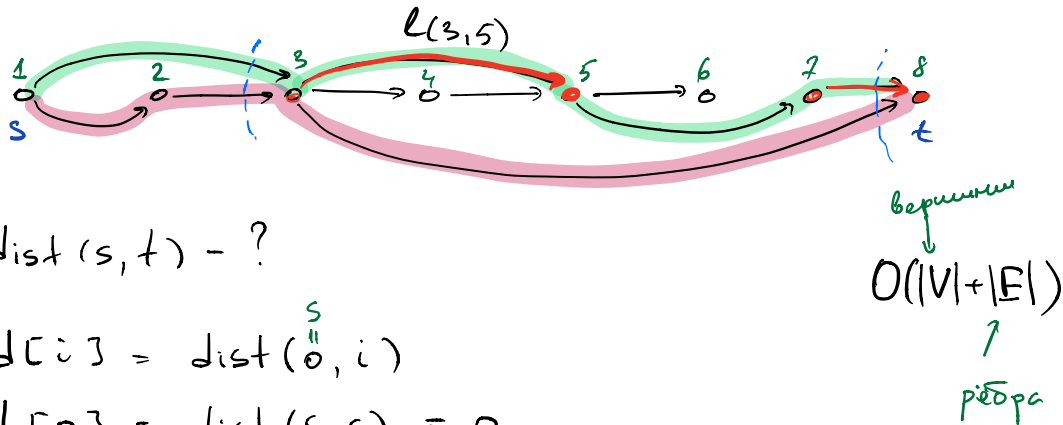


Динамическое программирование

Кратчайшие пути в ациклических графах



$\text{dist}(s, t) - ?$

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

База. $d[0] = \text{dist}(s, s) = 0$

Переход. $d[i] = \min_{(v,i) \in E} [d[v] + l(v,i)]$ $\text{prev}[i] = v$

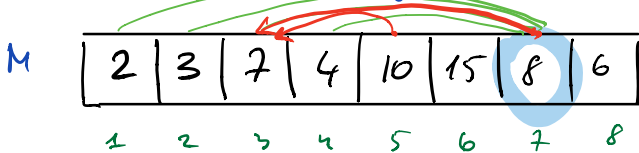
Порядок

Итоговое значение: $d[t] = \text{dist}(s, t)$

Чтобы решить задачу при помощи **динамического программирования**

1. Выделить подзадачи
2. Определить соотношение на подзадачах и базовое решение
3. Определить порядок вычисления подзадач

Задача о самой длинной возрастающей подпоследовательности



1. $L[i]$ - длина макс ↑↑ подпослед-ти, кот. занята в слове i

2. $L[1] = 1$ $O(n^2)$
 $L[i] = \max_{\substack{j < i \\ M[j] < M[i]}} \{ L[j] + 1 \}$ $prev[i] = j$ ↑ $prev$ max

3. Порядок естественный Можно за $O(n \log n)$

NB! соотношение задаёт граф на подпослед-тах

Подобно рекурсии с мемоизацией.

Задача о рюкзаке

Рюкзак размера W

В холодильнике есть предметы $1 \dots n$

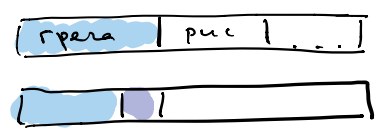
с весом w_i

с ценностью v_i

Набить рюкзак так, чтобы суммарная ценность

→ max

1. Непрерывный рюкзак
(жадный подход)



2. Дискретный рюкзак

- с повторениями (магазин техники)

$W = 10$ $w_1 = 6$ $v_1 = 24$

$$W_2 = 5 \quad V_2 = 15$$

Хорошее решение: $\{1, 5\} \quad V = 24$

Оптимальное решение: $\{2, 2\} \quad V = 30$

1. $V[k]$ - макс ценность рюкзака размера k
2. $V[0] = 0$
$$V[k] = \max_{\substack{1 \leq i \leq n \\ W_i \leq k}} \{V_i + V[k - W_i]\}$$
3. Порядок естественный

Время: $O(W \cdot n)$ Память: $O(W)$

Вопрос: А это точно полиномиальный алгоритм?

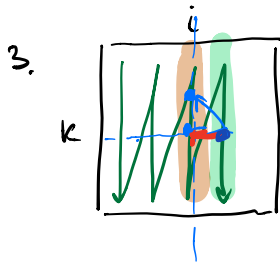
Если $|W| \approx n$ да, то итоговое время $O(2^n \cdot n)$

```
for k = 1 to W:
    for i = 1 to n:
        if V[k] < V[i - W_i] + V_i:
            V[k] = V[i - W_i] + V_i
            prev[k] = i
```

- Без повторов (карты, билеты)
(предметы в сумку, эл. компоненты)

1. $V[k, i]$ - решение для рюкзака размера k , если рассматриваются только предметы $\{1, \dots, i\}$
2. $V[0, i] = 0 \quad V[k, 0] = 0$

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



Время: $O(n \cdot W)$

Память: $O(n \cdot W)$

NB: Если нам нужны только опт. значение,
то можно хранить только предыдущий
столбец. \Rightarrow Память: $O(W)$

