каждый игрок встречается с каждым другим участником ровно один раз. Считается при этом, что результатом встречи может быть победа или поражение, ничьи правилами турнира не предусматриваются. Ребро (x, y) в турнире направлено от выигравшего игрока x к проигравшему y . Как следствие, исходящая степень $\text{outdeg}(x)$ произвольного турнира T определяет количество очков (score), набранных в турнире соответствующей командой.

7.4. Следующим важным понятием в теории графов является понятие смежности вершин.

7.4.1. Начнем с определения смежных вершин в неориентированном графе G .

Определение 7.13. Говорят, что в неориентированном графе G вершина y смежна с вершиной x , если в этом графе существует ребро $\{x, y\}$.

На множестве вершин V смежность задает некоторое отношение. Для неориентированного графа G это отношение является симметричным: если вершина x смежна с вершиной y , то и вершина y смежна с вершиной x .

7.4.2. Для ориентированного графа ситуация несколько сложнее.

Определение 7.14. Говорят, что в ориентированном графе D вершина y смежна с вершиной x , если в этом графе существует ребро (x, y) , исходящее из вершины x и входящее в вершину y . (см.рис.15,а). Как следствие, все смежные с x вершины — это вершины, в которые ведут ребра из вершины x .

Для изображенного на рис.15,а орграфа G вершина 2 смежна с вершиной 1.

Сразу заметим, что если существует ребро (x, y) и не существует ребро (y, x) , то вершина y смежна с вершиной x , а вот x вершиной, смежной с y , уже не является. Так, для изображенного на рис.15,а орграфа вершина 1 смежной с вершиной 2 не является. Вершина x будет смежной с y только в случае, когда в орграфе D существует ребро (y, x) (см. вершины 2 и 3 на рис.15,б). Иными словами, в ориентированном графе D отношение смежности на множестве V вершин симметричным не является.

7.5. Для хранения графа в памяти компьютера, как правило, используются две структуры, тесно связанные с понятием смежности — матрица смежности и список смежности.

7.5.1. Матрица смежности — это матрица M_a размерами $n \times n$, любой элемент a_{ij} которой описывает количество ребер, соединяющих вершины i и j . Так, для примера 7.2 соответствующая графу G матрица смежности имеет следующий вид:

$$M_a = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 1 \\ 2 & 0 & 0 & 3 \\ 0 & 1 & 3 & 0 \end{bmatrix}$$

В случае ориентированных графов элементу a_{ij} отвечает количество ребер, идущих из вершины i в вершину j . Так, для ориентированных графов, показанных на рис.15, матрицы смежности равны

$$M_a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{и} \quad M_a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Заметим, что для неориентированного графа матрица смежности всегда симметрична. Как следствие, все собственные значения этой матрицы являются вещественными числами.

В случае простого графа или орграфа все диагональные элементы $a_{ii} = 0$, так что сумма собственных значений такой матрицы, совпадающая со следом матрицы M_a , также равняется

нулю. Элементы, не лежащие на диагонали, равны единице в случае, если существует ребро, идущее из вершины i в вершину j , и нулю в случае, если такового ребра не существует.

7.5.2. Список смежности — это линейный массив L_a размера n , каждый элемент a_i которого содержит список (мультимножество) вершин, смежных с вершиной i . Для примера 7.2 соответствующий список имеет следующий вид:

- 1 (смежные с 1 вершины): 2, 3, 3;
- 2 (смежные с 2 вершины): 1, 2, 4;
- 3 (смежные с 3 вершины): 1, 1, 4, 4, 4;
- 4 (смежные с 4 вершины): 2, 3, 3, 3.

Для ориентированного графа напротив каждой вершины с номером i стоят вершины (возможно, повторяющиеся), в которые идут ребра из i . Как следствие, для показанного на рис.15,а орграфа список смежности записывается так:

- 1 (смежные с 1 вершины): 2;
- 2 (смежные с 2 вершины): 3, 3;
- 3 (смежные с 3 вершины): \emptyset .

8 Маршруты, пути, циклы в графе. Связные графы и орграфы

8.1. Следующая важная группа понятий теории графов связана с обходом графа вдоль некоторой последовательности его вершин и ребер. В этой связи нам понадобятся такие понятия, как маршруты и пути в графе, а также некоторые другие связанные с ними характеристики графа.

8.1.1. Начнем с определения маршрута в мультиграфе G .

Определение 8.1. *Маршрутом* (walk) в графе G из вершины x_0 в вершину x_k называется чередующаяся последовательность

$$W := x_0, e_1, x_1, e_2, x_2, \dots, x_{k-1}, e_k, x_k$$

вершин $x_i \in V$ и ребер $e_i \in E$, соединяющих вершины x_{i-1} и x_i . И вершины, и ребра в такой последовательности могут повторяться. Количество ребер в маршруте W называется *длиной* k этого маршрута.

В случае простого графа любой маршрут W полностью определяется последовательностью

$$x_0, x_1, x_2, \dots, x_{k-1}, x_k,$$

вершин $x_i \in V(G)$, любые два последовательных элемента x_{i-1}, x_i которой являются смежными вершинами (т.е. соединены между собой ребром $e_i = \{x_{i-1}, x_i\} \in E(G)$).

Вершины x_0 и x_k называют часто начальной и конечной вершинами маршрута W , а остальные вершины — внутренними его вершинами. Говорят также, что вершины x_0 и x_k связаны маршрутом W , а сам маршрут называют x_0x_k -маршрутом.

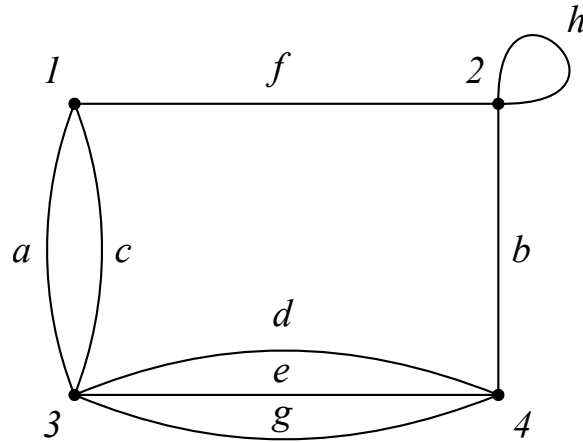


Рис. 17

Определение 8.2. Если все ребра e_1, \dots, e_k в маршруте различны, то такой маршрут называется *путем* (в английской литературе — trail) из вершины x_0 в вершину x_k . Если также и все вершины в данном пути различны, то такой путь называется *простым* (в английской литературе — path).

В качестве примера рассмотрим разные маршруты в графе, показанном на рис.17. Маршрут

$$1, a, 3, d, 4, e, 3, d, 4, b, 2, h, 2$$

путем не является — в нем повторяется ребро d . Маршрут

$$1, a, 3, d, 4, e, 3, g, 4, b, 2, h, 2$$

представляет собой путь (trail), который не является простым — в нем повторяются вершины 3, 4, 2. Наконец, маршрут

$$1, a, 3, e, 4, b, 2$$

является простым путем (path) из вершины 1 в вершину 2.

8.1.2. Понятие пути позволяет нам дать крайне важное определение связного графа.

Определение 8.3. Если вершины $x, y \in V$ графа G соединены хотя бы одним путем, то такие вершины называются *связанными*.

Несложно проверить, что связанность задает на множестве V вершин графа G отношение эквивалентности. Это отношение делит граф на классы эквивалентности, называемые *компонентами связности графа*.

Определение 8.4. В случае, когда в графе G существует лишь одна компонента связности, то есть в случае, когда любые две вершины x, y графа соединены хотя бы одним путем, граф называется *связным*. В противном случае граф называется *несвязным*.

8.1.3. Перейдем теперь к понятиям, характеризующим метрические свойства графа.

Определение 8.5. *Расстоянием* $d(x, y)$ между двумя связанными вершинами $x, y \in V(G)$ называется длина наименьшего пути между ними.

Понятно, что такой путь обязательно является простым. В случае, когда вершины не являются связанными, полагают по определению, что $d(x, y) = \infty$.

Определение 8.6. *Диаметром* графа называется максимальное расстояние между его вершинами:

$$\text{diam}(G) := \max_{x, y \in V(G)} d(x, y).$$

В случае несвязного графа считается, что $\text{diam}(G) = \infty$.

Определение 8.7. *Эксцентриситетом* $\varepsilon(x)$ вершины $x \in V(G)$ называется максимальное расстояние от x до любой другой вершины графа G :

$$\varepsilon(x) := \max_{y \in V(G)} d(x, y).$$

Определение 8.8. *Радиусом* $r(G)$ графа G называется минимальный из эксцентриситетов вершин графа G . Вершины, на которых этот минимум достигается, называются *центральными вершинами* графа G . Множество всех центральных вершин называется *центром* графа.

Некоторые несложные факты, связанные с введенными выше понятиями, приведены в упражнениях ?? и ??.

8.1.4. Следующая серия определений связана с замкнутыми маршрутами в графах, то есть маршрутами, в которых начальная x_0 и конечная x_k вершины совпадают.

Определение 8.9. Замкнутым путем (closed trail) или составным циклом (circuit) в графе G называется путь, в котором $x_0 = x_k$. Путь, в котором совпадают только начальная и конечная вершины, называется простым циклом (cycle). Часто слово “простой” опускают, понимая под циклом замкнутый простой путь.

Так, в графе G , показанном на рис.17, путь $1, a, 3, e, 4, d, 3, c, 1$ является составным циклом, а путь вида $1, a, 3, e, 4, b, 2, f, 1$ — простым циклом. Замкнутый маршрут вида $1, a, 3, e, 4, e, 3, c, 1$ составным циклом не является — в нем повторяется ребро e .

В простом графе невозможны циклы длины меньшей, чем три. Циклы длины три часто называются треугольниками. Графы, в которых такие циклы отсутствуют, называются графами, свободными от треугольников (triangle-free graphs).

Определение 8.10. Обхватом графа (girth) называется длина наименьшего цикла в нем. Если в графе циклы отсутствуют, то обхват такого графа считают равным бесконечности.

С учетом данного определения простые графы, свободные от треугольников — это графы, обхват которых больше трех.

В случае мультиграфов возможны и циклы длины 2, и циклы длины 1. Циклы длины 2 — это циклы типа $3, d, 4, e, 3$ на рис.17, появляющиеся при обходе двух ребер одного и того же мультиребра. Циклы длины 1 — это петли графа G . Так, петлю h на рис.17 можно рассматривать как цикл $2, h, 2$ длины 1.

Изолированная вершина по определению является замкнутым путем длины 0, однако циклом она не является.

8.1.5. В предыдущем параграфе мы ввели важный подкласс простых графов — так называемые двудольные графы $G[X, Y]$. Оказывается, что такие графы допускают характеристику в терминах циклов.

Теорема 8.11 (König). *Граф G является двудольным тогда и только тогда, когда в нем отсутствуют циклы нечетной длины (так называемые нечетные циклы).*

Доказательство. То, что в любом двудольном графе нечетные циклы отсутствуют, достаточно очевидно. Действительно, для того, чтобы, выйдя из произвольной вершины x блока X , затем в нее же и вернуться, нам необходимо сделать четное число шагов. Следовательно, любой цикл в двудольном графе $G[X, Y]$ обязан иметь четную длину.

Предположим теперь, что в графе G циклы нечетной длины отсутствуют. Сразу заметим, что при доказательстве нам достаточно ограничиться связными графами — любой несвязный граф является двудольным тогда и только тогда, когда двудольной является любая из его компонент связности. Кроме того, мы можем считать, что граф G построен на $n > 1$ вершинах — в случае $n = 1$ граф $G = K_1$ по определению является двудольным.

Выберем в G произвольную вершину $x \in V(G)$ и разобьем множество $V(G)$ вершин на два блока. К блоку Y отнесем все вершины $y \in V(G)$, для которых длина кратчайшего пути из z в y нечетна, а к блоку X отнесем все оставшиеся вершины. В частности, сама вершина $x \in X$, а все смежные с ней вершины принадлежат подмножеству Y . Докажем, что любые две вершины x', x'' подмножества X смежными не являются (рис. 18).

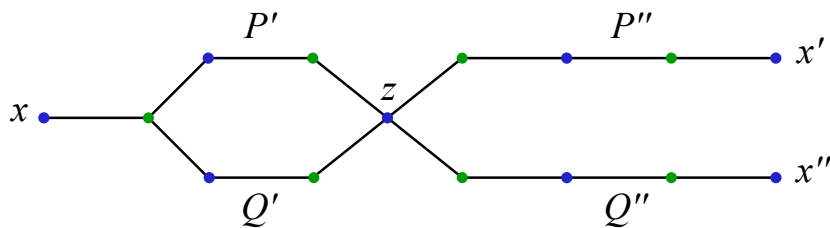


Рис. 18

Рассмотрим для этого произвольные кратчайшие пути P и Q , соединяющие эти вершины с вершиной x . Обозначим через $z \in V(G)$ последнюю общую вершину этих путей. Вершина z разделяет пути P и Q на два подпути — участки P' и Q' этих путей от x до вершины z , а также подпути P'' и Q'' , соединяющие z с вершинами x' и x'' соответственно. Сразу заметим, что длины путей P' и Q' обязаны совпадать — в противном случае мы могли бы сократить путь от x до одной из двух вершин x', x'' , что невозможно. Кроме того, так как x' и x'' принадлежат блоку X , то длины путей P и Q имеют одинаковую четность. Как следствие, одинаковую четность имеют и участки P'' и Q'' путей P и Q . Кроме того, P'' и Q'' не имеют никаких других общих вершин, помимо z . Добавление к ним ребра $e = \{x', x''\}$ приводит в образованию в G цикла нечетной длины, чего быть не может.

Аналогично доказывается, что и любые две вершины $y', y'' \in Y$ смежными не являются. Следовательно, граф G является двудольным. \square

Замечание 8.12. Идея, описанная при доказательстве теоремы 8.11, лежит в основе довольно элементарного алгоритма проверки графа на двудольность, основанная на поиске в ширину (см. первый параграф второй главы). Именно, выберем произвольную вершину x графа G и окрасим ее в синий цвет. Затем запустим из этой вершины поиск в ширину и начнем оставшиеся вершины окрашивать в два цвета (синий или зеленый) в зависимости от четности расстояния

от этих вершин до вершины x . В случае, если из какой-то вершины x' имеется ребро в уже окрашенную вершину x'' , нам следует проверить, что эти две вершины окрашены в разные цвета. Если это не так, граф двудольным не является. Если же в процессе работы алгоритма конфликтов не возникнет, граф является двудольным. Более формальное описание данного алгоритма, равно как и его реализацию на языке C++, можно посмотреть, например, в книге [?].

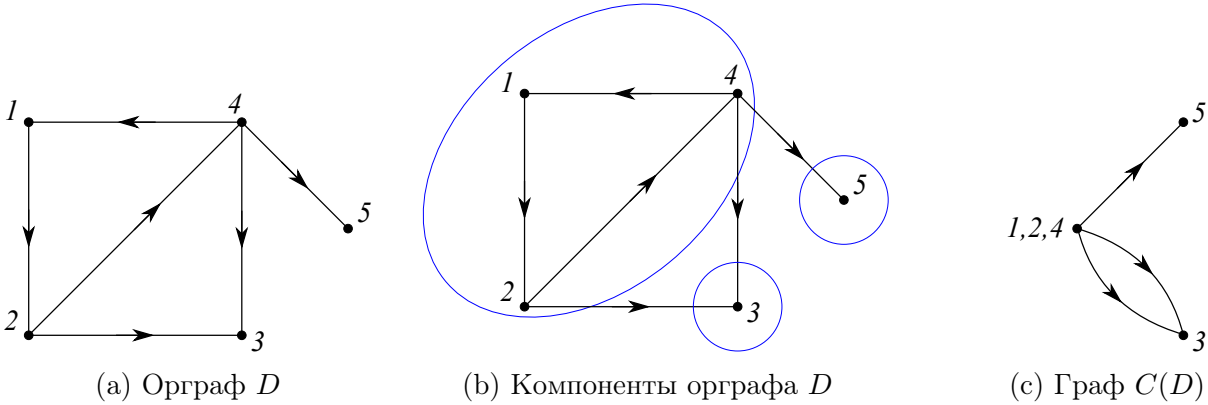


Рис. 19

8.2. Рассмотрим теперь понятие связности в ориентированном графе D .

8.2.1. Заметим, прежде всего, что понятия маршрута, пути, простого пути и цикла достаточно естественно переносятся на ориентированные графы. В качестве примера рассмотрим ориентированный граф D , показанный на рис.19,а. В таком графе последовательность

$$(1, 2, 4, 1, 2, 3)$$

задает нам маршрут из вершины 1 в вершину 3.

8.2.2. Теперь мы можем перейти к определению понятия связанных вершин в орграфе.

Определение 8.13. Вершины x и y орграфа D называются *связанными*, если в D существует хотя бы один путь из вершины x в вершину y , а также хотя бы один путь из вершины y в вершину x .

Вернемся к орграфу D на рис.19,а. В таком графе вершины 1 и 4 связаны между собой — мы можем пройти из 4 в 1 по ребру $(4, 1)$, а также из вершины 1 в вершину 4 по простому пути $(1, 2, 4)$. Вершины же 1 и 3 связанными между собой не являются — из вершины 1 в вершину 3 путь существует, однако из вершины 3 в вершину 1 добраться невозможно.

8.2.3. Как и в случае неориентированных графов, в случае орграфов отношение связности является отношением эквивалентности. Как всякое отношение эквивалентности, оно разбивает множество $V(D)$ вершин орграфа D на классы эквивалентности, называемые *компонентами сильной связности* орграфа D .

В качестве примера рассмотрим орграф, показанный на рис.19,а. Множество вершин такого орграфа разбивается отношением связности на три блока — блок, состоящий из вершин 1, 2, 4, блок, состоящий из единственной вершины 3, а также блок, состоящий из единственной вершины 5 (рис.19,б).

Определение 8.14. Орграф D называется *сильно связным*, если он состоит из единственной компоненты сильной связности, то есть если любые две его вершины являются связанными.

