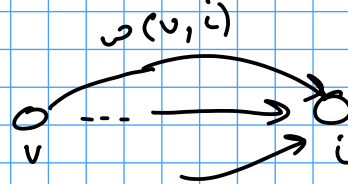
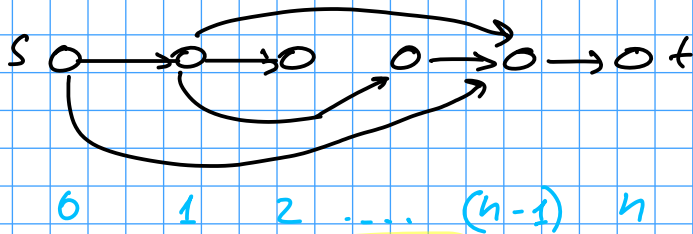


Динамическое программирование

Кратчайшие пути в ациклич. графах
 $O(V+E)$

$$d[i] = \text{dist}(s, i)$$



$$d[i] = \min_{(v, i) \in E} \{d[v] + w(v, i)\} \quad (*)$$

$d[n] = ?$

D - массив длины n.

В узле запомнить D значением $d[i]$.

$$O(V+E)$$

Шаги:

1. Выделили подзадачу ($d[i]$)
2. Вычислили соотношение (*)
3. Определить порядок вычисления подзадач.

Задача о самой длинной возрост. подпоследовательности

M

2	4	3	7	1	13	24	5
---	---	---	---	---	----	----	---

1. $L[i]$ - это самая длинная ПП кончить, закончив в ячейке i

$$2. L[i] = 1 + \max_{\substack{j < i \\ M[j] < M[i]}} \{L[j]\}$$

3. Порядок задач исходной последовательности.

$$L[0] = 0$$

for $i = 1$ to n

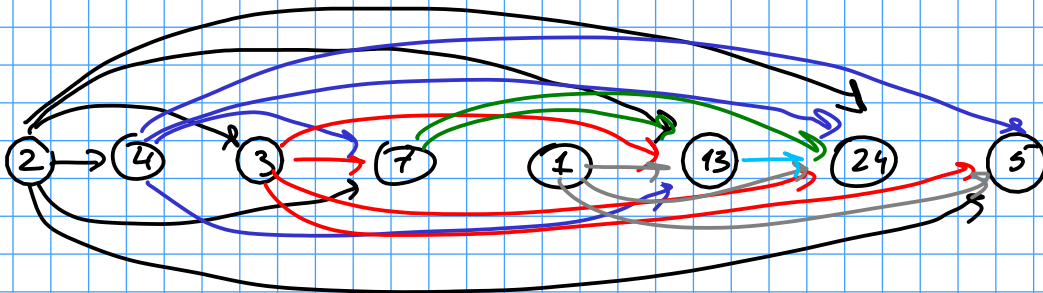
$$L[i] = \max_{\substack{j, M[j] < M[i]}} \{L[j]\} + 1$$

return $\max \{L[i]\}$

$$O(n^2)$$

$O(n \log n)$
(на практике)

Граф на подзадачах



Задача о рюкзаке

W - объём рюкзака

$$\{(w_i, v_i)\}_{i=1}^n \quad w_i - \text{вес}, \quad v_i - \text{стоимость}$$

1. "Непрерывный рюкзак"

Заполняем рюкзак товаром с наибольшей удельной стоимостью $\frac{v_i}{w_i}$ (жадный алгоритм)

