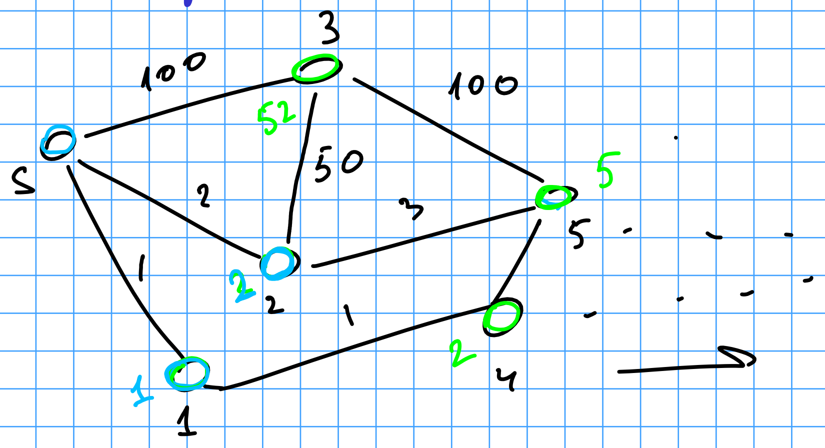
Алгоритм Дейкстры



Очередь с приоритетом

1. Make_priority_queue
2. Delete_min
3. Insert (v, t)
4. Decrease-Key (v, t)

Реализации:

- куча $O(\log n)$ для 2-4, $O(n)$ для 1
- массив $O(n)$ для 1-2, $O(1)$ для 3-4

```
Dijkstra (S, G):
    for v ∈ V:
        dist[v] = ∞
        prev[v] = 0
    dist[S] = 0
    P = make-priority-queue((S, 0))
    while P.size() > 0:
        (v, d) = P.delete_min()
        for (v, u) ∈ E:
            if dist[u] = ∞:
                P.insert((u, d + w(v,u)))
                dist[u] = d + w(v,u)
```

```
        prev[u] = v
    else if d + w(v,u) < dist[u]:
        P.decrease_key(u, d+w(v,u))
        dist[u] = d + w(v,u)
        prev[u] = v
```

Утв: (Корректность алгоритма)

В процессе алгоритма V разделяется на три подмножества

S – те вершины, кот. мы уже обраб.

R – вершины в очереди

T – вершины с $dist = \infty$

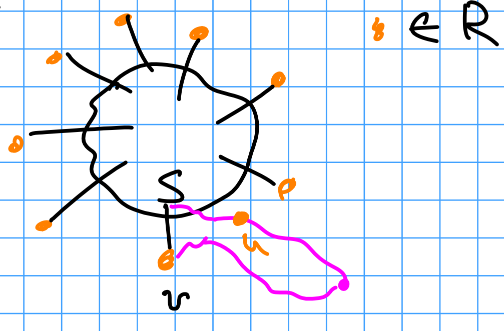
$$V = S \sqcup R \sqcup T$$

Если v переходит из R в $S \Rightarrow$ $dist[v] = dist(s,v)$.

▷ 1. База $S = \{s\}$

2. Пусть для всех вершин в S расстояние уже посчитано.

3. На след. шаге мы обрабатываем $v, d \leftarrow P.delete\_min()$.



? $d = dist(s,v)$?

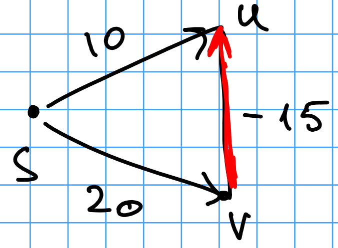
↯] $d > dist(s,v)$

$u \in R$ (первая вершина на кратчайшем пути $\notin S$)

$dist(s,v) \geq dist[u] \geq dist[v]$ ◁

В нашем случае отрицательные веса

Всё сломают?