Иногда наряду с этим понятием для орграфа вводят понятие слабой связности.

Определение 8.15. Орграф D называется *слабо связным*, если соответствующий ему неориентированный граф G , получающийся заменой всех ориентированных ребер на неориентированные, является связным.

Показанный на рис.19,а орграф D является примером слабо связного орграфа, не являющегося сильно связным.

8.2.4. Вернемся к отношению связности в орграфе, разбивающему множество $V(D)$ вершин орграфа D на классы эквивалентности — компоненты сильной связности. Когда мы рассматривали отношение связности в неориентированном графе G , мы говорили, что между компонентами связности такого графа ребра отсутствуют. В случае ориентированного графа это не так (см., например, рис.19,а) — в орграфе такие ребра могут существовать, однако направлены все они будут лишь в одну сторону, от одной компоненты связности к другой.

Именно, справедливо следующее достаточно очевидное утверждение.

Лемма 8.16. Пусть H_1, H_2 есть две различные компоненты сильной связности графа D , и пусть существует ребро $e \in E(D)$ из H_1 в H_2 . Тогда ребра из H_2 в H_1 отсутствуют.

Доказательство. Действительно, если бы ребро из H_2 в H_1 присутствовало в D , то тогда любые две вершины в множестве $H_1 \cup H_2$ вершин оказались бы связанными. Но тогда объединение $H_1 \cup H_2$ представляло бы собой компоненту сильной связности графа D , что невозможно — мы изначально предполагали, что H_1 и H_2 представляют собой две различные компоненты сильной связности. \square

8.2.5. По любому орграфу D можно построить так называемый граф $C(D)$ компонент сильной связности графа D , вершинами которого будут компоненты сильной связности графа D , а ребрами — ребра графа D , направленные из одной компоненты сильной связности D к другой (рис.19,с). Основное свойство такого орграфа $C(D)$ состоит в том, что в таком графе нет циклов.

Теорема 8.17. В орграфе $C(D)$ циклы отсутствуют, то есть он, как еще говорят, представляет собой ациклический орграф (DAG — directed acyclic graph).

Доказательство. Если бы в таком графе существовал цикл вида $H_1 \rightarrow H_2 \rightarrow \dots \rightarrow H_n \rightarrow H_1$, то любые две вершины в объединении $H_1 \cup H_2 \cup \dots \cup H_k$ оказались бы связанными. Действительно, внутри каждой компоненты H_i мы, по определению сильной связности, можем попасть из любой вершины в любую вершину H_i . Вершины же из разных компонент H_i и H_j мы также можем всегда связать с помощью пути, идущего из H_i в H_j , а также пути, соединяющего компоненты H_j и H_i . \square

8.2.6. Ориентированные ациклические графы достаточно часто встречаются в практических приложениях. В качестве характерного примера можно рассмотреть орграф D , моделирующий порядок изложения материала в учебнике или онлайн-курсе. Любой параграф (урок) можно изображать вершиной орграфа D . Ребро между вершинами x и y в D проводится в случае, если соответствующий вершине x параграф является пререквизитом к параграфу, моделируемому вершиной y , то есть в случае, когда для изучения материала, изложенного в параграфе

y , необходим материал, изложенный в параграфе x . Довольно очевидно, что в таком орграфе D циклов быть не должно — в противном случае параграф x ссылался бы на параграф y и наоборот, и было бы не очень понятно, как изучать подобным образом организованный материал.

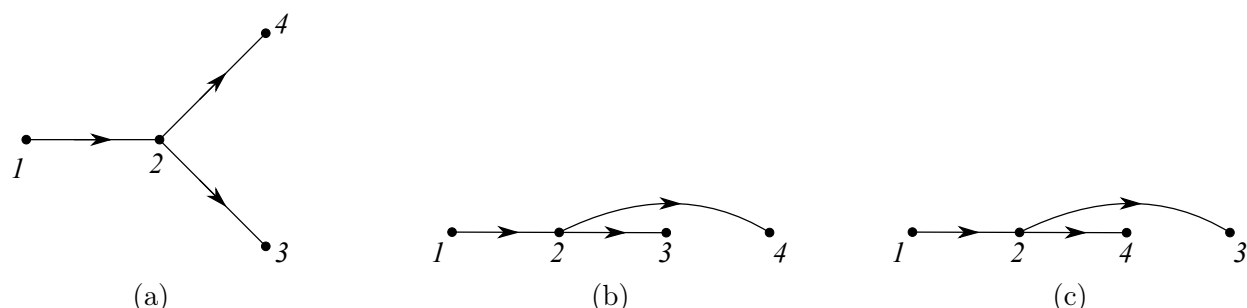


Рис. 20: Различные способы топологической сортировки вершин ациклического орграфа

Теперь предположим, что мы хотим организовать учебный курс, основанный на данном учебнике. Первый вопрос, который при этом возникает — это в каком порядке рассказывать изложенный в нем материал. В простейшем случае в учебнике информация подается линейно — за первым параграфом излагается второй, за вторым — третий, и так далее. Такого рода учебник моделируется простым ориентированным путем P , и проблем с организацией курса по этому учебнику не возникает. Однако в более сложно устроенных учебниках могут встречаться параграфы, которые можно проходить в различном порядке. Так, в учебнике, модель подачи материала в котором описывается орграфом D , показанным на рис.20,а, материал первых двух параграфов излагается линейно. Затем в книге идут два параграфа, не зависящие друг от друга. Как следствие, у лектора, использующего такой учебник в своем учебном процессе, возникает дилемма — он может либо рассказывать вначале материал параграфа 3, а затем — материал, изложенный в параграфе 4 (рис.20,б), либо начать с параграфа 4 и закончить параграфом 3 (рис.20,с).

С точки зрения теории графов тот или иной способ линейного упорядочивания вершин ациклического орграфа носит название *топологической сортировки* его вершин. Топологическая сортировка называется *правильной*, если после сортировки все ребра направлены слева направо. Иными словами, вершины орграфа отсортированы правильно в случае, если для любого ребра (x, y) вершина x имеет более ранний номер в линейном порядке по сравнению с номером, присвоенном вершине y . Как мы с вами видели на примере орграфа, показанного на рис.20, для одного и того же орграфа в общем случае существует довольно много различных способов правильно отсортировать его вершины.

На практике топологическую сортировку вершин ациклического орграфа проводят с помощью поиска в глубину (см. первый параграф второй главы). Такого рода алгоритм имеет линейное по количеству вершин в графе время работы.

9 Подграф графа G . Основные операции над графами

9.1. Вернемся к основным определениям теории графов и введем очень важное понятие подграфа графа G . Заодно мы введем две основные операции над графами — операцию удаления ребра и операцию удаления вершины в графе G .

9.1.1. Начнем с формального определения подграфа H графа G .

Определение 9.1. Подграфом графа G называется граф H , для которого выполнены следующие три условия:

1. $V(H) \subseteq V(G)$;
2. $E(H) \subseteq E(G)$;
3. любое ребро $e \in E(H)$, соединяющее пару вершин x и y в H , должно соединять ту же самую пару вершин в графе G .

По отношению к графу H граф G иногда называют *надграфом* или *суперграфом*.

Первые два условия, данные в определении подграфа, более-менее очевидны. Третье же условие требует некоторых комментариев. Рассмотрим в качестве первого примера знакомый нам граф G (рис.21,а), а также граф H , показанный на рис.21,б. У обоих этих графов множества вершин совпадают. Кроме того, множество $E(H)$ ребер графа H представляет собой подмножество множества $E(G)$ ребер графа G (в графе H ребро c отсутствует). Наконец, любое ребро, соединяющее вершины в графе H (например, ребро $f \in E(H)$), соединяет те же самые две вершины и в графе G (например, ребро f соединяет и в H , и в G одну и ту же пару вершин — вершины 1 и 2). Как следствие, граф H является подграфом графа G .

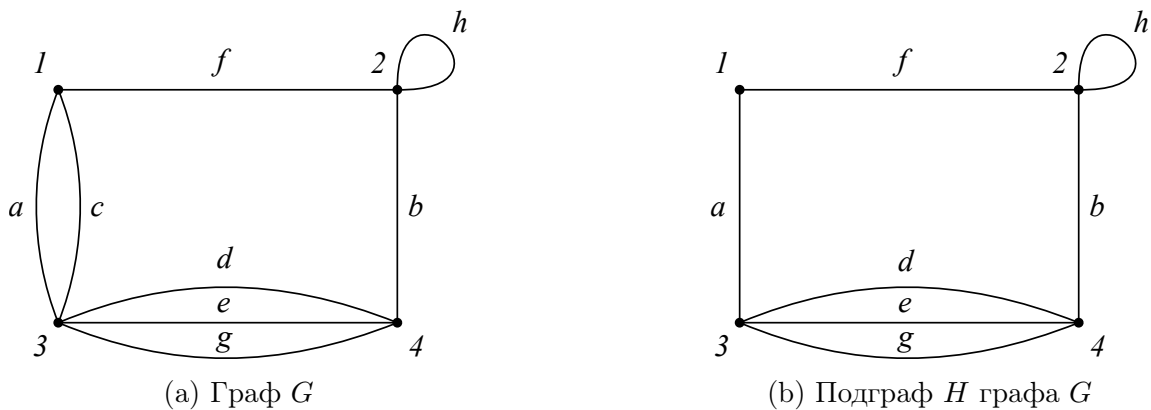
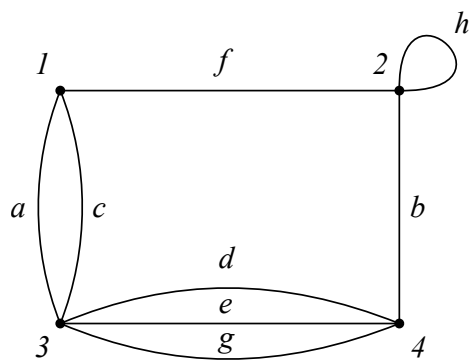


Рис. 21

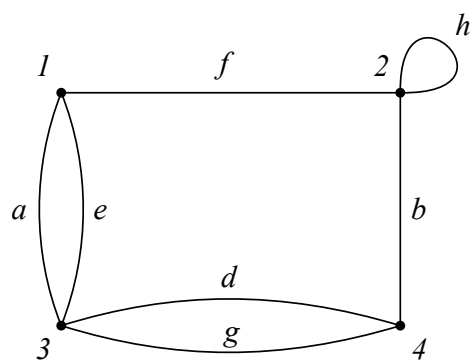
Давайте теперь посмотрим на графы G и H' , показанные на рис.22. Множества $V(G)$ и $V(H')$ у таких графов по-прежнему совпадают, множество $E(H')$ является подмножеством множества $E(G)$ ребер графа G . Однако у графа H' ребро e соединяет вершины 1 и 3, тогда как в графе G это же ребро соединяет вершины 3 и 4. Как следствие, условие 3 определения подграфа для H' нарушается, так что H' подграфом графа G не является.

9.1.2. Данное выше определение можно сделать несколько более конструктивным, если ввести две основные операции над графами — операцию удаления ребра и операцию удаления вершины. Оказывается, любой подграф H графа G — это граф, полученный из исходного графа G с помощью этих двух операций.

Начнем с более простой операции удаления ребра e . При такой операции множество вершин графа G не меняется, а из множества ребер удаляется элемент $e \in E(G)$. Полученный в результате этой операции граф обозначается $G - e$. Очевидно, что он является подграфом графа G .



(a) Граф G



(b) Граф H' , не являющийся подграфом графа G

Рис. 22

Действительно, в этом случае $V(H) = V(G)$, и $E(H) \subset E(G)$. Кроме того, так как мы никакие другие ребра графа G мы в процессе удаления ребра e не трогали, то в этом случае условие 3 определения подграфа у нас выполняется автоматически. Так, в приведенном на рис.21 примере мы в графе G удалили ребро c , соединяющее вершины 1 и 3, и получили подграф $H = G - c$.

В более общем случае мы таким образом можем удалить сразу несколько ребер, принадлежащих некоторому подмножеству S множества $E(G)$ ребер графа G . Полученный в результате этих операций подграф H обозначается $G - S$.

Перейдем теперь к чуть более сложной операции удаления вершины. Предположим, что мы хотим удалить в графе G вершину x . Если эта вершина является изолированной, то нам ничего больше делать не нужно. Если же этой вершине инцидентны какие-то ребра, то мы обязаны будем также вместе с вершиной x удалить и их. Действительно, ранее эти ребра вели в вершину x . Вести в никуда ребра не могут, так что нам вместе с вершиной x приходится удалять и все ребра, инцидентные данной вершине. Так, в примере мы удалили в графе G вершину c . Вместе с ней мы вынуждены были удалить и все три инцидентных ей ребра.

Полученный в результате удаления вершины x граф обозначается обычно $G - x$. Очевидно, что граф $H = G - x$ также является подграфом исходного графа G . Действительно, в этом случае $V(H) \subset V(G)$, $E(H) \subseteq E(G)$, а третье условие в определении подграфа у нас вновь выполняется автоматически — мы вновь ничего с оставшимися ребрами графа G не делали в процессе удаления вершины x .

В более общем случае $S \in V(G)$ мы с помощью операции удаления вершин из подмножества S получаем подграф $G - S$, в котором по сравнению с исходным графом G удалены все вершины подмножества S вместе со всеми ребрами, инцидентными этим вершинам.

Теперь рассмотрим граф H , полученный из графа G последовательным выполнением двух операций — операции удаления вершины и операции удаления ребра. Так как для каждой операции условие 3 в определении подграфа выполняется автоматически, то мы тем самым получаем, что граф H является подграфом графа G . Иными словами, мы получаем конструктивное определение подграфа H графа G как графа, полученного из G удалением некоторого количества вершин и/или ребер.

В качестве примера на рис.23,b показан граф H . Так как он получен из графа G , показанного на рис.22,a, удалением вершины 1 и ребер e , h , то граф H является подграфом графа G .

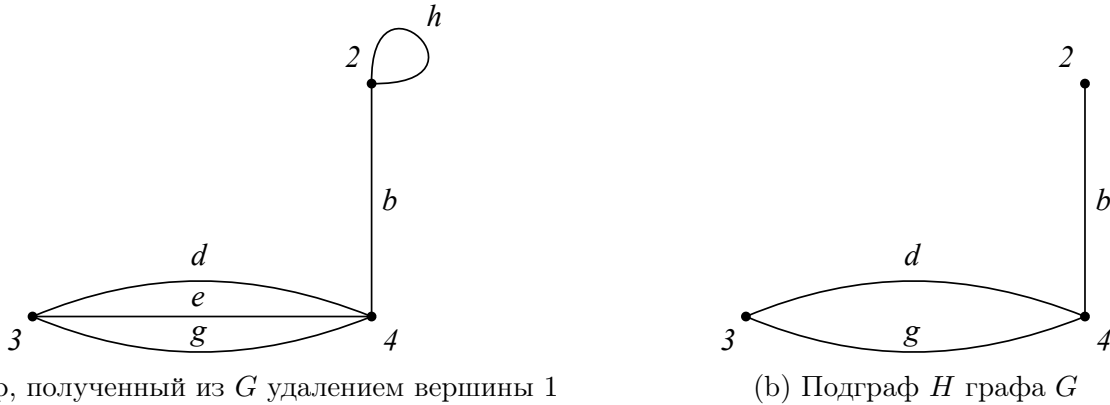


Рис. 23

9.1.3. Как мы уже заметили, любой подграф графа G получается из G последовательным выполнением двух операций — удаления вершин и удаления ребер. Естественным кажется рассмотреть два частных случая этой ситуации. Первый — это случай, когда мы в графе G удаляем только ребра, а второй — когда в G мы удаляем только вершины.

Если мы при получении из графа G подграфа H используем лишь операцию удаления ребер, то мы получаем подграф, множество вершин которого совпадает с множеством $V(G)$ вершин исходного графа. Такой подграф называется *остовным* подграфом (spanning subgraph) графа G (смотри рис.22,b). В частности, любой простой граф G , построенный на n вершинах, является остовным подграфом полного графа K_n .

Второй частный случай — когда мы в графе G удаляем одну или несколько вершин. В результате такой операции мы получаем подграф H , *индуцированным подмножеством оставшихся вершин графа G* . Иными словами, подграфом H графа G , индуцированным подмножеством вершин S , называется граф, полученный из G удалением всех вершин, не принадлежащих множеству S , вместе со всеми инцидентными этим вершинам ребрами (смотри рис.23,a).

Иногда нам будет удобно использовать понятие подграфа, индуцированного некоторым подмножеством F множества $E(G)$ ребер графа G . Такой подграф состоит из ребер, принадлежащих F , а также из вершин, являющихся концевыми вершинами этих ребер.

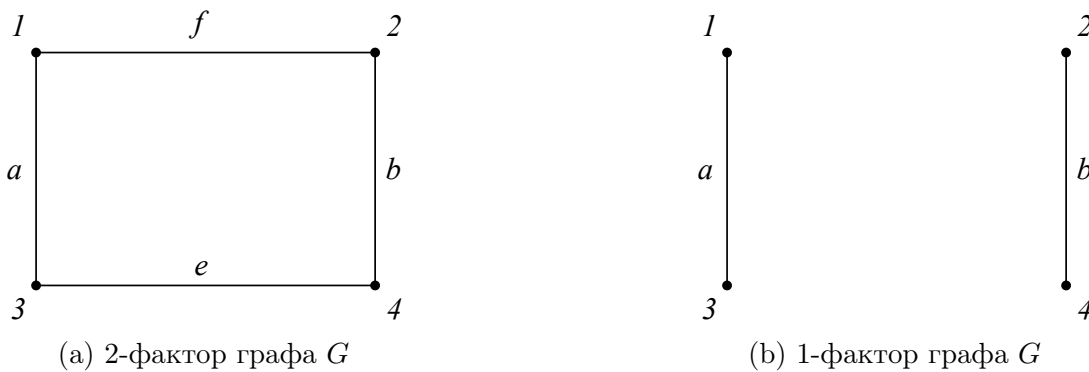


Рис. 24: k -факторы графа G

9.1.4. Достаточно часто в приложениях встречаются важные с практической точки зрения подклассы остовных подграфов. По историческим причинам некоторые из них имеют специальные названия. Так, остовный 1-регулярный подграф носит название 1-фактором графа G , а набор

ребер в таком подграфе называется *совершенным паросочетанием* в исходном графе G . Остовный k -регулярный подграф для произвольного натурального $k \geq 1$ называется k -*фактором* графа G .

