Содержание

Пример работы с файлами.

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. `http://acm.math.spbu.ru/~sk1/algo/sum/`

Пример работы с файлами.

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:
`http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html`
Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:
`http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html`
`http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html`

Выделение памяти.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу:
`http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html`

Обязательные задачи

Задача 12A. Здоровье Графа [1 sec, 256 mb]

Этично ли удалять рёбра у связанного графа?

Граф Безциклов решил проверить своё здоровье. Он хочет проверить, что все его рёбра достаточно крепко держатся в нём. Для этого он хочет посчитать *устойчивость* некоторых из них. *Устойчивостью* ребра называется количество простых путей, проходящих через это ребро.

Формат входных данных

Все числа в файле целые.
$0 \leqslant N \leqslant 10^5$, $0 \leqslant M \leqslant 10^5$ — количество вершин и рёбер.
Затем M пар чисел $1 \leqslant v_i, u_i \leqslant N$ — i-ое ребро соединяет вершины v_i и u_i.
$0 \leqslant Q \leqslant 10^5$ — количество запросов.
Затем Q чисел $1 \leqslant e_i \leqslant M$.
Граф неориентирован. Гарантируется, что Граф ацикличен.

Формат выходных данных

Для i-ого запроса вывести устойчивость e_i-ого ребра.

Примеры

health.in	health.out
2 1 1 2 1 1	1

Замечание

Обычная динамика по дереву.

Задача 12B. Расстояние от корня [1 sec, 256 mb]

В заданном корневом дереве найдите вершины, максимально удалённые от корня. Расстоянием между вершинами считается количество рёбер в пути.

Формат входных данных

В первой строке задано n — количество вершин в дереве ($1 \leqslant n \leqslant 100$). В следующих $n-1$ строках заданы вершины, являющиеся предками вершин 2, 3, ..., n. Вершина 1 является корнем дерева.

Формат выходных данных

В первой строке выведите максимальное расстояние от корня до остальных вершин дерева. Во второй строке выведите, сколько вершин дерева находятся от корня на таком расстоянии. В третьей строке выведите номера этих вершин через пробел в порядке возрастания.

Примеры

rootdist.in	rootdist.out
3 1 1	1 2 2 3
3 1 2	2 1 3

Замечание

Обычная динамика по дереву.

Задача 12C. Любители Кошек [1 sec, 256 mb]

В университетском клубе любителей кошек зарегистрировано n членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

Формат входных данных

В первой строке входного файла заданы числа n и m ($1 \leqslant n \leqslant 1\,000$, $1 \leqslant m \leqslant 30\,000$), где m обозначает общее число знакомств. В последующих m строках идут пары чисел a_i b_i, обозначающие, что a_i знаком с b_i. Информация об одном знакомстве может быть записана несколько раз, причем даже в разном порядке (как (x, y), так и (y, x)).

Формат выходных данных

В выходной файл необходимо вывести количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

Пример

catlover.in	catlover.out
3 3 1 2 2 3 3 1	1

Замечание

Предполагается решение за $\mathcal{O}(nm)$. Если каждый треугольник сразу считать ровно один раз, получится быстрее.

Задача 12D. Path. Кратчайший путь [1 sec, 256 mb]

Дан взвешенный ориентированный граф и вершина s в нем. Требуется для каждой вершины u найти длину кратчайшего пути из s в u.

Формат входных данных

Первая строка входного файла содержит n, m и s — количество вершин, ребер и номер выделенной вершины соответственно ($2 \leqslant n \leqslant 2\,000$, $1 \leqslant m \leqslant 6\,000$).

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее 10^{15} по модулю. В графе могут быть кратные ребра и петли.

Формат выходных данных

Выведите n строк — для каждой вершины u выведите длину кратчайшего пути из s в u, '*' если не существует путь из s в u и '-' если не существует кратчайший путь из s в u.

Пример

path.in	path.out
6 7 1 1 2 10 2 3 5 1 3 100 3 5 7 5 4 10 4 3 -18 6 1 -1	0 10 - - - *

Замечание

Форд-Беллман. Будьте аккуратны с переполнениями.

Задача 12E. Avia. Авиаперелеты [1 sec, 256 mb]

Главного конструктора Петю попросили разработать новую модель самолета для компании «Air Бубундия». Оказалось, что самая сложная часть заключается в подборе оптимального размера топливного бака.

Главный картограф «Air Бубундия» Вася составил подробную карту Бубундии. На этой карте он отметил расход топлива для перелета между каждой парой городов.

Петя хочет сделать размер бака минимально возможным, для которого самолет сможет долететь от любого города в любой другой (возможно, с дозаправками в пути).

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leqslant n \leqslant 1000$) — число городов в Бубундии. Далее идут n строк по n чисел каждая. j-ое число в i-ой строке равно расходу топлива при перелете из i-ого города в j-ый. Все числа не меньше нуля и меньше 10^9. Гарантируется, что для любого i в i-ой строчке i-ое число равно нулю.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — оптимальный размер бака.