2. Дискретный рюкзак с повторениями

$$W = 10$$

w	v
6	24
5	15

- Жадное решение

$$24$$

- Оптимальное решение

$$30$$

1. $V[k]$ макс. ценность

где рюкзак размера k

$$2. V[k] = \max_{1 \leq j \leq n} [V[k - w_j] + v_j]$$

3. Порядок: от 0 до W

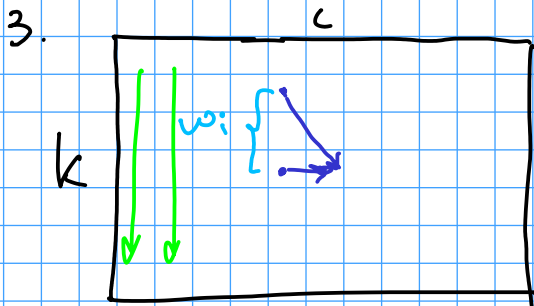
Время $O(W \cdot n)$
Память $O(W)$

NB: Алгоритм не оптимальный

3. Дискретный рюкзак без повторений

1. $V[k, i]$ - макс. ценность рюкзака размера k из первых i предметов.

2. $V[k, i] = \max \{ V[k, i-1], V[k-w_i, i-1] + v_i \}$



Время: $O(nW)$
Память: $O(nW)$

NB: если хранить только два столбца ($i-1$ и i) то память $O(W)$

Задача о перемножении матриц

$$(A_1 \times (A_2 \times A_3)) \times \dots \times A_n$$

$m_0 \times m_1 \times m_2 \dots m_n$

A_i размера $m_{i-1} \times m_i$

$$A \times B = C$$

$$A - n \times k$$

$$B - k \times m$$

$$C = n \times m$$

Пример:

$$A \times B \times C$$

$$(10 \times 20) \times (20 \times 40) \times (40 \times 100)$$

Умножение: $k \cdot n \cdot m$

$$(A \times B) \times C$$

$$8000 + 40000 = 48000$$

$$A \times (B \times C)$$

$$80000 + 20000 = 100000$$

1. $A[i, j] = \min$ cost-а операции умножения
 матриц $(A_i \times A_{i+1} \times \dots \times A_j)$

$A[1, n] = ?$

2. $A[i, j] = \min_{i \leq k < j} [A[i, k] + A[k+1, j] + m_{i-1} m_k m_j]$

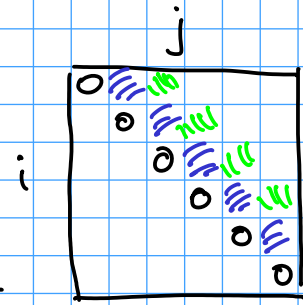
$A[i, i] = 0$

3. Порядок?

По длине отрезка:

Сначала $A[i, i+1]$

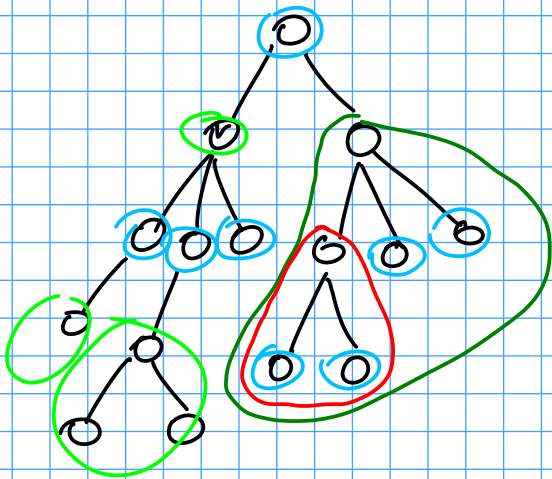
Потом $A[i, i+2] \dots$



Время: $O(n^3)$

Память: $O(n^2)$

Max независимое множество в дереве.



1. $IS[v]$ - макс. нез. мн-во
 в поддереве с корнем
 в v .

2. $IS[v] =$

$$\max \left\{ \sum_{w \text{ - сын } v} \underline{IS[w] + 1}, \sum_{w \text{ - сын } v} \underline{IS[w]} \right\}$$

3. Порядок: от листьев к
 корню.

NB: memoisation ("запоминание")

Иногда работает эрративнее. Нужна хеш-таблица

Пример: рюкзак с краешими весами