В качестве примера на рис.24 показаны графы, являющиеся 2-фактором (рис.24,а) и 1-фактором (рис.24,б) графа G , показанного на рис.22,а.

В случае мультиграфа G очень часто рассматривают простой граф, получающийся из исходного графа G удалением всех петель и заменой мультиребер на простые ребра. Такой граф, очевидно, является остовным подграфом исходного графа.

9.2. Вернемся к понятию связности в графах. Операции удаления ребра и вершины позволяют нам ввести важные понятия моста и точки сочленения в графе.

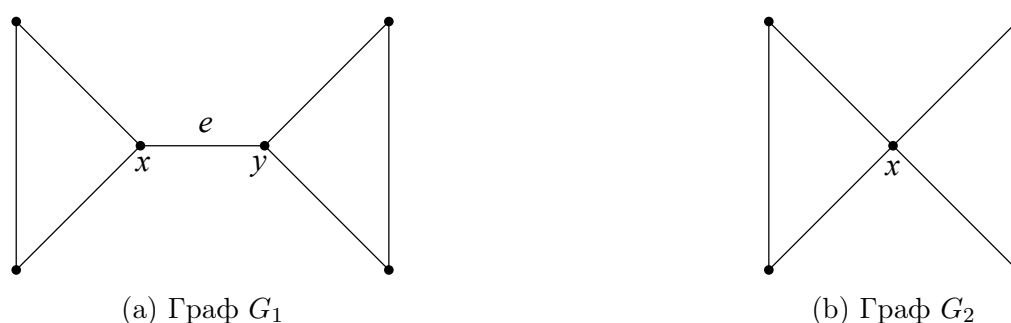


Рис. 25

9.2.1. Рассмотрим графы, показанные на рис.25. Видно, что связный граф G_1 (рис.25,а) перестает быть таковым после удаления ребра e , связывающего вершины x и y .

Определение 9.2. Ребро $e \in E(G)$ в связном графе G называется *мостом*, если получающийся после его удаления граф $G - e$ становится несвязным. В случае несвязного графа G мостом называется ребро, после удаления которого количество компонент связности увеличивается на единицу.

Теперь рассмотрим связный граф G , показанный на рис.25,б. У этого графа мостов нет, однако имеется вершина x , после удаления которой граф $G - x$ становится несвязным.

Определение 9.3. Вершина x называется *точкой сочленения* графа G , если после ее удаления количество компонент связности графа $G - x$ увеличивается по сравнению с количеством компонент связности исходного графа G .

Заметим, что даже в случае связного графа G удаление в нем точки сочленения может привести к появлению как двух, так и большего количества связных компонент.

9.2.2. Докажем простое утверждение, характеризующее точку сочленения в связном графе G .

Утверждение 9.4. Вершина x в связном графе G , построенном на $n \geq 3$ вершинах, есть точка сочленения графа G тогда и только тогда, когда в G существуют отличные от x вершины y и z , такие, что x содержится в каждом пути из y в z .

Доказательство. Предположим вначале, что x есть точка сочленения в связном графе G . Предположим, что утверждение неверно, то есть предположим, что для любой пары вершин y, z

найдется путь, не проходящий через x . Но тогда в графе $G - x$ любая пара вершин осталась бы связанной, что невозможно.

Теперь предположим, что в G существуют описанная в утверждении пара вершин y и z . Удаление вершины x в таком графе уничтожает любой путь из y в z , то есть превращает эти вершины в несвязанные. Это, в свою очередь, означает, что вершины y и z , лежащие ранее в одной компоненте связности графа G , стали принадлежать разным компонентам связности графа $G - x$. Иными словами, удаление x увеличивает количество связных компонент графа, то есть x есть точка сочленения в G . \square

9.2.3. Теперь докажем наиболее важное утверждение, характеризующее мосты в графе G .

Утверждение 9.5. Ребро $e = \{x, y\}$ в простом связном графе G является мостом тогда и только тогда, когда e не принадлежит ни одному из циклов графа G .

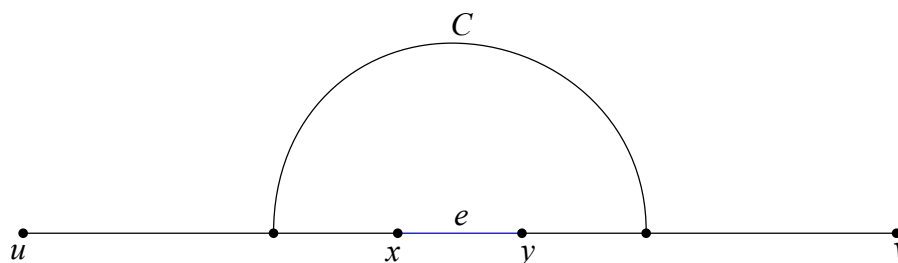


Рис. 26

Доказательство. Пусть ребро e является мостом в графе G . По определению, удаление e приводит к появлению в графе $G - e$ хотя бы одной пары $\{u, v\}$ несвязных между собой вершин (см.рис.26). В исходном графе G эти вершины связаны между собой простым путем $P = (u, v)$, который с необходимостью проходит через $e = \{x, y\}$. В графе $G - e$ вершины u и x связаны между собой участком $P_1 = (u, x)$ пути P , а вершины y и v — участком $P_2 = (y, v)$ этого пути. Если бы ребро e принадлежало циклу C в графе G , то x и y оказались бы связанными в графе $G - e$ путем $C - e$. Но тогда, по транзитивности, связанными бы оказались между собой и вершины u, v , что невозможно.

Обратно, предположим, что e не является мостом в графе G , то есть его удаление оставляет граф $G - e$ связным. Но тогда концы x и y ребра e в графе $G - e$ оказываются связанными некоторым xu -путем. Добавление к этому пути ребра e превращает этот путь в цикл. \square

Замечание 9.6. Доказанные выше утверждения являются, по сути, аналогами вершинной и реберной теорем Менгера, описывающих k -связные графы. Изучению таких графов будет посвящена отдельная глава.

9.3. Мы ввели операции удаления ребра и вершины. Нам понадобятся еще несколько операций над графами.

9.3.1. Начнем с чуть более сложной по сравнению с удалением ребра операции — операции стягивания ребра.

Определение 9.7. Говорят, что граф $G \setminus e$ получен из графа G стягиванием ребра e , если граф $G \setminus e$ получается из G удалением ребра e и стягиванием инцидентных e вершин x и y в одну (см.рис.27).

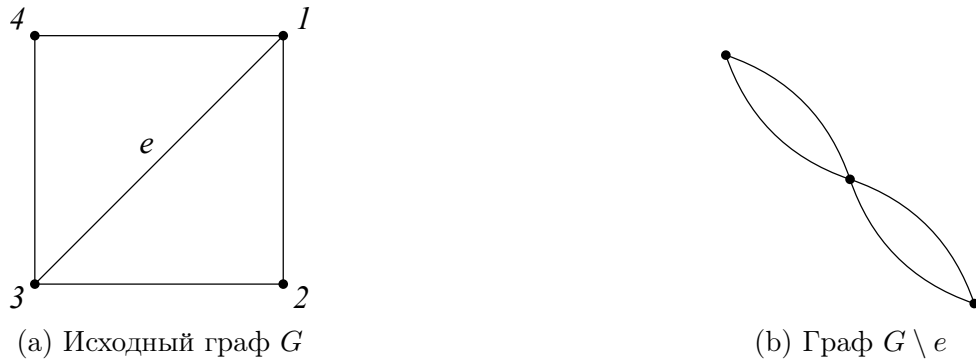


Рис. 27: Операция стягивания ребра

9.3.2. Следующие две полезные операции — это операции объединения и пересечения графов. Пусть G и H есть пара простых графов.

Определение 9.8. Объединением графов G и H называется граф $F = G \cup H$, множество вершин и множество ребер которого представляют собой объединение соответствующих множеств графов G и H :

$$V(F) = V(G) \cup V(H), \quad E(F) = E(G) \cup E(H).$$

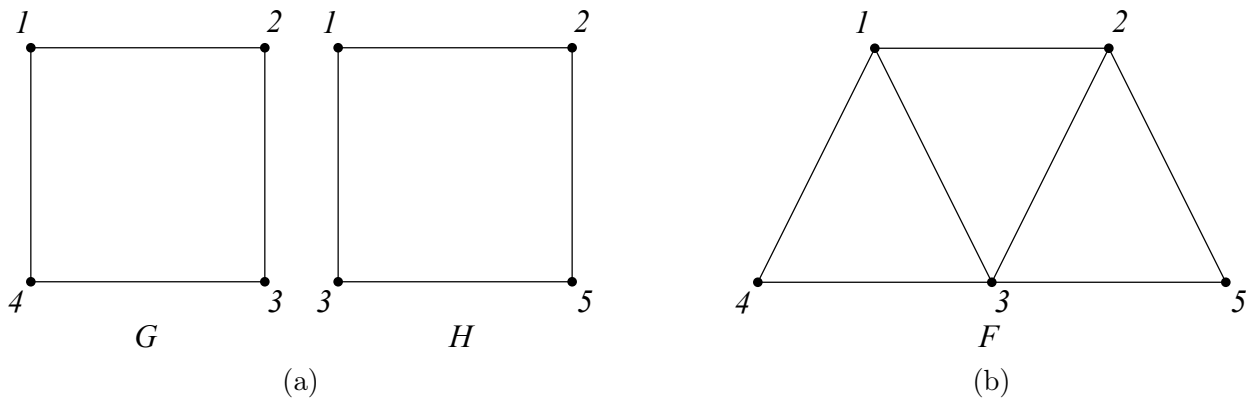


Рис. 28: Операция объединения графов G и H

На рис.28,b в качестве примера показан граф F , полученный в результате объединения графов G и H , показанных на рис.28,a.

В случае, если множества вершин (а следовательно, и ребер) графов G и H не пересекаются, то мы получаем так называемое несвязное объединение $G + H$ таких графов. По сути, любой несвязный граф представляет собой несвязное объединение двух или более связных компонент.

Определение 9.9. Пересечением графов G и H называется граф $F = G \cap H$, множество вершин и множество ребер которого представляют собой пересечение соответствующих множеств графов G и H :

$$V(F) = V(G) \cap V(H), \quad E(F) = E(G) \cap E(H).$$

На рис.29,b приведен граф F , полученный в результате пересечения графов G и H , показанных на рис.29,a.

Заметим, что мы уже, по сути, встречались с операциями объединения и пересечения графов, когда говорили о графе G и о его дополнении — графе \bar{G} . Действительно, в частном случае

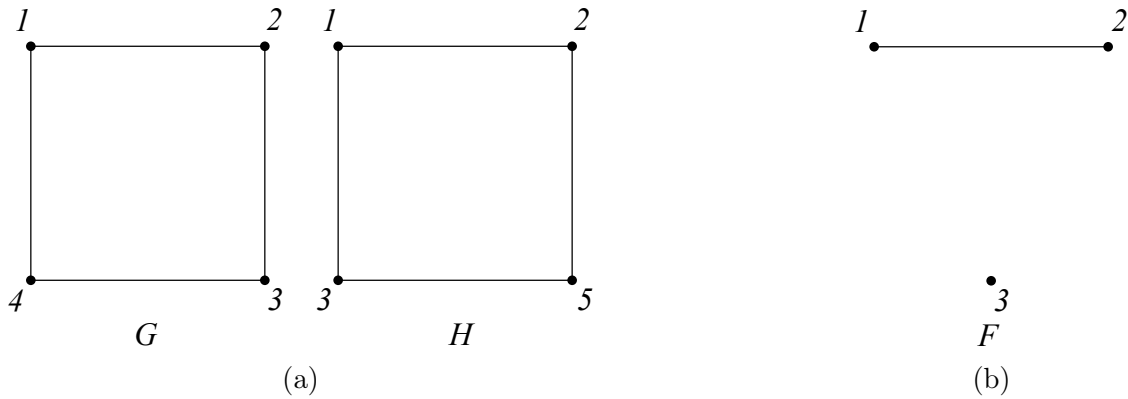


Рис. 29: Операция пересечения графов G и H

$H = \bar{G}$ объединение графов G и \bar{G} есть полный граф K_n на n вершинах, а пересечение графов G и \bar{G} — это пустой граф \bar{K}_n , построенный на n изолированных вершинах.

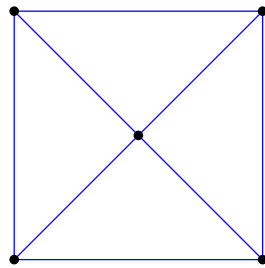


Рис. 30: Граф G

9.3.3. При изучении остовных подграфов H_1 и H_2 одного и того же графа G достаточно часто встречается операция Δ их симметрической разности. С помощью такой операции мы из остовных подграфов H_1 и H_2 получаем остовный подграф $H = H_1 \Delta H_2$, состоящий из ребер, принадлежащих H_1 , но не принадлежащих H_2 , и наоборот.

Рассмотрим в качестве примера граф G , показанный на рис.30. Два его остовных подграфа H_1 и H_2 показаны на рис.31,a и 31,b, а их симметрическая разность — остовный подграф $H_3 = H_1 \Delta H_2$ — представлен на рис.31,c. Общие для H_1 и H_2 ребра помечены на этих рисунках синим цветом, а ребра, встречающиеся в одном подграфе и не встречающиеся в другом — красным и зеленым цветами. Результирующий подграф H_3 состоит из ребер, окрашенных в красный и зеленый цвет.

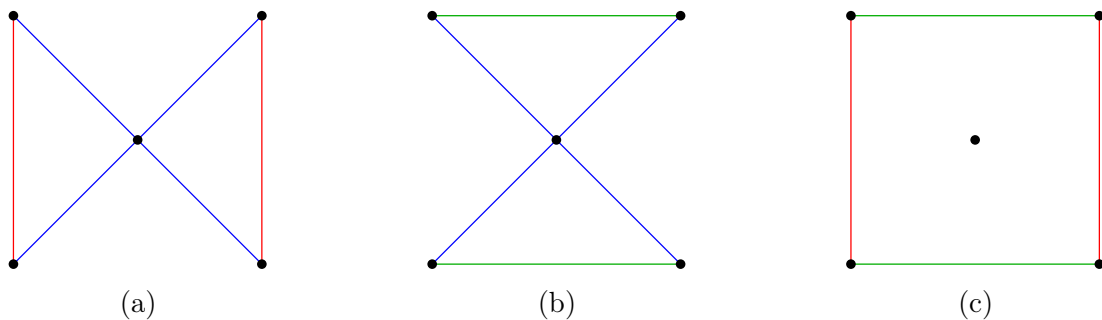


Рис. 31: Операция симметрической разности двух остовных подграфов

10 Изоморфизм и автоморфизм графов

10.1. Постараемся вначале ответить на вопрос, сколько же существует различных графов, построенных на n вершинах.

10.1.1. Сразу заметим, что в случае мультиграфов мы наряду с количеством вершин должны также фиксировать и количество ребер. Действительно, любые две вершины мультиграфа соединить произвольным количеством ребер, поэтому количество мультиграфов, построенных на n вершинах, без ограничения на количество ребер может быть сколь угодно большим. Поэтому мы ограничимся пока что задачей подсчета простых графов.

10.1.2. Количество g_n простых неориентированных графов достаточно легко сосчитать, используя формальное определение простого графа как некоторого подмножества множества $V^{(2)}$. Действительно, количество всех двухэлементных подмножеств n -элементного множества вершин (то есть мощность множества $V^{(2)}$) равно $|V^{(2)}| = \binom{n}{2}$. Нас же интересует множество Σ всех подмножеств $V^{(2)}$. Количество элементов в этом множестве, как известно, равно

$$|\Sigma| = 2^{|V^{(2)}|} = 2^{\binom{n}{2}}.$$

Следовательно, количество g_n всех простых графов на n вершинах равняется $2^{\binom{n}{2}}$.

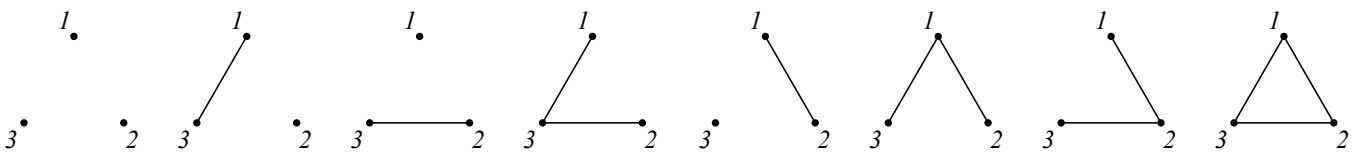


Рис. 32: Восемь простых графов на трех вершинах

Например, существует ровно $g_3 = 2^{\binom{3}{2}} = 2^3 = 8$ различных простых графов, построенных на трех вершинах (смотри рис.32).

10.1.3. Столь же просто подсчитать количество различных простых орграфов. Действительно, всего существует $n(n-1)$ упорядоченных пар отличных друг от друга вершин. Как следствие, количество различных простых орграфов, построенных на n вершинах, равно $2^{n(n-1)}$.

10.2. Заметим, что многие из приведенных на рис.32 графов похожи друг на друга в том смысле, что отличаются они друг от друга только перенумерацией вершин. Таковыми, в частности, являются три графа, имеющих только лишь одно ребро, а также три графа, имеющих ровно два ребра. Формализовать эту похожесть можно с помощью понятия изоморфизма графов.

10.2.1. Определим вначале понятие изоморфизма для простых неориентированных графов.