$S = \{s\} \to \{s, u\} \to \{s, u, v\}$

$\text{dist} \quad 0 \qquad 0 \quad 10 \qquad 0 \quad 10 \quad 20$

5

Оценка сложности

Сложность = |Make_priority_queue| +

V · |Delete_min| +

E · max(|Decrease_key|, |Insert|)

В случае кучи:

$$O(V) + O(V \cdot \log V) + O(E \cdot \log V) =$$
$$= O((V+E) \cdot \log V)$$

В случае массива:

$$O(V) + O(V \cdot V) + O(E \cdot 1) = O(V^2)$$

В разреженных графах — куча

В плотных графах — массив

В d-ичной куче

$$O(V) + O(d \cdot \log_d V \cdot V) + O(\log_d V \cdot E) =$$
$$= O(V \cdot d \cdot \log_d V + E \cdot \log_d V)$$

$$d \sim \frac{E}{V} \qquad E \sim V \Rightarrow d = 2$$
$$E \sim V^2 \Rightarrow d \sim V$$

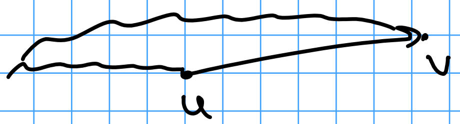
NB: Дейкстра не работает с отриц. весами.

Кратчайшие пути с отриц. весами.

Утв: В неор. графе с отриц. весами нет понятия кратчайшего пути.

Утв: Если в ор. графе есть цикл отриц. веса ⇒ понятие кратч. пути не опр.

Операция релаксации ребра.



$$dist[v] = \min(dist[v],\ dist[u] + w(v,u))$$

Утв: Релаксация „безопасна“

$dist[v] \geq dist(s,v)$

после релаксации

Утв:

$\exists\ \pi = s, v_1, v_2, \dots, v_{k-1}, v$ – кратч. путь от s до v

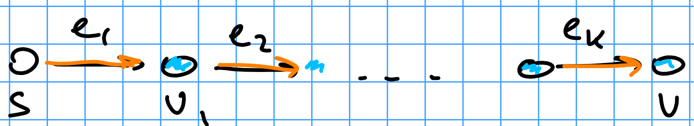
$|\pi| \leq V$

\# кол-во рёбер в $\pi \leq V - 1$

$e_1, \dots, e_k$ – рёбра π

Давайте проведём релаксации $e_1, e_2, \dots, e_k$.

$\Rightarrow dist[v] = dist(s,v)$



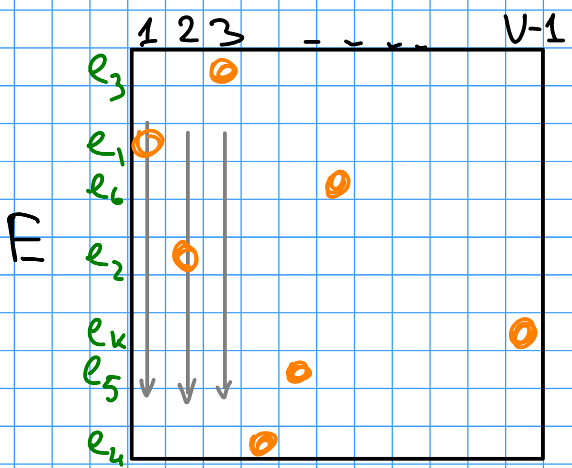
Проблема: мы не знаем порядка $e_1 \dots e_k$

Идея: Давайте прорелаксируем все рёбра $V - 1$ раз.

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

$O(V \cdot E)$

Утв: Корректность алгоритма



Необходимая последовательность релаксаций – подпоследоват. всех релаксаций.

Применимость: отсутствие отриц. циклов

Утв: Если мы сделаем V итерации и на последней итерации dist изменится $\Rightarrow$ в графе есть цикл отриц. веса

Утв: Есть отриц. цикл $\Rightarrow$ на V-ой итерации dist изменится.