Пример

avia.in	avia.out
4 0 10 12 16 11 0 8 9 10 13 0 22 13 10 17 0	10

Задача 12F. Кодовый замок [1 sec, 256 mb]

Петя опоздал на тренировку по программированию! Поскольку тренировка проходит в воскресенье, главный вход в учебный корпус, где она проходит, оказался закрыт, а вахтёр ушёл куда-то по своим делам. К счастью, есть другой способ проникнуть в здание — открыть снаружи боковую дверь, на которой установлен кодовый замок.

На пульте замка есть d кнопок с цифрами $0, 1, \ldots, d-1$. Известно, что код, открывающий замок, состоит из k цифр. Замок открывается, если последние k нажатий кнопок образуют код.

Поскольку Петя не имеет понятия, какой код открывает замок, ему придётся перебрать все возможные коды из k цифр. Но, чтобы как можно скорее попасть на тренировку, нужно минимизировать количество нажатий на кнопки. Помогите Пете придумать такую последовательность нажатий на кнопки, при которой все возможные коды были бы проверены, а количество нажатий при этом оказалось бы минимально возможным.

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа d и k — количество кнопок на пульте и размер кода, соответственно ($2 \leqslant d \leqslant 10$, $1 \leqslant k \leqslant 20$).

Формат выходных данных

В первой строке выходного файла выведите искомую последовательность. Если последовательностей минимальной длины, перебирающих все возможные коды, несколько, можно выводить любую из них. Гарантируется, что d и k таковы, что минимальная длина последовательности не превосходит 1 мебибайта.

Пример

codelock.in	codelock.out
2 3	0001011100

Пояснение к примеру

Последовательность в примере перебирает все коды длины 3 в следующем порядке: 000, 001, 010, 101, 011, 111, 110, 100.

Задача 12G. Из истории банка Гринготтс [1 sec, 256 mb]

Чтобы понять название задачи, можно прочитать красивую легенду.
http://acm.timus.ru/problem.aspx?space=1&num=1441

Задача же заключается в том, чтобы рёбра неориентированного графа разбить на минимальное число путей.

Формат входных данных

Дан граф. На первой строке число вершин n ($1 \leqslant n \leqslant 20\,000$) и число рёбер m ($1 \leqslant m \leqslant 20\,000$). Следующие m строк содержат описание рёбер графа. Каждая строка по два числа a_i b_i ($1 \leqslant a_i, b_i \leqslant n$). Между каждыми двумя вершинами не более одного ребра. Граф связен.

Формат выходных данных

На первой строке минимальное число путей. На каждой следующей описание очередного пути – номера вершин в порядке прохождения.

Примеры

euler.in	euler.out
7 7 1 2 4 1 6 7 5 7 7 4 2 3 4 2	3 5 7 4 2 1 4 2 3 6 7

Дополнительные задачи

Задача 12J. Редукция дерева [1 sec, 256 mb]

Задано неориентированное дерево, содержащее n вершин. Можно выбрать некоторое ребро и удалить его, при этом инцидентные ему вершины не удаляются. Таким образом можно удалить из дерева некоторый набор рёбер. В результате дерево распадается на некоторое количество меньших деревьев. Требуется, удалив наименьшее количество рёбер, получить в качестве хотя бы одной из компонент связности дерево, содержащее ровно p вершин.

Формат входных данных

Первая строка входного файла содержит пару натуральных чисел n и p ($1 \leqslant p \leqslant n \leqslant 1000$). Далее в $n-1$ строке содержатся описания рёбер дерева. Каждое описание состоит из пары натуральных чисел a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$) — номеров соединяемых ребром вершин.

Формат выходных данных

В первую строку выведите наименьшее количество рёбер q в искомом наборе. Во вторую строку выведите номера удаляемых рёбер. Номера рёбер определяются порядком их задания по входном файле. Рёбра нумеруются с единицы. Если оптимальных решений несколько, разрешается выводить любое.

Система оценки

Подзадача 1 (50 балла) $1 \leqslant p \leqslant n \leqslant 200$.
Подзадача 2 (50 балла) $1 \leqslant p \leqslant n \leqslant 1000$.

Пример

tree.in	tree.out
11 6 1 2 1 3 1 4 2 6 2 7 1 5 2 8 4 9 4 10 4 11	2 3 6

Задача 12K. Жить или не жить? [1 sec, 256 mb]

Гамильтонов путь — это путь в графе, проходящий по каждой вершине ровно один раз. Задача о его нахождении является NP-полной, а значит, современная наука не обладает возможностью эффективно решить её.

В этой задаче не нужно реализовывать алгоритм, который работает на всех возможных графах. Чтобы получить `Accepted`, достаточно верно найти *Гамильтонов путь* на имеющихся у жюри 99 тестах.

Формат входных данных

На первой строке входного файла записаны целые числа n и k — количества вершин и рёбер, соответственно. В следующих k строках записаны номера вершин, соединённых рёбрами. Петли и кратные рёбра отсутствуют, граф неориентирован. Число вершины в графе не больше 40. Думаете это мало? а теперь попробуйте сдать задачу :-)

Гарантируется, что в графе есть гамильтонов путь.