Определение 10.1. Говорят, что два простых графа G_1 и G_2 *изоморфны* друг другу, если существует взаимно-однозначное отображение $\varphi: V(G_1) \rightarrow V(G_2)$, которое сохраняет отношение смежности. Последнее означает, что если в графе G_1 некоторая пара вершин $\{x, y\}$ соединена ребром, то в графе G_2 соответствующая ей пара вершин $\{\varphi(x), \varphi(y)\}$ также соединена ребром, и наоборот.

Пример 10.2. Рассмотрим графы G_1 и G_2 , изображенные на рис.33. Они изоморфны друг другу, так как существует взаимно-однозначное отображение $\varphi: \{1, 2, 3\} \rightarrow \{1, 2, 3\}$ вида

$$\varphi(1) = 1, \quad \varphi(2) = 3, \quad \varphi(3) = 2,$$



Рис. 33: Два изоморфных графа

при котором ребро $\{1, 2\}$ графа G переходит в ребро $\{\varphi(1), \varphi(2)\} = \{1, 3\} \in E(G_2)$, а ребро $\{2, 3\}$ графа G переходит в ребро $\{\varphi(2), \varphi(3)\} = \{3, 2\}$ графа G_2 .

10.2.2. Как и любой изоморфизм, изоморфизм графов вводит отношение эквивалентности на множестве всех неориентированных простых графов на n вершинах. Отношение эквивалентности, в свою очередь, разбивает все множество таких графов на классы эквивалентности — классы изоморфных друг другу графов. В теории графов эти классы эквивалентности имеют специальное название.

Определение 10.3. Любой описанный выше класс эквивалентности графов называется *непомеченным* графом. В этом смысле обычный простой граф (то есть любой представитель класса эквивалентности) часто называется *помеченным* графом.

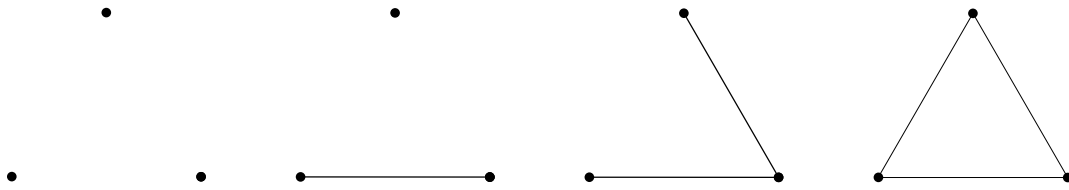


Рис. 34: Четыре простых непомеченных графа на трех вершинах

Так, из рис.32 видно, что существует четыре различных непомеченных графа на трех вершинах: один отвечает пустому графу, второй — трем помеченным графам с одним ребром, третий — трем помеченным графам с двумя ребрами, и наконец четвертый — полному графу K_3 (см.рис.34). Задача перечисления непомеченных графов на n вершинах для произвольного значения параметра n является значительно более сложной задачей по сравнению с задачей перечисления соответствующих помеченных графов.

10.2.3. Давайте теперь для полноты картины дадим определение изоморфизма для более общего случая неориентированных мультиграфов. Как мы помним, в общем случае любой неориентированный граф G задается тройкой, состоящей из множества $V(G)$ вершин, множества $E(G)$ ребер, а также отображения $I(G)$, сопоставляющего любому ребру $e \in E(G)$ неупорядоченную пару вершин $\{x, y\}$, которые это ребро e соединяет.

Определение 10.4. Говорят, что графы

$$G_1 = (V(G_1), E(G_1), I(G_1)) \quad \text{и} \quad G_2 = (V(G_2), E(G_2), I(G_2))$$

изоморфны друг другу, если существует пара взаимно-однозначных отображений

$$(\varphi, \vartheta), \quad \varphi: V(G_1) \rightarrow V(G_2), \quad \text{и} \quad \vartheta: E(G_1) \rightarrow E(G_2),$$

сохраняющих отношение смежности в графах G_1 и G_2 . Последнее в данном случае означает, что если отображение $I(G_1)$ сопоставляет в графе G_1 ребру $e \in E(G_1)$ пару $\{x, y\}$, то в графе G_2 отображение $I(G_2)$ сопоставляет ребру $\vartheta(e) \in E(G_2)$ пару вершин $\{\varphi(x), \varphi(y)\}$, и наоборот.

10.3. Часто на практике достаточно важно уметь отвечать на вопрос, являются ли два различных графа изоморфными друг другу или нет. Оказывается, проверка двух графов на изоморфизм является в общем случае довольно-таки нетривиальной задачей.

10.3.1. Начнем вначале с простого примера.

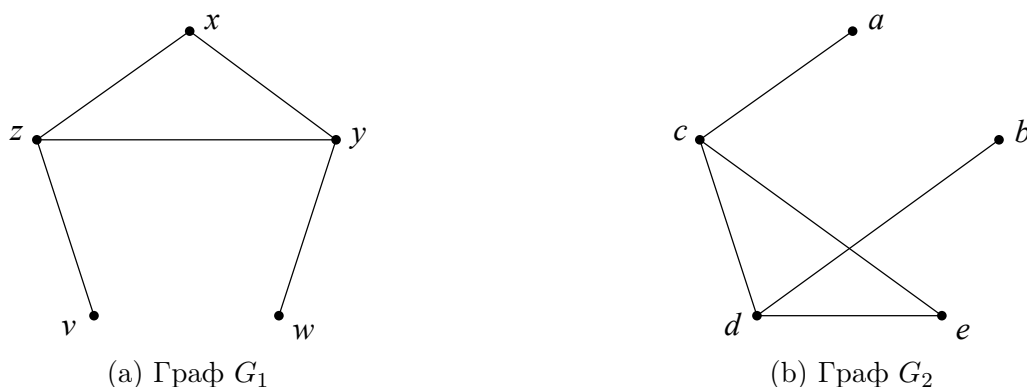


Рис. 35: Установление изоморфизма между графами

Пример 10.5. Рассмотрим два графа, изображенных на рис.35. Возникает вопрос, являются ли два таких графа изоморфными друг другу.

Для ответа на этот вопрос нам нужно, вообще говоря, найти взаимно-однозначное отображение φ из множества вершин $V(G_1)$ в множество $V(G_2)$, сохраняющее отношение смежности. Легче всего в таких простых задачах действовать следующим образом: нужно удалить пометки вершин у первого и у второго графов, пометить вершины первого графа числами от 1 до n , а затем попытаться разметить вершины второго графа теми же числами от 1 до n так, чтобы такой изоморфизм φ был бы очевиден.

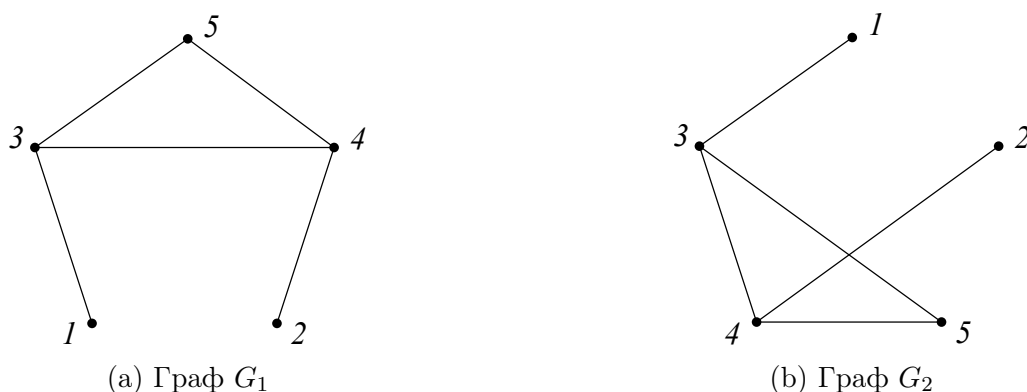


Рис. 36: Установление изоморфизма между графами

В данном конкретном примере сделать это довольно просто (см.рис.36): у графов G_1 и G_2 имеются всего лишь две вершины, степень которых равна единице. В графе G_1 они помечены цифрами 1 и 2. Поэтому нам будет удобно теми же самыми цифрами пометить одновалентные вершины и в графе G_2 . Далее, в графе G_1 эти вершины соединены с вершинами 3 и 4, имеющими степень, равную трем. Пометим аналогичные вершины в G_2 теми же цифрами. Нам остается пометить в графе G_2 оставшуюся вершину цифрой 5 и убедиться, что отображение $\varphi(i) = i$ задает изоморфизм этих графов.

Пример 10.6. Рассмотрим теперь графы, показанные на рис.37,а. Для таких графов уже не вполне очевидно то, что они изображают один и тот же непомеченный граф. Для доказательства этого факта можно, например, попытаться разметить вершины двух графов так, чтобы пара вершин $\{x, y\}$, связанная ребром в графе G_1 , оказалась бы связанной ребром и в графе G_2 , и наоборот. В данном примере это можно все же сделать достаточно просто, заметив, что граф G_2 состоит из двух непересекающихся циклов длины 4. Соответственно, выделив в графе G_1 замкнутый цикл аналогичной длины, мы можем уже однозначно пометить оставшиеся вершины графа G_1 так, чтобы списки смежности этих графов совпали (смотри рис.37,б).

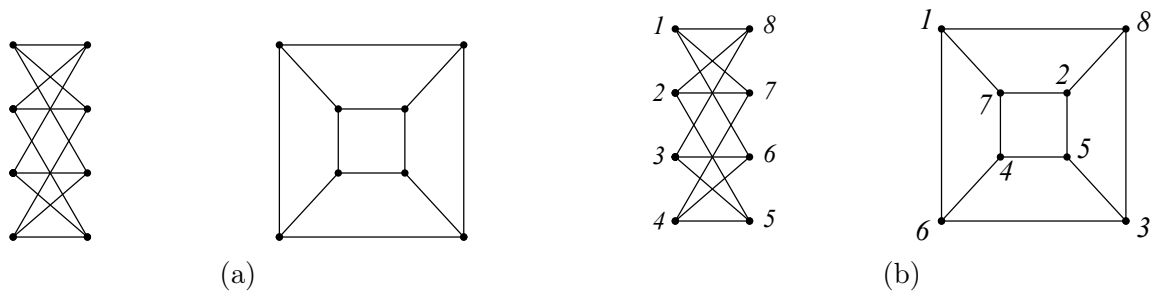


Рис. 37: Установление изоморфизма между графами

10.3.2. В практических задачах довольно часто встречаются графы, в которых содержатся сотни тысяч вершин. Для таких графов никакие визуальные методы оценки уже не работают. Не существует пока и никаких достаточно простых критериев, позволяющих достаточно быстро ответить на вопрос, являются ли два заданных графа изоморфными или нет. Конечно же, есть очевидные необходимые условия изоморфности двух графов — так, например, у изоморфных графов последовательности степеней их вершин должны, очевидно, совпадать. Но в случае, если подобного рода очевидные условия выполнены, то не остается ничего другого, как проверять заданную пару графов на изоморфность простым перебором.

Для этого вершины обоих графов помечают числами из одного и того же множества $[n]$, затем разметку первого графа фиксируют, а пометки второго графа начинают переставлять, вообще говоря, всеми $n!$ способами. Если на каком-то шаге матрицы смежности или же списки смежности этих двух графов совпадут, то графы являются изоморфными друг другу. В противном случае они неизоморфны.

Такого рода алгоритмы крайне неэффективны хотя бы потому, что при больших n значение $n!$ крайне велико. Существуют достаточно сложные алгоритмы, позволяющие более эффективно проверять на изоморфизм специальные подклассы графов, однако никакого быстрого алгоритма, позволяющего делать это в общем случае, пока не существует.

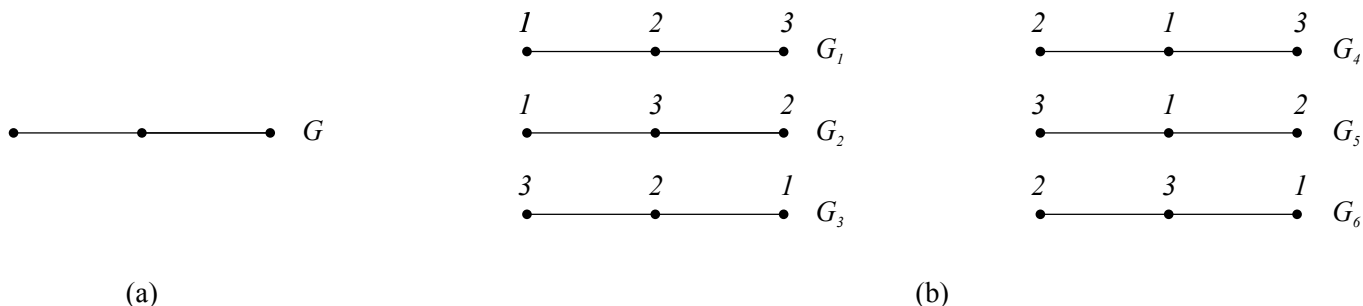


Рис. 38

10.4. Наряду с изоморфизмом, в теории графов вводится также понятие автоморфизма — изоморфизма графа G в себя.

10.4.1. Прежде чем давать формальное определение автоморфизма, вернемся еще раз к очень важным понятиям непомеченных и помеченных графов. Давайте подумаем, сколькими способами мы можем разметить один и тот же непомеченный граф на n вершинах. Очевидно, что мы это можем сделать $n!$ способами. Однако при этом возникает вопрос — действительно ли все такие способы пометки дадут нам *различные* помеченные графы?

В качестве примера рассмотрим простой непомеченный граф на трех вершинах, имеющий два ребра (рис.38,а). Пометим для него $3!$ способами его вершины. В результате мы получим 6 помеченных графов, изображенных на рис.38,б. Однако мы с вами уже перебрали все помеченные графы, построенные на трех вершинах, и выяснили, что существует только лишь три *различных* помеченных графа, имеющих два ребра. Следовательно, среди шести графов, показанных на рис.38,б, половина лишняя в том смысле, что среди этих графов имеются графы, представляющие собой один и тот же помеченный граф.

Несложно, в частности, убедиться в том, что графы G_1 и G_3 , показанные на рис.38,б, представляют собой один и тот же помеченный граф — первый из них переходит во второй при его развороте на 180° . То же касается и графов G_2 и G_6 , а также графов G_4 и G_5 .

10.4.2. Давайте теперь постараемся разобраться с причиной того, что при разметке вершин непомеченного графа $n!$ способами мы получаем лишние графы. Для этого возьмем непомеченный граф \tilde{G} , как-то разметим его вершины (например, так, как это показано на рис.39,а), а затем рассмотрим поподробнее все функции $\varphi: [3] \rightarrow [3]$, задающие перестановки вершин полученного таким образом помеченного графа G_1 .



Рис. 39: Изоморфные помеченные графы на трех вершинах

Рассмотрим вначале функцию φ_1 , оставляющую единицу на месте и меняющую местами вершины 2 и 3:

$$\varphi_1(1) = 1, \quad \varphi_1(2) = 3, \quad \varphi_1(3) = 2.$$

В результате мы из графа G_1 получим помеченный граф G_2 (рис.39,б). Естественно, что графы G_1 и G_2 изоморфны друг другу, так как они получены из одного и того же непомеченного графа. Однако важно то, что это — два различных графа: списки смежности этих двух графов различны:

$G_1 : \begin{array}{l} 1 \text{ смежна с } 2 \\ 2 \text{ смежна с } 1 \text{ и } 3; \\ 3 \text{ смежна с } 2 \end{array}$	$G_2 : \begin{array}{l} 1 \text{ смежна с } 3 \\ 2 \text{ смежна с } 3 \\ 3 \text{ смежна с } 1 \text{ и } 2. \end{array}$
--	--



Рис. 40: Идентичные помеченные графы на трех вершинах

Рассмотрим теперь перестановку φ_2 вершин графа G_1 вида

$$\varphi_2(1) = 3, \quad \varphi_2(2) = 2, \quad \varphi_2(3) = 1.$$

В результате такой перестановки мы из графа G_1 (рис.40,a) получим граф G_3 (рис.40,b). Важно, что у такого графа G_3 список смежности полностью совпадает со списком смежности исходного графа. Следовательно, в результате подобной перестановки φ_2 мы получаем тот же самый помеченный граф, что у нас был до перестановки. Такая перестановка вершин, при которой граф как помеченный объект переходит сам в себя, как раз и называется *автоморфизмом* графа.

Более формальное определение автоморфизма графа звучит следующим образом.

Определение 10.7. Взаимно-однозначное отображение $\varphi : V(G) \rightarrow V(G)$ множества $V(G)$ вершин простого графа G в себя называется автоморфизмом графа G , если она сохраняет отношение смежности в графе G . Последнее означает, что если вершины x и y соединены в графе G ребром $\{x, y\}$, то в том же самом графе G и образы $\varphi(x)$, $\varphi(y)$ этих вершин соединены между собой ребром.

10.4.3. Давайте еще раз сравним определения изоморфизма и автоморфизма графов. На первый взгляд эти определения очень похожи друг на друга, и сразу не очень понятно, чем они отличаются друг от друга. Казалось бы, отличие этих определений состоит в том, что в определении изоморфизма фигурируют различные множества $V_1(G_1)$ и $V_2(G_2)$ вершин, тогда как в определении автоморфизма эти множества совпадают. Однако это отличие в двух определениях абсолютно непринципально. Как правило, вершины любого графа помечаются элементами одного и того же множества $[n] = \{1, 2, \dots, n\}$ первых n натуральных чисел, так что и в определении изоморфизма, и в определении автоморфизма в роли отображения φ обычно выступает какая-то перестановка $\sigma : [n] \rightarrow [n]$ первых n натуральных чисел.

Ключевое же отличие этих двух определений заключается в следующем. В определении автоморфизма мы в результате отображения φ получаем *тот же самый* граф G , тогда как в случае изоморфизма, не являющегося автоморфизмом, граф G_1 под действием отображения φ переходит в некоторый *другой* граф G_2 , отличный от исходного графа G_1 , причем это верно даже в том случае, когда множества $V_1(G_1)$ и $V_2(G_2)$ вершин этих двух графов совпадают. Легче всего это отличие описать в терминах матрицы или списка смежности графов: в случае автоморфизма список смежности графа G не изменится, тогда как в результате действия изоморфизма списки смежности графов G_1 и G_2 будут, вообще говоря, отличаться друг от друга.

