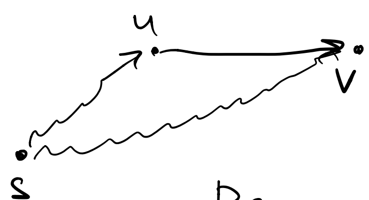
Кратчайшие пути в графах с отриц. рёбрами

Замечание: Если в графе есть цикл отрицательного веса, то кратчайшие расстояние не опр.

Операция релаксации



$d[u]$
$d[v]$

длины каких-то путей

Релаксация ребра (u,v)

$$d[v] = \min\{d[v],\ d[u] + \underbrace{w(u,v)}_{\text{вес ребра } (u,v)}\}$$

Утв $d[v] \geq dist(s,v)$

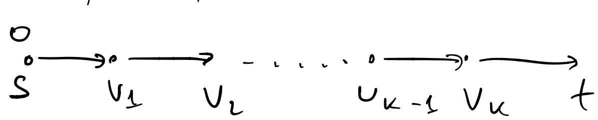
Релаксация может только улучшить оценку, но не испортить её

Утв Путь $s \to v_1 \to v_2 \to \dots \to v_k \to t$ — кратчайший путь $s \rightsquigarrow t$.

Давайте последовательно сделаем релаксацию всех рёбер пути: $(s,v_1), (v_1,v_2), \dots, (v_k,t)$

$\Rightarrow$ Вычислим расстояние до всех вершин $v_1, \dots, v_k, t$

relax(s, v_1)
relax(v_1, v_2)
⋮



Идея 1. Прорелаксируем все пути длиной $\leq |V|-1$ ребро.

$$\sim (|V|-1)! \sim |V|^{|V|}$$ много!

Идея 2. Давайте выберем посл-ть релаксаций рёбер такую, что $\forall$ посл-ть длины $\leq |V|-1$ будет её подпоследовательностью.

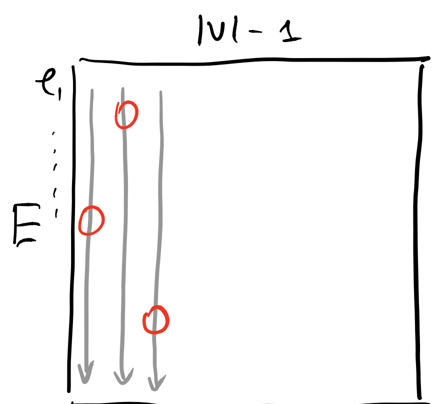
1 2 . 7 . 10 1 2 3 ... 10 1 2 . ... 10 1 2 . 8 10

5 3 1 7

7 4 2 8

7 3 1 5

7 3 2 8



Bellman-Ford((V, E), s):

```
for v ∈ V:
    d[v] = +∞
    prev[v] = 0
d[s] = 0
for i = 1 to |V| - 1:
    for (u, v) ∈ E:
        if d[v] > d[u] + w(u, v):
            d[v] = d[u] + w(u, v)
            prev[v] = u
```

$O(V \cdot E)$

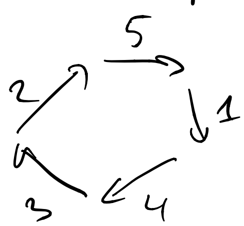
(*)

Утв Алгоритм Беллмана-Форда находит кратчайшие пути от s до всех вершин в графах без отриц. циклов.

(*)
```
for (u,v) ∈ E:
    if d[v] > d[u] + w(u,v):
        throw Exception("Negative cycle")
```

Утв: Если бросил исключение $\Rightarrow$ $\exists$ отриц. цикл.

Утв: Есть отриц. цикл $\Rightarrow$ исключение (*)



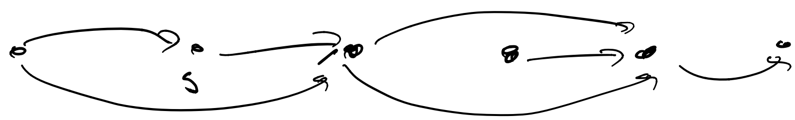
Замечание: Если на итерации i ни для одной вершины расстояние не улучшилось, то алг. можно остановить

Замечание: Если в графе есть отриц. цикл, то мы можем всё улучшать расстояние до некоторых вершин.

$\Rightarrow$ На $\forall$ итерации есть улучшение $\Rightarrow$ (*)

Кратчайшие пути в ациклических графах

За $O(V+E)$



Кратчайшие пути м/у всеми парами вершин

Алгоритм Флойда-Уоршалла

```
Floyd-Warshall(V, E):
    for i in V:
        for j in V:
            d[i,j] = +∞ , prev[i,j] = 0
            if (i,j) ∈ E:
                d[i,j] = w(i,j)
                prev[i,j] = i
            if i == j:
                d[i,j] = 0
    for k in V:
        for i in V:
            for j in V:
                if d[i,j] > d[i,k] + d[k,j]:
                    d[i,j] = d[i,k] + d[k,j]
                    prev[i,j] = prev[k,j]
```

Время $O(V^3)$

Память $O(V^2)$

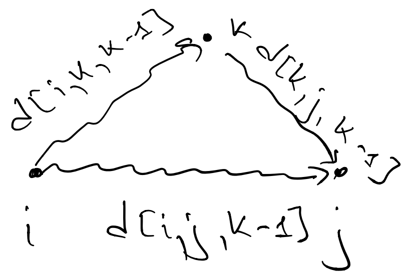
предпосл. вершина на пути $i \rightsquigarrow j$

Динамическое программирование

1. $d[i,j,k] = dist(i,j)$, если промеж. вершины из мн-ва $\{1, \dots, k\}$

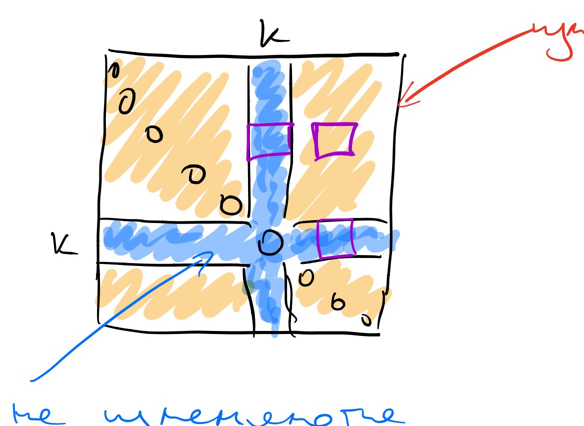
2. $$d[i,j,0] = \begin{cases} w(i,j) & (i,j) \in E \\ +\infty & \text{otherwise} \end{cases}$$

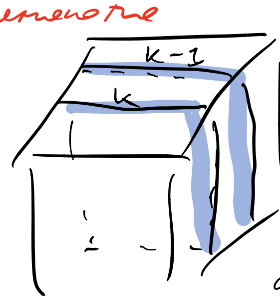
$$d[i,j,k] = \min\{ d[i,j,k-1],\ d[i,k,k-1] + d[k,j,k-1] \}$$

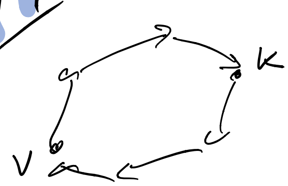


3. for k { for i { for j .. } } }

Утв Достаточно двумерного массива







Утв Алг. Ф-У. работает с отриц. весами

Утв Отриц. циклы $\Leftrightarrow$ отриц. числа на диаг.