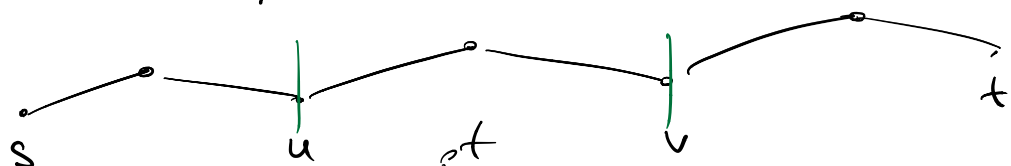
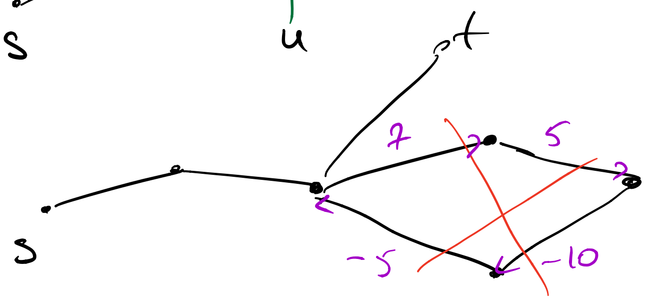
Кратчайшие пути

Утв Простой путь – путь без циклов.

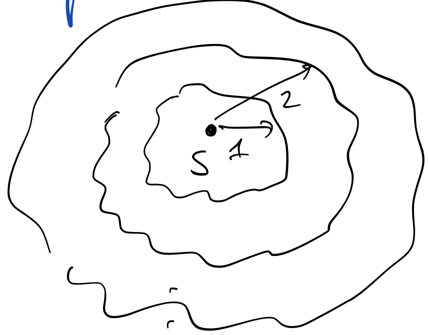
Утв ∀ сегмент кратчайшего пути – тоже кратчайший





Утв ∀ кратчайший путь – простой

Кратчайшие пути в невзвешенных графах



Поиск в ширину
Breadth-width search

```
def BFS((V,E), s):
    for v in V:
        dist[v] = +∞
        prev[v] = 0
    dist[s] = 0
```

Кратчайшие пути от s до всех вершин

```
Q = Queue()
Q.enqueue(s)
while Q.size() >= 1:
    v = Q.dequeue()
    for (v, u) ∈ E:
        if dist[u] = +∞:
            dist[u] = dist[v] + 1
            prev[u] = v
            Q.enqueue(u)
```

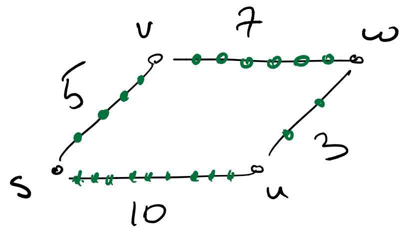
Утв (корректность)

Вершины обрабатываются в порядке ↑ $dist(s, v)$

Сложность: $O(V+E)$ для списков смежности

$O(V^2)$ для матрицы смежности

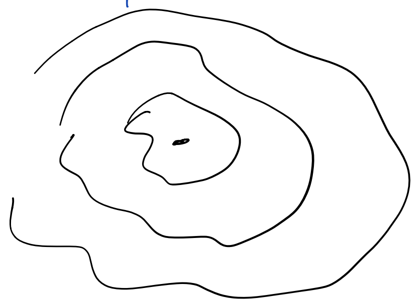
Кратчайшие пути во взвешенных графах

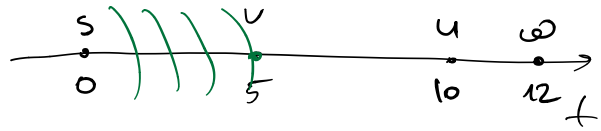


фиктивные вершины

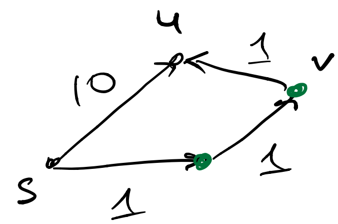
Минусы:
- большие веса
- не целые веса
- отрицательные веса