10.4.4. Рассмотрим теперь множество всех автоморфизмов какого-то простого помеченного графа G . Очевидно, что тождественная перестановка id принадлежит этому множеству. Далее, для любого автоморфизма φ обратный к нему автоморфизм φ^{-1} также не меняет граф G , то есть принадлежит тому же множеству автоморфизмов. Наконец, если φ_1 и φ_2 являются автоморфизмами графа, то и их композиция задает нам некоторый автоморфизм. Иными словами, операция композиции автоморфизмов задает на множестве всех автоморфизмов группу $\text{Aut}(G)$, которая называется *группой автоморфизмов* помеченного графа G .

Так, в разобранный выше примере группа автоморфизмов помеченного графа G_1 состоит из двух перестановок — тождественной перестановки id , а также перестановки, меняющей местами вершины 1 и 3. Иными словами, эта группа изоморфна группе \mathbb{Z}_2 .

10.4.5. Несмотря на то, что группа автоморфизмов определена для помеченного графа G , она, на самом-то деле, отражает внутреннюю симметрию исходного непомеченного графа \tilde{G} . Так, например, мы видим, что показанный на рис.38,a непомеченный граф \tilde{G} на трех вершинах симметричен относительно поворота на 180° . Проблема состоит лишь в том, что не очень понятно, как у непомеченного графа такие симметрии искать. И здесь нам как раз и приходится на помощь

помеченный граф: мы берем непомеченный граф \tilde{G} , как-то размечаем его вершины, получаем помеченный граф G , вычисляем для него группу автоморфизмов $\text{Aut}(G)$, а затем говорим, что эта группа и отражает внутреннюю симметрию исходного непомеченного графа \tilde{G} .

10.4.6. После сделанных замечаний мы можем вернуться к проблеме, с которой мы начинали — к определению количества различных помеченных графов.

Утверждение 10.8. *Количество N различных помеченных графов, получаемых из непомеченного графа \tilde{G} разметкой его вершин, рассчитывается по формуле*

$$N = \frac{n!}{|\text{Aut}(G_1)|}, \quad (41)$$

где G_1 — произвольно выбранный помеченный граф, полученный из \tilde{G} некоторой разметкой его вершин.

Доказательство. Выберем произвольный помеченный граф G_1 и сосчитаем для него группу автоморфизмов $\text{Aut}(G_1)$. Предположим вначале, что порядок этой группы совпадает с $n!$. Это означает, что любая перестановка вершин переводит граф G_1 в себя, то есть у нас получается лишь один помеченный граф, отвечающий исходному непомеченному графу \tilde{G} . В этом случае формула (41) верна — она дает $n!/n! = 1$, то есть единственный помеченный граф.

Теперь предположим, что $|\text{Aut}(G_1)| = k$, $k < n!$. В этом случае среди всех $n!$ перестановок вершин найдется перестановка $\tilde{\varphi}$, не принадлежащая $\text{Aut}(G_1)$. Иными словами, эта перестановка переведет нам граф G_1 в некоторый изоморфный, но отличный от G_1 граф G_2 . Применим тогда к графу G_1 все перестановки вида

$$\tilde{\varphi} \cdot \varphi_i, \quad \varphi_i \in \text{Aut}(G_1).$$

Количество таких перестановок совпадает, очевидно, с мощностью $|\text{Aut}(G_1)| = k$ группы автоморфизмов графа G_1 . Действительно, если $\tilde{\varphi} \cdot \varphi_i = \tilde{\varphi} \cdot \varphi_j$, то и $\varphi_i = \varphi_j$, поэтому для $\varphi_i \neq \varphi_j$ мы получаем и две различные перестановки вида $\tilde{\varphi} \cdot \varphi_i$ и $\tilde{\varphi} \cdot \varphi_j$. При этом все такие перестановки переведут граф G_1 в граф G_2 : вначале мы с помощью перестановки φ_i переводим граф G_1 в себя, а затем с помощью $\tilde{\varphi}$ из графа G_1 получаем граф G_2 .

Предположим теперь, что у нас нашлась еще какая-то перестановка $\hat{\varphi}$, отличная от $2k$ уже найденных перестановок и переводящая G_1 в некоторый граф G_3 . Тогда все перестановки вида $\hat{\varphi} \cdot \varphi$, $\varphi \in \text{Aut}(G_1)$ переводят G_1 в G_3 . При этом, так как $\hat{\varphi} \notin \text{Aut}(G_1)$, то G_3 отличен от G_1 . Покажем, что этот граф отличен и от G_2 . Действительно, если бы перестановки $\tilde{\varphi} \cdot \varphi_i$ и $\hat{\varphi} \cdot \varphi_j$ переводили граф G_1 в один и тот же граф G_2 , то мы бы имели равенства вида

$$\hat{\varphi} \cdot \varphi_j = \tilde{\varphi} \cdot \varphi_i \iff \hat{\varphi} = \tilde{\varphi} \cdot \varphi_i \cdot \varphi_j^{-1} = \tilde{\varphi} \cdot \varphi_k, \quad \varphi_k \in \text{Aut}(G_1),$$

а это невозможно — мы предположили, что $\hat{\varphi}$ отлична от перестановок вида $\tilde{\varphi} \cdot \varphi_k$.

Продолжая указанный процесс, мы в конце концов исчерпаем все множество Φ перестановок. При этом все это множество у нас окажется разбитым на N блоков, каждый из которых имеет мощность k . Каждый такой блок соответствует перестановкам, переводящим граф G_1 в некоторый изоморфный, но отличный от G_1 помеченный граф G_i . Общее же количество N различных помеченных графов связано с количеством $n!$ всех перестановок множества $V(G_1)$ вершин графа G_1 формулой

$$N \cdot k = N \cdot |\text{Aut}(G_1)| = n!,$$



Рис. 41: Идентичные помеченные графы на трех вершинах

из которой и следует формула (41). □

10.4.7. Переведем проведенные выше рассуждения на язык теории групп. Для этого рассмотрим множество \mathcal{X} всех помеченных графов, построенных на n -множестве $\{1, 2, \dots, n\}$ вершин, а также действие группы S_n перестановок на этом множестве \mathcal{X} . Действие перестановки σ на помеченный граф G при этом определим следующим образом: любая пометка i вершины графа переходит под действием перестановки σ в пометку $\sigma(i)$. В качестве примера рассмотрим граф G_1 на рис.41,а, а также перестановку $\sigma = (1, 2, 3)$. В этом случае пометка 1 вершины графа G_1 переходит под действием перестановки σ в пометку $\sigma(1) = 2$, пометка 2 — в пометку $\sigma(2) = 3$, а пометка 3 — в пометку $\sigma(3) = 1$. В результате такого действия мы получим граф G_4 , показанный на рис.41,б.

Говорят, что графы G_1 и G_2 изоморфны друг другу, если существует такая перестановка σ множества вершин, которая переводит граф G_1 в граф $G_2 = \sigma \circ G_1$, то есть такая перестановка, которая любое ребро $\{i, j\}$ в графе G_1 переводит в ребро $\{\sigma(i), \sigma(j)\}$. На языке теории групп это означает, что эти графы эквивалентны относительно действия группы S_n , то есть принадлежат одной и той же орбите

$$S_n \circ G = \{\sigma \circ G : \sigma \in S_n\}$$

элемента G множества \mathcal{X} всех графов, построенных на n вершинах, под действием группы S_n . Каждая такая орбита на языке теории графов представляет собой некоторый непомеченный граф на n -множестве вершин.

Рассмотрим теперь подмножество $\text{Aut}(G)$ перестановок σ , любой элемент которого оставляет некоторый помеченный граф $G \in \mathcal{X}$ неподвижным:

$$\text{Aut}(G) := \{\sigma \in S_n : \sigma \circ G = G\}.$$

Несложно понять, что такое подмножество перестановок образует подгруппу $\text{Aut}(G)$ группы S_n перестановок. Такая подгруппа на языке теории групп называется *стабилизатором* элемента G множества \mathcal{X} . Эта подгруппа (как и любая подгруппа H группы S_n) задает разбиение группы S_n всех перестановок на блоки $\sigma \cdot \text{Aut}(G)$, называемые смежными классами группы S_n по подгруппе $\text{Aut}(G)$. Размер этих блоков одинаков и равен порядку $|\text{Aut}(G)|$ группы $\text{Aut}(G)$. Следовательно, порядок $|S_n|$ всей группы перестановок равен произведению порядка $|\text{Aut}(G)|$ группы автоморфизмов графа G на количество смежных классов $S_n/\text{Aut}(G)$ группы S_n по подгруппе $\text{Aut}(G)$ (теорема Лагранжа):

$$|S_n| = n! = |\text{Aut}(G)| \cdot |S_n/\text{Aut}(G)|.$$

В рассмотренном нами примере мы разбили множество S_3 всех перестановок вершин графа G_1 на блоки, состоящие из перестановок вида $\tilde{\varphi} \cdot \varphi$, где $\varphi \in \text{Aut}(G_1)$, и показали, что размер каждого блока равен двум — порядку группы $\text{Aut}(G_1)$.

Затем мы заметили, что любому такому блоку перестановок отвечает некоторый граф G_i , изоморфный исходному графу G . На языке теории групп данный факт формулируется в виде достаточно важного, и в то же время достаточно просто доказываемого утверждения — так

называемой Orbit-Stabiliser Theorem. Согласно этой теореме, имеется взаимно-однозначное соответствие между всеми элементами орбиты $S_n \circ G$ графа G (то есть всеми изоморфными G графами на n вершинах) и фактор-множеством $S_n / \text{Aut}(G)$, то есть множеством всех смежных классов группы S_n по стабилизатору $\text{Aut}(G)$. Следствием этого утверждения как раз и является равенство

$$|S_n| = n! = |\text{Aut}(G)| \cdot |S_n \circ G| = |\text{Aut}(G)| \cdot N,$$

где N — количество графов на n вершинах, изоморфных G .

10.4.8. Дадим еще одно важное определение.

Определение 10.9. Граф G называется асимметричным, если его группа автоморфизмов тривиальна, то есть состоит только из тождественной перестановки множества $[n]$ его вершин.

Важно заметить, что чем выше n , тем большее количество графов асимметрично. Следовательно, при больших n количество \mathcal{M}_n непомеченных графов можно оценить по формуле

$$\mathcal{M}_n \sim \frac{2^{\binom{n}{2}}}{n!}.$$

11 Основные свойства деревьев

11.1. Одним из самых важных понятий теории графов является понятие дерева.

11.1.1. Начнем с формального определения этого понятия.

Определение 11.1. Деревом называется простой связный граф без циклов. Произвольный (не обязательно связный) простой граф без циклов называется лесом.

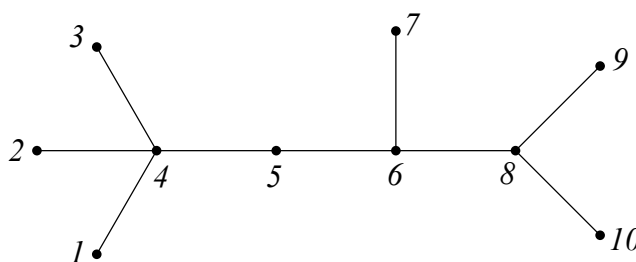


Рис. 42: Пример дерева на десяти вершинах

На рис.42 показан пример дерева T , построенного на 10 вершинах.

11.1.2. Изучим простейшие свойства деревьев. Начнем со следующего важного определения.

Определение 11.2. Вершина графа G , имеющая единичную степень, называется листом.

У дерева T , показанного на рис.42, имеется шесть листьев — вершины 1, 2, 3, 7, 9 и 10.

Лемма 11.3. У любого дерева T , построенного на $n \geq 2$ вершинах, имеется как минимум два листа.

Доказательство. Рассмотрим произвольный простой путь $P = (x_1, x_2, \dots, x_k)$ максимальной длины в T (рис.43). Для изображенного на рис.42 дерева таким путем является, например, путь $P = (1, 4, 5, 6, 8, 9)$. Очевидно, что такой путь в T всегда существует, причем его длина k равна диаметру $\text{diam}(T)$ дерева T . Покажем, что концы этого пути – вершины x_1 и x_k – обязаны быть листьями.

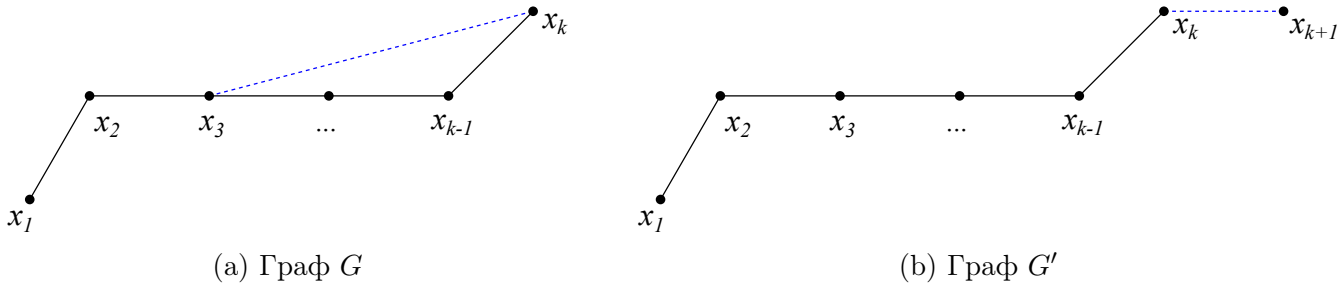


Рис. 43

Предположим, что это не так, то есть предположим, например, что степень вершины x_k больше или равна двум. Это означает, что из вершины x_k помимо ребра $\{x_{k-1}, x_k\}$ исходит еще хотя бы одно ребро. Всего имеет место два варианта развития событий. В первом из них исходящее из x_k ребро приходит в одну из вершин пути P , отличную от x_{k-1} (рис.43,а). Но в этом случае в дереве T появляется цикл, а это невозможно. Во втором варианте инцидентное x_k ребро приходит в какую-то вершину x_{k+1} , отличную от вершин пути P (рис.43,б). Но тогда в графе G имеется путь $(x_1, x_2, \dots, x_k, x_{k+1})$, длина которого на единицу больше длины исходного пути P . А это, в свою очередь, противоречит максимальнойности пути P . Полученное противоречие доказывает наше утверждение. □

11.1.3. Из доказанной леммы достаточно легко следует одно из основных свойств дерева T .

Теорема 11.4. *В дереве T , построенном на n вершинах, имеется ровно $(n - 1)$ ребро:*

$$|E| = |V| - 1 = n - 1.$$

Доказательство проведем индукцией по количеству n вершин в графе. База индукции очевидна — дерево, состоящее из $n = 1$ вершины, не имеет ни одного ребра. Предположим теперь, что утверждение доказано для деревьев, построенных на n вершинах, и покажем, что оно остается верным для произвольного дерева с $(n + 1)$ -й вершиной.

Действительно, по доказанной выше лемме 11.3 у любого такого дерева имеется хотя бы один лист x . Удалим теперь этот лист x вместе с инцидентным ему ребром e . Полученный в результате такой операции граф T' останется, очевидно, связным. Действительно, любой путь, соединяющий в T произвольные две вершины, отличные от x , ребро e содержать не может. Поэтому удаление e на связность оставшихся в T' вершин никак не повлияет. Наконец, от удаления ребра e никаких дополнительных циклов также появиться не может. Следовательно, граф T' является деревом, построенным на n вершинах. Но у такого дерева по индукционному предположению имеется $(n - 1)$ ребро. Следовательно, у исходного дерева T имеется ровно n ребер. □

Достаточно очевидно и обратное утверждение.

Теорема 11.5. *Любой простой связный граф G , построенный на n вершинах и имеющий $(n - 1)$ ребро, является деревом.*

Доказательство. Действительно, пусть это не так. Тогда в графе G имеется цикл C . В параграфе, посвященном основным операциям над графами, мы ввели важное понятие моста как ребра, после удаления которого простой связный граф теряет связность. В том же параграфе мы предъявили важную характеристику моста — мы сказали, что мостом в графе G является ребро, не принадлежащее никакому циклу в графе. Это утверждение можно еще переформулировать так: любое ребро e , принадлежащее циклу C , мостом в графе G не является. Следовательно, удаление такого ребра не приводит к потере связности графа G .

Рассмотрим тогда простой связный граф $G_1 = G - e$. Если в нем остались циклы, мы выберем один из них и удалим любое ребро, принадлежащее этому циклу. Продолжая далее, мы на каком-то шаге получим, наконец, простой связный граф без циклов, то есть дерево. Но мы только что доказали, что в дереве на n вершинах имеется ровно $n - 1$ ребро. Следовательно, нам для получения дерева никаких ребер удалять не пришлось, то есть исходный граф G изначально представлял собой дерево. \square

Следствие 11.6. *Всякий связный граф, построенный на n вершинах, имеет как минимум $(n - 1)$ ребро.*

Доказательство этого утверждения абсолютно аналогично доказательству теоремы 11.5. Действительно, рассмотрим произвольный связный граф. Если такой граф не является деревом, то у него имеется хотя бы один цикл. В этом цикле мы всегда можем удалить одно ребро, и граф при этом останется связным. Будем продолжать этот процесс до тех пор, пока в графе не останется ни одного цикла. В результате получим простой связный граф без циклов, то есть дерево. Количество ребер у дерева равно $(n - 1)$. Следовательно, у исходного графа имеется как минимум $(n - 1)$ ребро. \square

11.1.4. Следствие 11.6 по-другому можно переформулировать так: у всякого связного графа существует остовное дерево. Получить это дерево можно, либо удаляя лишние ребра, либо (что более рационально) используя так называемый алгоритм поиска в глубину.