Формат выходных данных

Выведите перестановку из n чисел — номера вершин в порядке гамильтонова пути.

Пример

pathard.in	pathard.out
4 5 1 2 1 3 2 3 1 4 4 3	2 3 1 4

Замечание

Задача для тех, кто хочет попихать перебор с отсечениями =)

Задача 12L. Дорожные работы [1 sec, 256 mb]

В республике Икс издавна действует двухпартийная система. Каждый год граждане, имеющие избирательные права, голосуют, какой партии они больше доверяют — партии Мошенников или партии Грабителей, и в течение этого года вся реальная власть сосредоточена в руках избранной партии.

В последние M лет между партиями разразилась нешуточная война по перестройке дорожной сети республики «под себя». Партия Мошенников стремится построить как можно больше государственных дорог, чтобы прикарманить побольше бюджетных денег на их «обслуживание», а партия Грабителей стремится сделать платными как можно большее число дорог. Движение на всех дорогах республики Икс двустороннее.

Известно, что в течение одного года правления партии Мошенников удавалось построить ровно одну новую дорогу (которая поначалу является бесплатной), а партии Грабителей — ввести плату за проезд по одной из бесплатных на текущий момент дорог (при этом деньги на содержание этой дороги выделяются уже не из бюджета, а из средств, вырученных за проезд).

Президент республики, в настоящее время не имеющий реального политического влияния, решил привлечь внимание общественности к проблеме дорог. Он назвал дорожную сеть *удобной* (для простых граждан), если из любого города можно доехать до любого, используя только бесплатные дороги, но при этом количество бесплатных дорог (а, соответственно, и бюджетные средства на их содержание, полученные сбором налогов с граждан республики) — минимально возможное.

Вам поручено написать программу, которая определяет, была ли дорожная сеть удобной по завершении i-го года «дорожной войны».

Формат входных данных

В первой строке ввода заданы два числа — N ($1 \leqslant N \leqslant 1000$), число городов в республике Икс, и M ($1 \leqslant M \leqslant 100\,000$), продолжительность порядком затянувшейся «дорожной войны». Далее следуют M строк, первый символ каждой из которых — это `F`, если в данный год у власти была партия Мошенников, и `R` — если партия Грабителей, а далее в строке следуют два числа — номера городов u_i и v_i — пара городов, дорога между которыми стала объектом пристального внимания соответствующей партии (была построена новая дорога, если у власти была партия Мошенников, и одна из существующих дорог была сделана платной, если у власти была партия Грабителей). Вполне возможна ситуация, когда между двумя городами окажется более одной дороги, или будет построена дорога из города в себя — мало ли, что там удумает партия Мошенников.

Гарантируется, что входные данные корректны, то есть, все числа u_i и v_i лежат в пределах от 1 до N, и если известно, что в какой-то год дорога между двумя городами была сделана платной, то это значит, что перед началом года была хотя бы одна бесплатная дорога между этими городами.

Формат выходных данных

Для каждого года выведите в отдельной строке `YES`, если дорожная сеть по завершении соответствующего года была удобной, и `NO` в противном случае.

Пример

roadwork.in	roadwork.out
4 8 F 1 2 F 1 3 R 1 3 F 2 3 F 3 4 F 1 3 R 1 3 F 1 1	NO NO NO NO YES NO YES NO

Замечание

Допустимо решение за $\mathcal{O}(NM)$. Нужно его аккуратно написать.

Задача 12M. Потенциал [1 sec, 256 mb]

Дан взвешенный ориентированный граф. Пусть у каждой вершины есть потенциал Φ_i. Тогда к весу каждого ребра прибавляется потенциал начала и вычитается потенциал конца.

Требуется найти такие целые Φ_i, чтобы веса у всех рёбер были одинаковыми.

Формат входных данных

В первой строке задано целое число t — количество тестовых случаев.

В первой строке каждого тестового случая заданы целые числа n и m ($1 \leqslant n \leqslant 300\,000$, $0 \leqslant m \leqslant 300\,000$) — количество вершин и рёбер в графе. В следующих m строках задано по три целых числа x_i, y_i и w_i ($1 \leqslant x_i, y_i \leqslant n$, $-10^9 \leqslant w_i \leqslant 10^9$) — начало, конец и вес ребра. Гарантируется, что граф не содержит кратных рёбер и петель.

Также гарантируется, что сумма всех n и m по всем тестовым случаям не превосходит $600\,000$.

Формат выходных данных

Для каждого тестового случая выведите «YES», если существует целочисленное решение, и «NO» в противном случае.

Если ответ положительный, то в следующей строке выведите n целых чисел — потенциалы вершин. Все выведенные числа должны быть не больше 10^{18} по абсолютной величине. Гарантируется, что если ответ существует, то существует и ответ, удовлетворяющий этому ограничению.

Если возможных ответов несколько, выведите любой из них.

Пример

potential.in	potential.out
2 5 4 1 2 -1 2 3 2 3 4 1 4 5 179 5 5 1 2 1 2 3 1 3 4 1 4 5 0 5 1 2	YES 0 -1 1 2 181 YES 0 0 0 0 -1