Алгоритм Дейкстры (для неотр. весов)






```
def Dijkstra((V,E), s):
    for v ∈ V:
        dist[v] = +∞
        prev[v] = 0
    dist[s] = 0
    P = PriorityQueue()
    P.insert(0, s)
    while P.size() ≥ 1:
①       v = P.extract_min()
        for (v,u) ∈ E:
            if dist[u] == +∞:
                dist[u] = dist[v] + ω(v,u)
                prev[u] = v
②               P.insert(dist[u], u)
            else if dist[u] > dist[v] + ω(v,u):
                dist[u] = dist[v] + ω(v,u)
                prev[u] = v
③               P.decrease_key(dist[u], u)
```



Утв Алгоритм Д. корректен

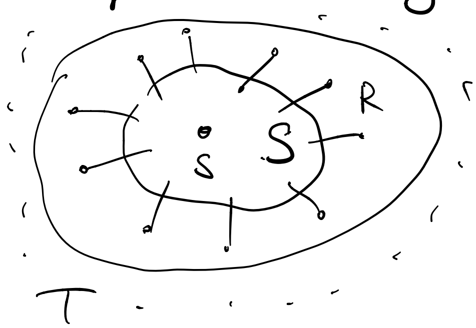
Алгоритм разбивает V на три части:

S – обработанные вершины

R – вершины в очереди

T – вершины с $dist = +\infty$

Вершины путешествуют $T \to R \to S$



Если v переходит из R в S $\Rightarrow$

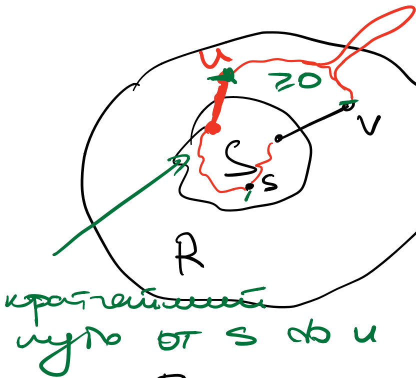
$dist[v] = dist(s, v)$

По индукции:

1. База: $S = \{s\}$
   $dist[s] = dist(s, s) = 0$
2. Предположение: $\exists$ для всех вершин в S это верно (т.е. расстояние корректно)
3. Пусть на этом шаге мы $\measuredangle$ вершину v с min dist из вершин в очереди

   $S \leftarrow S \cup \{v\}$

Покажем, что $dist[v] = dist(s, v)$



От обратного:

$\exists\ dist[v] > dist(s,v)$

$dist[u] \geq dist[v]$

Длина пути ч/з вершину u не может быть < dist[v].

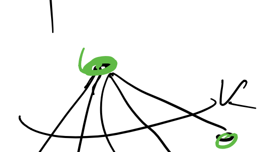
$\square$

Сложность:

$|V| \times$ extract_min
$|V| \times$ insert
$|E| \times$ decrease_key

Реализации очереди с приоритетом

	Массив	Двоичн. куча	k-ичная куча
extract_min	$O(n)$	$O(\log n)$	$O(k \log_k n)$
insert	$O(1)$	$O(\log n)$	$O(\log_k n)$
decrease_key	$O(1)$	$O(\log n)$	$O(\log_k n)$



1. На массиве

$$O(V^2 + V + E) = O(V^2)$$

2. На двоичной куче

$$O((V+E)\log n)$$

3. На k-ичной куче

$$O(V\cdot k\log_k n + E\cdot\log_k n)$$

Выбор реализации в зависимости от плотности графа

1. $E \sim V^2 \Rightarrow O(V^2)$ на массиве
2. $E \sim V \Rightarrow O(E\log V)$ на двоичной куче
3. $E \sim V^{1.5}$

$k = \sqrt{V}$

на k-ичной куче:

$$O(V\cdot\sqrt{V}\cdot\log_{\sqrt{V}} V + V^{1.5}\cdot\log_{\sqrt{V}} V) =$$
$$O(V^{1.5})$$

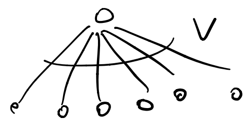
В произвольном случае:

$$k \sim \frac{E}{V}$$

$$E = V^{1+\varepsilon} \Rightarrow O(V^{1+\varepsilon})$$

$$E \sim V \Rightarrow k = 2$$

$$E = V^2 \Rightarrow k = V$$



Отрицательные рёбра