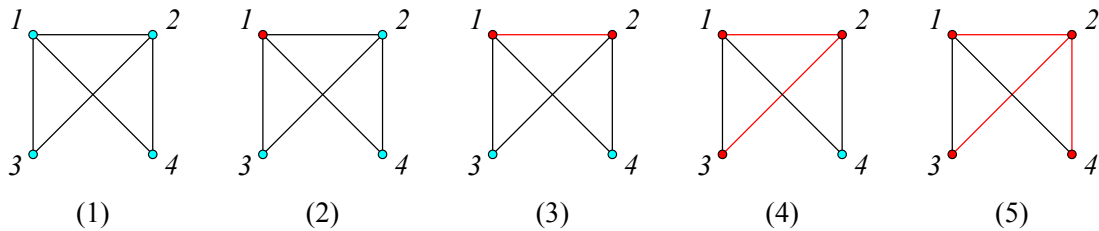


Рис. 44: Алгоритм поиска в глубину

В простейшем своем исполнении этот алгоритм выглядит так. Пусть у нас имеется связный граф G . Будем считать, что все его вершины и ребра окрашены в синий цвет. Выберем произвольную вершину x этого графа и окрасим ее в красный цвет. Затем начнем перебирать все инцидентные x ребра. В случае, если ребро e соединяет x с еще не окрашенной вершиной y , окрашиваем ребро $e = \{x, y\}$ в красный цвет и рекурсивно повторяем процесс, запуская алгоритм поиска в глубину с вершины y . Если для текущей вершины z все смежные с ней вершины оказываются окрашенными в красный цвет, то алгоритм поиска в глубину для нее заканчивается и мы переходим на более высокий уровень рекурсии.

На рис.44 приведено пошаговое выполнение алгоритма на примере графа G , построенного на четырех вершинах. Как видно, в результате работы алгоритма поиска в глубину все вершины перекрашиваются в красный цвет. Красным же цветом оказываются помечены и ребра, причем

красные вершины и ребра представляют собой остовное дерево $T(G)$ графа G . Действительно, по окончании работы алгоритма мы получаем окрашенный в красный цвет связный подграф, количество ребер в котором на единицу меньше количества вершин, а мы только что доказали, что такой подграф обязательно является деревом.

В чуть более общем виде несвязного графа G псевдокод алгоритма поиска в глубину можно записать так:

Процедура DFS(x), x - вершина графа G :

```
t := t + 1;
pre[x] := time;
color[x] := 1;
for ( y : (x,y) принадлежит E(G) )
  begin
    if ( color[y] == 0 )
      begin
        parent[y] := x;
        DFS(y);
      end
    end
  end
t := t + 1;
post[x] := time;
color[x] := 2;
```

Основная программа:

```
// Инициализация начальных значений
for ( x : x - вершина графа G )
  begin
    color[x] := 0;
    parent[x] := -1;
    pre[x] := 0;
    post[x] := 0;
  end
t := 0;
// Запуск основного алгоритма
for ( x : x - вершина графа G )
  begin
    if ( color[x] == 0 )
      begin
        DFS(x);
      end
    end
  end
```

В этом варианте алгоритма вместо двух цветов вершин мы используем три — цвет 0 (голубой цвет на рис.45) для еще необнаруженной вершины, цвет 1 (желтый цвет на рис.45) для обнаруженной, но не до конца обработанной вершины (то есть для вершины, у которой есть смежные с ней вершины, окрашенные в синий цвет), а также цвет 2 (красный цвет на рисунке) для обнаруженной и обработанной вершины. Кроме того, мы вводим в алгоритм параметр t — время, а также массивы $pre[]$ и $post[]$, отмечающие время захода и выхода в конкретную вершину x .

Так, время $pre[2]$ захода в вершину 2 для алгоритма, проиллюстрированного на рис.45, равно двум, а время $post[2]$ выхода равно семи. В массиве $parent[]$ у нас по результатам работы алгоритма для каждой вершины x , кроме начальной вершины для каждой компоненты связности графа G , хранится вершина, являющаяся предшественником вершины x при обходе графа G в глубину.

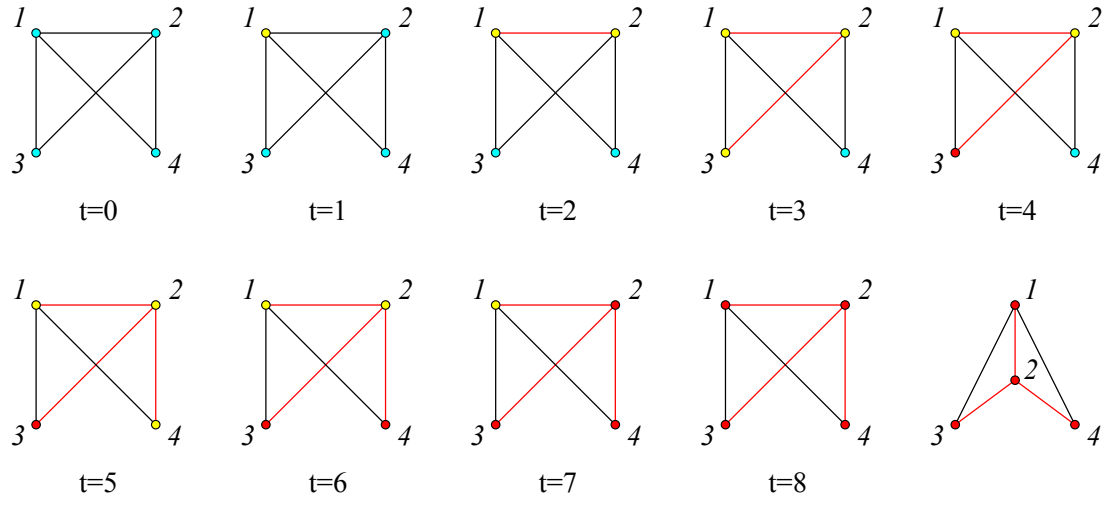


Рис. 45: Алгоритм поиска в глубину

Алгоритм поиска в глубину очень часто используется в разнообразных приложениях, связанных с теорией графов. Например, с помощью этого алгоритма можно найти все компоненты связности неориентированного графа, осуществить топологическую сортировку вершин ориентированного ациклического графа, найти мосты и/или точки сочленения в связном графе и так далее.

11.1.5. Отметим еще две важные характеристики дерева. Первая из них описывает ребра дерева.

Утверждение 11.7. *Связный граф T является деревом тогда и только тогда, когда любое его ребро является мостом.*

Доказательство практически очевидно. Действительно, по определению, в дереве T циклы отсутствуют. Следовательно, ни одно из его ребер циклу не принадлежит. Согласно основной характеристике моста это и означает, что любое ребро графа T является мостом. Обратно, если любое ребро связного графа является мостом, то оно не принадлежит никакому циклу. Поэтому данный граф обязан быть деревом. □

Вторая характеристика связана с понятием связности в графе. Мы знаем, что в любом связном графе любые две вершины связаны по крайней мере одним путем. Дерево — это связный граф, в котором для любых двух вершин такой путь единственен.

Утверждение 11.8. *Граф T является деревом тогда и только тогда, когда для любых двух его вершин существует единственный простой путь, соединяющий эти вершины.*

Доказательство. Пусть в T для любой пары вершин существует единственный соединяющий их простой путь. Тогда T является связным графом. Если бы в таком графе существовал цикл C , то между любыми двумя вершинами этого цикла существовало бы два различных пути, соединяющие эти вершины, а это невозможно.

Обратно, пусть T является деревом, и пусть в нем существуют два различных простых пути P и Q , соединяющих какие-то две вершины x и y . Рассмотрим симметрическую разность $P\Delta Q$ этих путей. Подграф $P\Delta Q$ состоит из одного или нескольких циклов, а это противоречит тому, что T является деревом. \square

Следствие 11.9. Пусть T есть остовное дерево связного графа G , и пусть e есть ребро, не принадлежащее T . Тогда граф $T + e$ содержит единственный цикл, причем этот цикл проходит через ребро e .

Доказательство. Заметим, прежде всего, что любой цикл в графе $T + e$ обязан содержать ребро $e = \{x, y\}$ — в противном случае мы бы получили цикл и в дереве T , чего быть не может. Кроме того, C есть цикл в $T + e$ тогда и только тогда, когда $C - e$ есть xy -путь в исходном дереве T . Но в дереве T xy -путь единственен. Следовательно, $T + e$ содержит единственный цикл, проходящий через ребро e . \square

11.1.6. Рассмотрим теперь простой связный граф G , в котором имеется хотя бы одна пара отличных друг от друга остовных деревьев T и T' .

Утверждение 11.10. Рассмотрим два различных остовных дерева T и T' простого связного графа G . Пусть e есть ребро остовного дерева T , не принадлежащее T' . Тогда в остовном дереве T' найдется ребро e' , не принадлежащее остовному дереву T , такое, что граф $T - e + e'$ является остовным деревом графа G .

Решение. Удаление ребра $e = \{x, y\}$ разваливает T на две связные компоненты. Обозначим через V_1 и V_2 множества вершин этих двух компонент, такие, что $x \in V_1, y \in V_2$. В дереве T' существует единственный путь, соединяющий вершины x и y . Такой путь обязан иметь ребро e' из V_1 в V_2 . Так как e есть единственное ребро в дереве T между V_1 и V_2 , то ребро e' не принадлежит дереву T . Граф $T - e + e'$ является связным, содержит $n - 1$ ребро, а следовательно, является остовным деревом графа G .

Заметим, что так как в дереве T имеется единственный путь между вершинами x и y , то добавление к этому пути ребра e' приводит к появлению единственного цикла в графе $T + e'$. Удаление же из этого графа ребра e уничтожает цикл, превращая граф $T - e + e'$ в связный ациклический подграф — еще одно остовное дерево графа G . \square

11.2. Иногда наряду с обычными деревьями и лесами рассматривают так называемые *корневые* деревья и леса.

Определение 11.11. *Корневым деревом* называется дерево с выделенной вершиной, называемой *корнем*. *Корневым лесом* называется лес, каждая компонента связности которого представляет собой корневое дерево.

11.2.1. Наглядно корневое дерево можно представлять себе так. Представим дерево в виде некоторого физического объекта, вершинами которого являются некоторые шарниры, а ребрами — соединяющие эти шарниры трубки. Подвешивая такое дерево за какую-то вершину x , мы как раз и получим корневое дерево с корнем в вершине x .

В качестве примера подвесим показанное на рис.46,а дерево за вершину b . В результате получим корневое дерево, показанное на рис.46,б.

11.2.2. В любом корневом дереве T_r с корнем в вершине r у нас имеется единственный путь из корня в произвольную вершину x . Длина этого пути определяет *уровень* вершины x в корневом

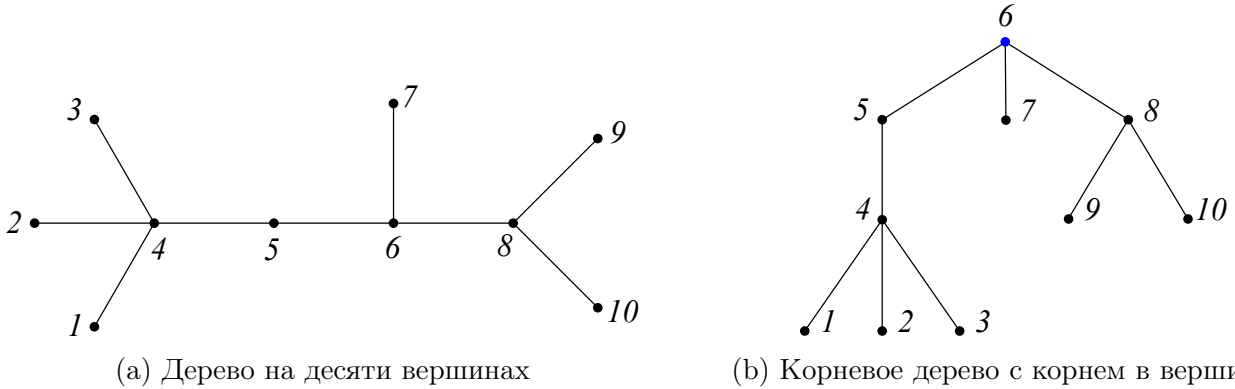


Рис. 46: Дерево и корневое дерево

дереве T_r . *Высотой* дерева называется длина наибольшего пути от корня r дерева до вершины x . Ясно, что в последнем случае вершина x обязана быть листом дерева T_r .

Так, изображенное на рис.46,b дерево T_r имеет высоту, равную трем. Уровень вершины 4 равен двум.

11.2.3. Иногда корневое дерево T_r удобно ориентировать от корня r дерева к его листьям (см.рис.47), получая ориентированный граф \vec{T}_r , который в английском языке носит название arborescence. В таком орграфе входящая степень любой вершины, отличной от корня, будет равняться единице. Корень ориентированного дерева \vec{T}_r будет в этом случае источником, то есть вершиной, входящая степень которой равна нулю, а любой лист будет стоком, то есть вершиной, исходящая степень которой равна нулю.

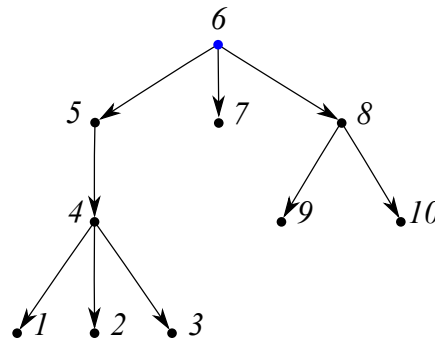


Рис. 47

В дереве \vec{T}_r вершина x называется родителем (parent) вершины y , а вершина y — сыном (child) вершины x , если существует ориентированное ребро (x, y) , ведущее из x в y . Так, на рис.47 вершина 5 является родителем для вершины 4, а вершина 4 — сыном для вершины 5.

Если существует ориентированный путь из вершины x в вершину z , то вершина x называется предком (ancestor) для вершины z , а вершина z — потомком (descendant) для вершины x . На рис.47 вершины 1, 2 и 3 являются потомками для вершины 5, а вершина 5 — их общим предком.

В случае, если наибольшая исходящая степень вершины в дереве равняется m , то такое дерево называется m -арным деревом. Так, дерево, показанное на рис.47, является 3-арным деревом. Часто в случае m -арных деревьев дополнительно подразумевается, что на множестве потомков произвольной вершины x задан некоторый порядок. Например, говоря про бинарные деревья,

часто подразумевают, что это деревья, в которых каждая из вершин может иметь как максимум два различных сына — левого и/или правого.

11.2.4. Выбор корня в дереве T задает на множестве $V(T)$ вершин этого дерева частичный порядок, превращая это множество в частично упорядоченное. В одной из следующих глав мы подробнее поговорим о таком множестве, а сейчас лишь дадим его определение.

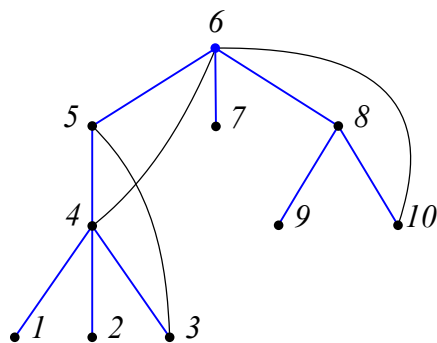
Определение 11.12. *Частично упорядоченным множеством* называется множество P (конечное или счетное) с введенным на нем бинарным отношением \preceq (так называемым *отношением частичного порядка*), удовлетворяющим следующим трем аксиомам:

1. Рефлексивность: для любого $x \in P$ справедливо $x \preceq x$.
2. Транзитивность: если $x \preceq y$ и $y \preceq z$, то $x \preceq z$.
3. Антисимметричность: если $x \preceq y$ и $y \preceq x$, то $x = y$.

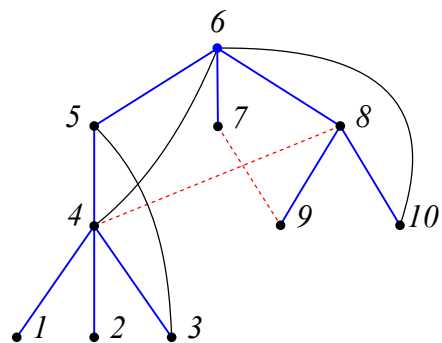
В случае корневого дерева мы будем считать, что $x \preceq y$, если x лежит на пути, соединяющем корень r дерева с вершиной y . Иными словами, это означает, что в соответствующем T_r ориентированном дереве \vec{T}_r вершина x является предком для вершины y . Кроме того, любые вершины, лежащие на пути, соединяющем корневую вершину с одним из его листов, сравнимы между собой, то есть, как говорят, образуют *цепь* в рассматриваемом частично упорядоченном множестве.

11.3. Давайте ненадолго вернемся к уже изученному нами алгоритму поиска в глубину в связном графе G .

11.3.1. Ранее мы показали, как с использованием данного алгоритма построить остовное дерево $T(G)$ графа G . Заметим сразу же, что в результате работы алгоритма мы, по сути, получаем не просто дерево, а корневое дерево — в качестве корня у нас выступает вершина, с которой мы запускаем наш алгоритм (вершина 1 на рис.44). Такое корневое дерево $T^*(G)$ обладает одним очень полезным свойством, о котором следует сказать несколько слов. Оказывается, любое такое дерево является *нормальным*.



(a) Нормальное остовное дерево



(b) Остовное дерево, нормальным не являющееся

Рис. 48

Определение 11.13. Остовное дерево $T^*(G)$ в связном графе G называется *нормальным*, если концевые вершины x, y любого ребра $e = \{x, y\}$, не принадлежащего дереву $T^*(G)$, сравнимы относительно частичного порядка \preceq , введенного на множестве вершин корневого дерева $T^*(G)$.

Иными словами, дерево $T^*(G)$ нормально, если один из концов x, y любого ребра, не принадлежащего этому дереву, лежит на единственном пути между вторым его концом и корнем дерева $T^*(G)$.

Рассмотрим в качестве примера остовное корневое дерево $T^*(G)$ для графа G , показанного на рис.48,а. В качестве корня дерева выбрана вершина 6, помеченная синим цветом на рисунке. Синим же цветом помечены ребра остовного дерева $T^*(G)$. Черные ребра — это ребра, не вошедшие в остовное дерево. Заметим, что в этом дереве имеется шесть листьев — вершины 1, 2, 3, 7, 9, 10. Каждый из этих листьев соединен с корневой вершиной 6 некоторым простым путем. Все вершины, лежащие на любом из этих путей, сравнимы между собой относительно частичного порядка \preceq . Заметим теперь, что концевые вершины всех черных ребер лежат на каком-то из этих путей, соединяющих один из листов с корнем. Так, ребро $\{3, 5\}$ лежит на единственном пути $\{3, 4, 5, 6\}$, соединяющем лист 3 дерева $T^*(G)$ с корнем — вершиной 6, ребро $\{4, 6\}$ лежит, к примеру, на пути $\{1, 4, 5, 6\}$, и так далее. Как следствие, такое остовное дерево является нормальным.

Рассмотрим теперь граф, показанный на рис.48,б. В таком графе помеченное синим цветом дерево $T^*(G)$ по-прежнему является остовным, однако нормальным оно уже не является. Действительно, рассмотрим ребра $\{7, 9\}$ и $\{4, 8\}$, помеченные красным цветом на рисунке. В дереве $T^*(G)$ отсутствует путь из листа в корень, который содержал бы обе концевые вершины ребра $\{7, 9\}$. Тот же факт имеет место и для ребра $\{4, 8\}$.

Давайте теперь поймем, почему дерево $T^*(G)$ для графа, показанного на рис.48,б, невозможно построить с помощью алгоритма поиска в глубину. Предположим, что, выйдя из корня, мы пришли вначале в вершину 5, а затем — в вершину 4. После обхода смежных с 4 вершин 1, 2, 3 мы обязаны пойти в еще одну смежную с 4 вершину — вершину 8. Следовательно, мы должны включить ребро $\{4, 8\}$ в строящееся остовное дерево. Наоборот, придя вначале в вершину 8, мы бы по тем же соображениям должны были бы зайти во все смежные с ней вершины, и в том числе в вершину 4. И тогда и в этом случае ребро $\{4, 8\}$ было бы ребром остовного дерева, строящегося поиском в глубину.

11.3.2. Отметим, однако, что построенное с помощью алгоритма поиска в глубину дерево $T^*(G)$ наряду с отмеченными выше достоинствами обладает и одним существенным недостатком. Именно, мы с вами знаем, что в любом дереве существует единственный путь между любыми двумя вершинами. В частности, в дереве $T^*(G)$ существует единственный путь между корневой вершиной и любой другой вершиной графа G . Длина этого пути совпадает с расстоянием между этими вершинами в дереве $T^*(G)$. Однако в исходном графе G расстояние между такими вершинами может быть меньше.

В качестве примера рассмотрим вершины 1 и 3 на рис.44. В построенном нами остовном дереве $T^*(G)$ расстояние между этими вершинами равно двум. При этом расстояние между этими же вершинами в исходном графе G меньше, чем в дереве $T^*(G)$ — в графе G это расстояние равно единице.

Итак, с помощью алгоритма поиска в глубину мы, помимо всего прочего, находим какие-то пути между корневой вершиной и оставшимися вершинами графа G , однако эти пути не обязательно оказываются кратчайшими с точки зрения путей в исходном графе G . В некоторых же приложениях нам необходимо получать такое остовное дерево $\tilde{T}(G)$, в котором расстояние между корневой вершиной и оставшимися вершинами графа G совпадает с расстоянием между этими вершинами в исходном графе, то есть дерево, состоящее из кратчайших путей

между такими вершинами. Алгоритм, позволяющий такого рода дерево получать, называется *алгоритмом поиска в ширину*.

При построении алгоритма нам понадобится дополнительная абстрактная структура данных, называемая очередью Q . В такой структуре данных элементы линейно упорядочены, то есть образуют линейный список, добавлять элементы мы можем по одному в конец этого списка, а удалять — также по одному, но с начала этого списка. Такого рода структуру данных иногда еще называют структурой данных вида First-In-First-Out (первым вошел — первым вышел: первый элемент, добавленный в очередь, первым же из нее выйдет).

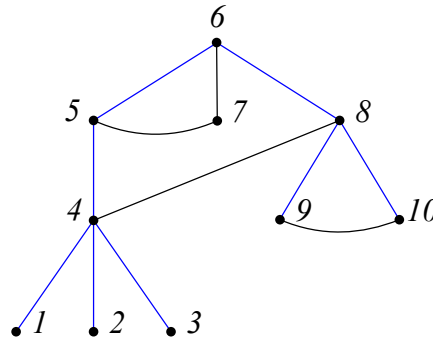


Рис. 49

Работа алгоритма начинается с того, что мы помещаем в очередь Q первую вершину графа — будущую корневую вершину строящегося остовного дерева $T(G)$ (вершина 6 на рис.49). Одновременно с этим действием мы будем считать, что эта вершина окрашивается в какой-то новый цвет, например, в красный. Теперь на каждом шаге алгоритма мы будем совершать следующие действия: вынимать вершину x из очереди Q , смотреть на все смежные с x вершины, и те из них, которые окрашены в черный цвет, помещать в очередь, одновременно окрашивая их в красный цвет. Алгоритм закончится тогда, когда в очереди Q никаких вершин не останется. Соответствующий описанному алгоритму псевдокод можно записать следующим образом:

Процедура $BFS(x,G)$, x - стартовая вершина графа G :

// Инициализация начальных значений

for (y : y - вершина графа G)

begin

color[y] := 0;

parent[y] := -1;

dist[y] := 0;

end

// Помещаем в очередь корневую вершину

color[x] := 1;

$Q.push(x)$;

// Запуск алгоритма поиска в ширину

while (Q не пусто)

begin

$y := Q.pop$;

for (z : (y,z) принадлежит $E(G)$)

begin

```

    if ( color[z] == 0)
      begin
        color[z] := 1;
        Q.push(z);
        dist[z] := dist[y] + 1;
        parent[z] := y;
      end
    end
  end
end

```

В этом алгоритме мы дополнительно ввели массив `dist[]` кратчайших расстояний от заданной вершины до корня, а также массив `parent[]`, позволяющий для любой вершины определить единственного его родителя в строящемся остоном дереве.

Результат работы алгоритма для дерева, показанного на рис.49, можно свести в следующую таблицу:

Обрабатываемая вершина	Содержание очереди
	[6]
6	[8, 7, 5]
5	[4, 8, 7]
7	[4, 8]
8	[10, 9, 4]
4	[3, 2, 1, 10, 9]
9	[3, 2, 1, 10]
10	[3, 2, 1]
1	[3, 2]
2	[3]
3	[]

Дадим в заключение физическую интерпретацию остоного дерева в графе, полученного алгоритма поиска в ширину. Если считать вершины графа шариками, а ребра — соединяющими эти шарики пружинками, то, подвесив граф за выбранный шарик, мы получим картинку подобную изображенной на рис.49. При этом все шарики (вершины графа) распределятся по слоям — на нулевом слое будет корневая вершина, на следующем — вершины, смежные с корневой, и так далее.

12 Перечисление деревьев. Формула Cayley

12.1. Перейдем теперь к задаче перечисления всех (помеченных) деревьев, построенных на n вершинах.

12.1.1. Несложно убедиться, что существует по одному дереву, построенному на одной и на двух вершинах, а также три дерева, построенные на трех вершинах (см.рис.50). Все эти три дерева отличаются друг от друга только расположением меток вершин. В случае $n = 4$ таких деревьев существует уже 16 штук — 12 деревьев, отвечающих линейной цепочке вершин, и

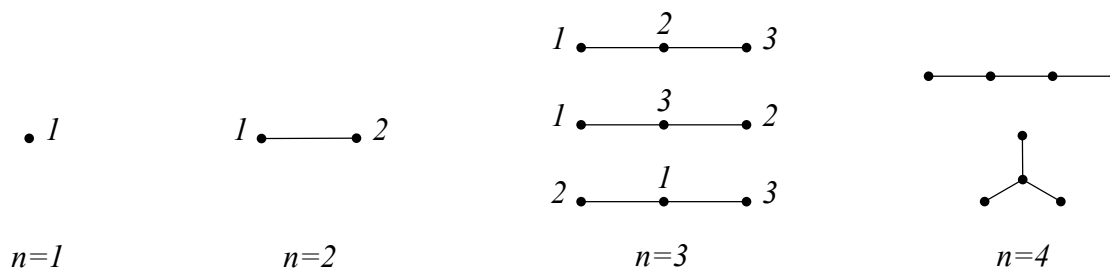


Рис. 50: Неизоморфные друг другу деревья для $n \leq 4$

4 дерева, построенных на “треножнике”. Задача этого пункта — доказать следующий общий результат, известный как формула Кэли.

Теорема 12.1 (Формула Кэли). *Количество a_n различных помеченных деревьев на n вершинах равно n^{n-2} .*

12.1.2. К настоящему моменту известно около шестнадцати различных способов доказательства этого результата (см., например, [?]). Метод, связанный с производящими функциями, мы рассмотрим в курсе комбинаторики. Здесь же мы приведем доказательство этой формулы, впервые предложенное Хайнсом Прюфером (Heinz Prüfer) в 1918 году. Это доказательство является, пожалуй, наиболее простым и понятным, и входит практически в любой учебник по комбинаторике и теории графов. В основе доказательства лежит идея построения по данному дереву некоторой строки длины $n - 2$ над алфавитом из n чисел, по которой можно однозначно восстановить исходное дерево.

12.1.3. Рассмотрим дерево T , вершины которого помечены элементами множества $[n]$, $n \geq 2$, и покажем вначале, как сопоставить такому дереву числовую последовательность вида

$$P(T) = (y_1, y_2, \dots, y_{n-2}), \quad y_i \in [n],$$

называемую кодом Прюфера. Для этого на первом шаге выберем среди всех листьев дерева T вершину x_1 с минимальным номером, удалим эту вершину вместе с инцидентным ей ребром $e_1 = \{x_1, y_1\}$, а затем запишем смежную с удаленным листом вершину y_1 в качестве первого символа в нашем коде. Затем возьмем дерево T_1 , полученное из T удалением вершины x_1 , и выберем среди его листьев вершину x_2 с минимальным номером. Удалим этот лист вместе с инцидентным ему ребром $\{x_2, y_2\}$, а в код Прюфера запишем инцидентную x_2 вершину y_2 . Будем повторять этот процесс до тех пор, пока у нас не останется дерево, состоящее ровно из двух вершин и соединяющего их ребра.

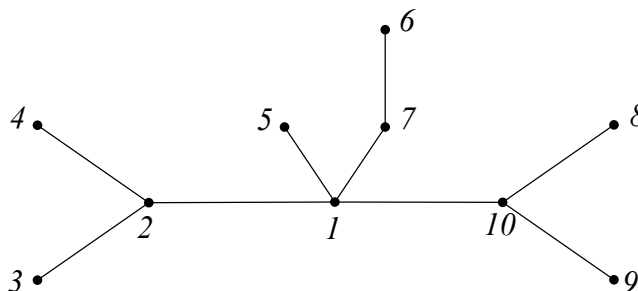


Рис. 51: Дерево T на десяти вершинах

Продемонстрируем этот алгоритм на дереве T , показанном на рис.51. Среди всех его листьев минимальный номер имеет вершина $x_1 = 3$. Удалим эту вершину 3 вместе с инцидентным ей

ребром $e_1 = \{3, 2\}$, а в последовательность Прюфера добавим смежную с ней вершину y_1 , то есть вершину 2. Среди листьев дерева T_1 , полученного из T удалением вершины 3, минимальный номер имеет вершина $x_2 = 4$. Ее удаление добавляет нам второй член $y_2 = 2$ последовательности Прюфера. Продолжая этот процесс, на заключительном, 8-м шаге мы получаем дерево, построенное на вершинах 10 и 9, а также код Прюфера

$$P(T) = (2, 2, 1, 1, 7, 1, 10, 10).$$

Нам осталось доказать, что описанное выше отображение, сопоставляющее любому помеченному дереву T числовую последовательность $P(T)$, взаимно-однозначно. Для этого нам достаточно показать, что по любому коду Прюфера $P(T)$ мы сможем однозначно восстановить исходное дерево T .

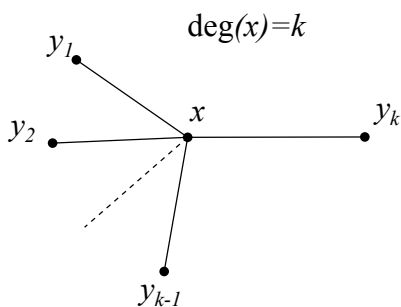


Рис. 52

12.1.4. Заметим, прежде всего, что листья исходного дерева T в этом коде вовсе не встречаются. Действительно, удаляя лист, мы записываем в последовательность Прюфера не номер листа, а номер смежной с ним вершины. Рассмотрим теперь более общий случай вершины x , степень $\deg(x) = k$ которой строго больше единицы (рис.52). В процессе построения кода Прюфера мы постепенно удаляем смежные с x вершины y_i , $i = 1, \dots, k$, до тех пор, пока степень вершины x не станет равной единице. На этом шаге сама вершина x стала листом, и, если нам придется удалить вершину x , то мы уже в последовательность Прюфера включим не вершину x , а смежную с ней вершину y_k . Таким образом, любая вершина x встречается в коде Прюфера $P(T)$ ровно $\deg_T(x) - 1$ раз.

12.1.5. Давайте теперь подумаем, что в идеале нам требуется, чтобы восстановить дерево на n вершинах. Если бы мы в процессе удаления вершины записывали бы не инцидентную ей вершину, а пару вершин — смежную и удаляемую, то мы легко бы по такому набору пар вершин восстановили исходное дерево. Действительно, рассмотрим пару последовательностей

$$\begin{array}{cccccccc} 2 & 2 & 1 & 1 & 7 & 1 & 10 & 10 \\ 3 & 4 & 2 & 5 & 6 & 7 & 1 & 8 \end{array}$$

для показанного на рис.51 дерева T . Первая из них — это последовательность Прюфера $P(T)$. Вторая последовательность — это список удаляемых на каждом шаге листьев. Зная эти пары чисел, мы знаем все ребра исходного дерева T кроме, разве что, последнего ребра. Однако и его мы легко восстановим. Действительно, последним у нас всегда остается ребро, соединяющее вершину, имеющую наибольший номер n , с некоторой вершиной z . Вершина z — это единственная вершина, отличная от вершины с максимальным номером, не встречающаяся во второй последовательности — последовательности удаляемых вершин. Так, в случае дерева, показанного на рис.51, у нас в конце остается ребро, соединяющее вершину 10 с вершиной $z = 9$, не встречающейся во второй строчке таблицы и отличной от 10.

Итак, все, что нам не хватает — это восстановить по верхней последовательности $P(T)$ — коду Прюфера — соответствующую ей нижнюю последовательность вершин, удаляемых на соответствующем шаге алгоритма:

$$P(T) = (y_1, y_2, \dots, y_{n-3}, y_{n-2}) \\ (x_1, x_2, \dots, x_{n-3}, x_{n-2}) + \{x_{n-1}, x_n \equiv n\}.$$

Так как любую вершину в процессе построения кода Прюфера $P(T)$ мы удаляем лишь однажды, то все числа в этой последовательности $\{x_i\}$ различны. Кроме того, в эту последовательность не входит вершина $x_n \equiv n$ с максимальным номером, а также вершина x_{n-1} , которая остается соединенной с n ребром на последнем шаге построения последовательности Прюфера.

12.1.6. Начнем строить последовательность $\{x_i\}$ с восстановления числа x_1 , то есть с восстановления первой удаленной нами вершины. Напомним, что на первом шаге мы брали листья дерева, выбирали из них вершину с минимальным номером и ее удаляли. С точки зрения кода Прюфера листья — это все числа, не вошедшие в последовательность $P(T) = \{y_i\}$. Минимальное из этих чисел и будет нужным нам числом x_1 . Так, для дерева, показанного на рис.51, в код Прюфера $P(T)$ не входят числа

$$\{3, 4, 5, 6, 8, 9\}.$$

Число 3 является минимальным среди этих чисел. Следовательно, именно вершину под номером 3 мы и удалили из дерева T на первом шаге построения кода Прюфера $P(T)$ для дерева T . В общем же случае мы рассматриваем множество чисел вида

$$X = \{1, 2, \dots, n - 1, n\},$$

исключаем из него числа, входящие в последовательность $P(T)$, выбираем из оставшихся минимальное — число x_1 , и записываем его под числом y_1 :

$$(y_1, y_2, \dots, y_{n-3}, y_{n-2}) \\ (x_1,$$

Тем самым мы находим первое ребро — ребро $\{x_1, y_1\}$ — восстанавливаемого нами дерева.

12.1.7. Теперь нам нужно восстановить число x_2 , то есть вершину, удаленную на втором шаге построения кода Прюфера $P(T)$. На этом шаге мы рассматривали поддерево T_1 , полученное из T удалением листа x_1 , и в этом поддереве снова выбирали лист с минимальным номером. С точки зрения кода Прюфера поддереву T_1 отвечает числовая последовательность, полученная из $P(T)$ удалением первого члена y_1 :

$$P(T_1) = (y_2, y_3, \dots, y_{n-3}, y_{n-2}).$$

Действительно, удаляя в дереве T вершину x_1 , мы удаляем и соединяющее ее с вершиной y_1 ребро, а следовательно, уменьшаем степень y_1 на единицу. Но тогда мы должны уменьшить на единицу и вхождение y_1 в код Прюфера. Этого мы и достигаем, удаляя из последовательности $P(T)$ число y_1 , стоящее в этой последовательности на первом месте.

А теперь для восстановления x_2 мы должны провести те же рассуждения, что и на предыдущем шаге. Именно, у нас имеется множество $X \setminus \{x_1\}$ оставшихся после первого шага чисел. Исключим из него числа, входящие в код Прюфера $P(T_1)$ поддерева T_1 , а затем выберем среди оставшихся чисел минимальное. Выбранное число x_2 и будет соответствовать вершине, которую

мы удалили на втором шаге при построении кода Прюфера исходного дерева T . Записывая его под числом y_2 в коде Прюфера, мы восстанавливаем второе ребро дерева T :

$$\begin{aligned} &(y_1, y_2, \dots, y_{n-3}, y_{n-2}) \\ &(x_1, x_2, \end{aligned}$$

Так, в рассматриваемом примере у нас после первого шага осталось множество чисел

$$X_1 := X \setminus \{x_1\} = \{1, 2, 4, 5, 6, 7, 8, 9, 10\}.$$

Исключим из них числа, входящие в обрезанный слева на единицу код Прюфера

$$P(T_1) = (2, 1, 1, 7, 1, 10, 10).$$

Среди оставшихся в множестве X_1 чисел минимальным является число 4. Следовательно, удаленным на втором шаге оказалось ребро $\{x_2, y_2\} = (4, 2)$ дерева T .

12.1.8. Теперь мы можем уже описать и общий шаг алгоритма восстановления дерева T по коду Прюфера $P(T)$. Предположим, что мы уже восстановили ребра $\{x_1, y_1\}, \dots, \{x_{k-1}, y_{k-1}\}$ дерева T . Рассмотрим множество чисел

$$X_{k-1} := X \setminus \{x_1, x_2, \dots, x_{k-1}\}.$$

Исключим из этого множества все числа, встречающиеся в коде Прюфера $P(T_{k-1})$, полученном из исходного кода Прюфера $P(T)$ удалением стоящих на первых $(k-1)$ позициях чисел y_1, \dots, y_{k-1} . Выбирая из оставшихся в множестве X_{k-1} чисел минимальное число x_k и записывая его под числом y_k , мы тем самым восстанавливаем ребро $\{x_k, y_k\}$ дерева T , отвечающего последовательности Прюфера $P(T)$:

$$\begin{aligned} &(y_1, y_2, \dots, y_k, \dots, y_{n-3}, y_{n-2}) \\ &(x_1, x_2, \dots, x_k, \end{aligned}$$

12.1.9. Продолжая этот процесс, мы восстановим все числа x_k , $k = 1, \dots, (n-2)$, то есть $n-2$ ребра $\{x_i, y_i\}$ дерева T , $i = 1, \dots, n$. Оставшееся ребро $\{x_{n-1}, n\}$ отвечает двум не вошедшим в набор $\{x_i\}$ числам x_{n-1} и n . Последовательно присоединяя к ребру $\{x_{n-1}, n\}$ ребра $\{x_{n-2}, y_{n-2}\}$, $\{x_{n-3}, y_{n-3}\}$, \dots , мы построим в конечном итоге дерево T , отвечающее коду Прюфера $P(T)$.

В рассматриваемом примере мы по коду Прюфера

$$P(T) = (2, 2, 1, 1, 7, 1, 10, 10)$$

восстанавливаем числа

$$(x_1, \dots, x_8) = (3, 4, 2, 5, 6, 7, 1, 8),$$

а следовательно, 8 ребер дерева T . Добавляя эти ребра к ребру $\{9, 10\}$, мы в результате получаем дерево T , показанное на рис.51.

13 Подсчет остовных деревьев в графе. Матричная теорема о деревьях

13.1. Доказанную в предыдущем параграфе формулу Кэли мы можем также интерпретировать как формулу подсчета количества всех остовных деревьев в полном графе K_n . Основная задача

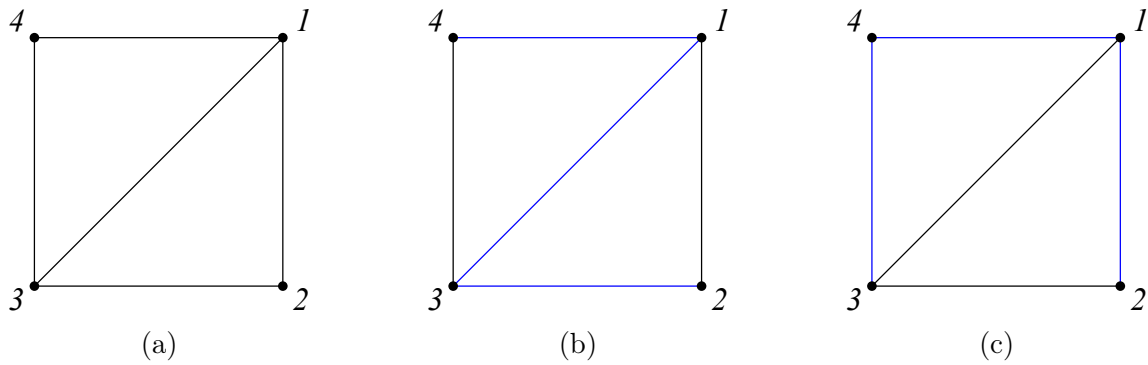


Рис. 53

данного параграфа — научиться решать более общую задачу подсчета количества $t(G)$ всех различных остовных деревьев для произвольного связного графа G .

13.1.1. Начнем с примера графа G , показанного на рис.53,а. У такого графа имеется восемь различных остовных деревьев — четыре из них проходят через диагональ этого графа (см.рис.53,б), а четыре охватывают внешний цикл графа G (рис.53,с).

В рассмотренном примере мы все множество остовных деревьев разбили на два блока, один из которых содержит деревья, проходящие через выбранное ребро e , а второй — деревья, через e не проходящие. Подобного рода рассуждения лежат в основе рекурсивной процедуры подсчета количества $t(G)$ всех остовных подграфов графа G , основанной на следующем утверждении.

Утверждение 13.1. Пусть e — ребро, не являющееся петлей графа G . Тогда

$$t(G) = t(G - e) + t(G \setminus e), \quad (42)$$

где $G - e$ — граф, полученный из G удалением ребра e , а $G \setminus e$ — граф, полученный из G стягиванием этого ребра.

Доказательство. Все множество остовных деревьев графа G разбивается на два блока — блок деревьев, содержащих ребро e , и блок деревьев, e не содержащих. Очевидно, что все остовные деревья, которые не содержат ребро e , являются остовными деревьями графа $G - e$. Поэтому количество таких деревьев совпадает с количеством $t(G - e)$. Давайте теперь докажем, что количество остовных деревьев, содержащих e , совпадает с количеством остовных деревьев графа $G \setminus e$.

Действительно, рассмотрим остовное дерево T графа G , содержащее ребро e . Стягивание этого ребра уменьшает количество ребер и количество вершин в графе $T \setminus e$ на единицу. Кроме того, эта операция не приводит к потере связности получающегося графа $G \setminus e$. Следовательно, граф $T \setminus e$ вновь является деревом — остовным деревом графа $G \setminus e$. Кроме того, никакие два различных остовных дерева графа G , содержащих e , не сольются в одно в результате этой операции. Обратно, любое остовное дерево в графе $G \setminus e$ превращается в остовное дерево графа G , содержащее e , расщеплением вершины на две и добавлением между ними ребра. \square

13.1.2. Сделаем несколько замечаний, касающихся вычислительных аспектов расчета $t(G)$ по формуле (42). Прежде всего, даже если мы стартуем с простого связного графа, то в процессе удаления ребер граф может перестать быть связным, а в процессе стягивания ребер в графе могут появиться петли и мультиребра. Так, например, после стягивания ребра e_1 у графа G , показанного на рис.54,а, ребро e_2 , связывающее в исходном графе G те же вершины, что и e_1 , превращается в петлю.



Рис. 54: Операция стягивания ребра

Со случаем несвязных графов справиться просто — у таких подграфов остовных деревьев не существует, поэтому для таких графов $t(G) = 0$. Мультиребра в графе не мешают нам пользоваться рекуррентным соотношением (42). Однако при появлении петель нам нужно быть осторожнее — рекуррентное соотношение (42) перестает работать в случае, когда ребро e представляет собой петлю. Рассмотрим, к примеру, граф, состоящий из одиночной вершины и петли. У такого графа имеется единственное остовное дерево, состоящее из изолированной вершины. Формула же (42) применительно к данному случаю даст нам $t(G) = 2$. Как следствие, в процессе подсчета по формуле (42) нам нужно такие петли либо просто игнорировать, либо сразу удалять в процессе их появления — на количество остовных деревьев это никак не повлияет.

Наконец, рекурсивный подсчет по формуле (42) требует задания начальных условий. В процессе удаления и стягивания ребер мы в конце концов получим графы, любое ребро которых представляет собой петлю. Если в полученном графе при этом останется одна вершина, то количество остовных деревьев в таком графе равно единице. В противном случае $t(G) = 0$.

13.1.3. Полученное нами рекуррентное соотношение (42) приводит к достаточно громоздким вычислениям даже для случая графов, имеющих не очень большое количество m ребер. С увеличением m количество вычислений экспоненциально возрастает. К счастью, наряду с (42) для подсчета $t(G)$ имеется замкнутая аналитическая формула, полученная еще в 1847 году Кирхгофом. Эта формула носит название матричной теоремы о деревьях (Matrix Tree Theorem) и выражает $t(G)$ через определитель некоторой матрицы. Определитель же квадратной матрицы размером $n \times n$ мы можем сосчитать, используя менее чем n^3 операций.

Рассмотрим связный граф G без петель, построенный на множестве вершин $[n] = \{1, 2, \dots, n\}$. Мы знаем, что матрица M_a смежности такого графа симметрична, и все диагональные элементы этой матрицы равны нулю. Возьмем теперь матрицу M_d размером $n \times n$, в которой диагональные элементы $d_{i,i}$ равны степеням $\deg(i)$ соответствующих вершин в графе G , а остальные элементы равны нулю, и вычтем из нее матрицу M_a . В результате получим матрицу $L = M_d - M_a$, которая носит название *матрицы Кирхгофа*.

Теорема 13.2 (Matrix tree theorem). Пусть x_k есть произвольная вершина графа G , а L^* есть матрица, полученная из матрицы Кирхгофа L удалением k -й строки и k -го столбца, отвечающих вершине x_k . Тогда количество $t(G)$ остовных деревьев графа G равно определителю матрицы L^* :

$$t(G) = \det(L^*).$$

Прежде чем доказывать эту теорему, посмотрим, как она работает для примера графа G , по-

казанного на рис.53,а. Для такого графа матрица Кирхгофа имеет вид

$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix}.$$

Удаляя из нее первую строку и первый столбец, мы получаем матрицу

$$L^* = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{pmatrix},$$

определитель которой равен 8.

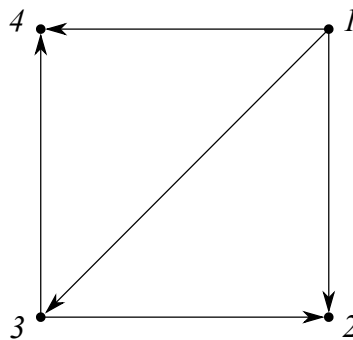


Рис. 55

13.1.4. При доказательстве матричной теоремы о деревьях удобно перейти от неориентированного графа G к некоторому ориентированному графу D , полученному произвольной ориентацией ребер графа G (см.рис.55). Оказывается, что при этом имеется достаточно простая связь между матрицей Кирхгофа L и матрицей M_i инцидентности орграфа D . Именно, справедлива следующая

Лемма 13.3. Пусть D есть орграф, полученный из G произвольной ориентацией ребер графа G , а M_i есть матрица инцидентности орграфа D . Тогда

$$L = M_i \cdot M_i^T.$$

Доказательство. Напомним, что в матрице M_i элемент $m_{i,j} = 1$, если ребро e_j выходит из вершины i , $m_{i,j} = -1$ в случае, если e_j входит в вершину i , и $m_{i,j} = 0$ в остальных случаях. Любой элемент $l_{i,j}$ матрицы $M_i \cdot M_i^T$ есть скалярное произведение i -й и j -й строк матрицы M_i .

В случае $i \neq j$ это скалярное произведение дает нам столько -1 , сколько имеется ребер, соединяющих вершины i и j . Действительно, если $m_{i,k} = 1$ и $m_{j,k} = -1$, то это означает, что в графе G имеется ребро e_k , соединяющее вершины i и j . В противном случае хотя бы один из этих сомножителей равен нулю, поэтому слагаемое $m_{i,k} \cdot m_{j,k} = 0$ и в скалярное произведение вклада не дает.

В случае $i = j$ мы имеем $m_{i,k}^2$, что дает единицу в случае, если ребро e_k инцидентно вершине i . Суммируя по всем k , мы получаем на диагонали число $l_{i,i}$, равное степени вершины i . \square

Следствие 13.4. Матрица L^* , полученная из матрицы Кирхгофа L удалением k -й строки и k -го столбца, равна

$$L^* = M_i^* \cdot (M_i^*)^T,$$

где M_i^* есть матрица, полученная из M_i удалением k -й строки.

Для ориентированного графа D , показанного на рис.55, матрица \mathbf{M}_i имеет вид

$$\mathbf{M}_i = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 & -1 \end{pmatrix},$$

а матрица \mathbf{M}_i^* , полученная в результате удаления первой строки, равна

$$\mathbf{M}_i^* = \begin{pmatrix} -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 & -1 \end{pmatrix}.$$

13.1.5. Приступим теперь к доказательству матричной теоремы о деревьях. Пусть B есть квадратная подматрица $(n-1) \times (n-1)$ матрицы \mathbf{M}_i^* . Пусть также G есть неориентированный граф, полученный из D заменой любого ориентированного ребра на неориентированное (underlying graph). Столбцы подматрицы B отвечают какому-то набору из $n-1$ ориентированных ребер в орграфе D . Соответствующие этому набору неориентированные ребра вместе с инцидентными им вершинами образуют в underlying graph G некоторый подграф H . Так, столбцы подматрицы

$$B = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

полученной из матрицы \mathbf{M}_i^* орграфа, показанного на рис.55, удалением последних двух столбцов, индуцируют в графе G на рис.53,а подграф

$$H = \{\{1, 2\}, \{1, 3\}, \{1, 4\}\}.$$

Покажем, что определитель подматрицы B равен ± 1 в случае, если отвечающий этой подматрице подграф H образует остовное дерево графа G , и нулю в противном случае.

Предположим вначале, что подграф H представляет собой остовное дерево. В этом случае в H обязательно найдутся хотя бы два листа. Так как в подматрице B отсутствует только лишь одна строка исходной матрицы \mathbf{M}_i инцидентности орграфа D , то в ней обязательно найдется хотя бы одна строка j , отвечающая вершине x подграфа H степени 1. В этой строке имеется единственный ненулевой элемент $m_{j,k} = \pm 1$. Раскладывая определитель матрицы B по j -й строке, мы получаем равенство вида

$$\det B = m_{j,k} \cdot \det B' = \pm 1 \cdot \det B',$$

где B' — подматрица, полученная из B вычеркиванием j -й строки и k -го столбца. На языке подграфа H данная операция отвечает удалению вершины x степени 1 вместе с инцидентным ей ребром. В результате мы получаем подграф $H - x$, который вновь является остовным деревом, но уже графа $G - x$. Продолжая описанный процесс далее, мы на последнем шаге придем к остовному дереву, построенному на двух вершинах, для которого $\det B = \pm 1$.

Так, в рассматриваемом примере все три вершины 2, 3, 4 подграфа H , построенного на ребрах $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, являются листьями. Выбирая в качестве x вершину 2, мы получаем из B матрицу

$$B' = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix},$$

отвечающую подграфу $H - \{2\}$, индуцированному ребрами $\{1, 3\}$ и $\{1, 4\}$. Удаляя в подграфе $H - \{2\}$ вершину 3, мы получаем матрицу

$$B'' = (-1),$$

отвечающую дереву $\{1, 4\}$. Определитель данной матрицы равен -1 , а следовательно, -1 равен и определитель исходной матрицы B .

Теперь допустим, что соответствующий подматрице B подграф H не является остовным деревом. Следовательно, в таком подграфе обязательно найдется цикл C . Например, если для орграфа D , показанного на рис.55, выбрать в матрице \mathbf{M}_i^* первый, второй и четвертый столбцы, то мы получим матрицу

$$B = \begin{pmatrix} -1 & 0 & -1 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

отвечающую подграфу H , индуцированному ребрами $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$. В данном случае сам подграф H представляет собой цикл в графе G на рис.53,а.

Покажем, что в таком случае обязательно найдется нетривиальная линейная комбинация столбцов, равная нулю. Отсюда будет следовать, что строки матрицы линейно зависимы, то есть что $\det B = 0$. Рассмотрим вначале ребра, не входящие в цикл C . Соответствующие им столбцы матрицы B включим в линейную комбинацию с коэффициентами, равными нулю. Предположим теперь, что направление всех ребер, входящих в цикл, совпадает с каким-то из направлений обхода цикла C . В этом случае линейная комбинация соответствующих этим ребрам столбцов равнялась бы нулю (сколько ребер вошло в вершину x цикла, столько же ребер из нее вышло). Так, если бы у нас вместо ребра $(1, 2)$ в ориентированном подграфе, отвечающем H , было бы ребро $(2, 1)$, то вместо подматрицы B мы бы имели подматрицу

$$B' = \begin{pmatrix} 1 & 0 & -1 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

линейная комбинация столбцов которой в точности равнялась бы нулю. Однако в цикле C могут найтись и ребра, которые направлены в сторону, противоположную выбранному направлению обхода цикла C (ребро $(1, 2)$ в орграфе D на рис.55). Тогда мы можем включить отвечающие этим ребрам столбцы (первый столбец подматрицы B) в линейную комбинацию с коэффициентами, равными -1 , и вновь получим линейную комбинацию столбцов, равную нулевому столбцу.

Для завершения доказательства теоремы нам теперь достаточно воспользоваться формулой Бине-Коши, позволяющей вычислять определитель квадратной матрицы, являющейся произведением двух прямоугольных матриц. Согласно этой формуле,

$$\det (\mathbf{M}_i^* \cdot (\mathbf{M}_i^*)^T) = \sum (\det B) \cdot (\det B^T) = \sum (\det B)^2,$$

где суммирование проводится по всем $(n - 1) \times (n - 1)$ -подматрицам B матрицы \mathbf{M}_i^* . Мы только что показали, что $(\det B)^2 = 1$ в случае, если B отвечает остовному поддереву графа G , и $(\det B)^2 = 0$ в противном случае. Тем самым матричная теорема о деревьях доказана. \square