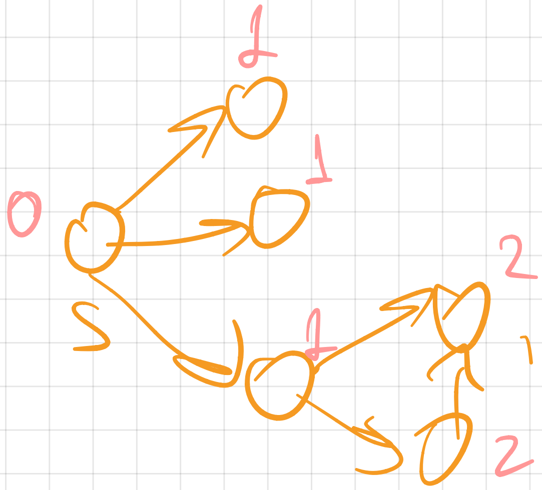
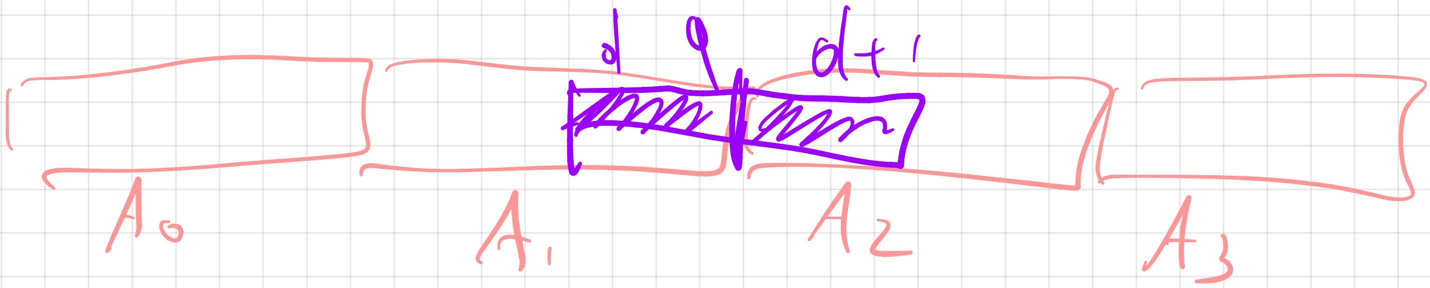
BFS

```
q = Queue()
q ← {s}
dist[s] = 0

while v ← q:
    for u in adj[v]:
        if dist[u] is None:
            dist[u] = dist[v] + 1
            q ← u.
```





Утв: в q живут в-ны не более двух расст. и более малые расст. раньше более поздних.

Также, если в-на живёт в очереди у

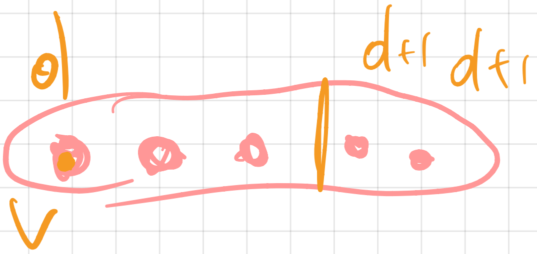
Ещё верно посчитано расст.

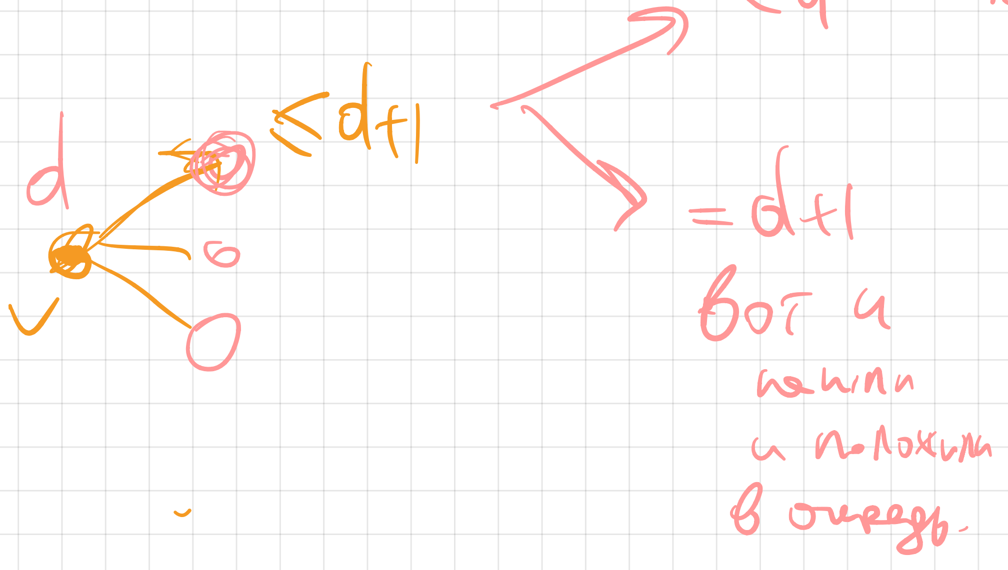
Если мин расст. у в-н в очереди d, то все в-ны расст $\leq d$ найден ответ верно

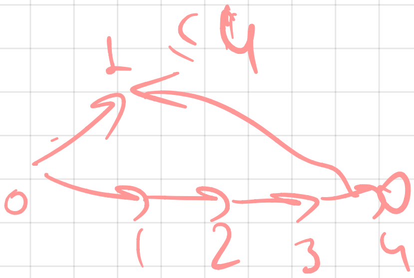
Д-в по индукции

База в-на S $d=0$

Переход





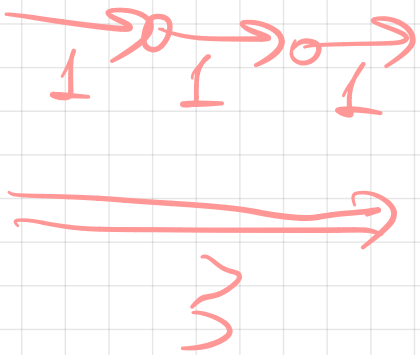


$O(V+E)$

BFS 1-k

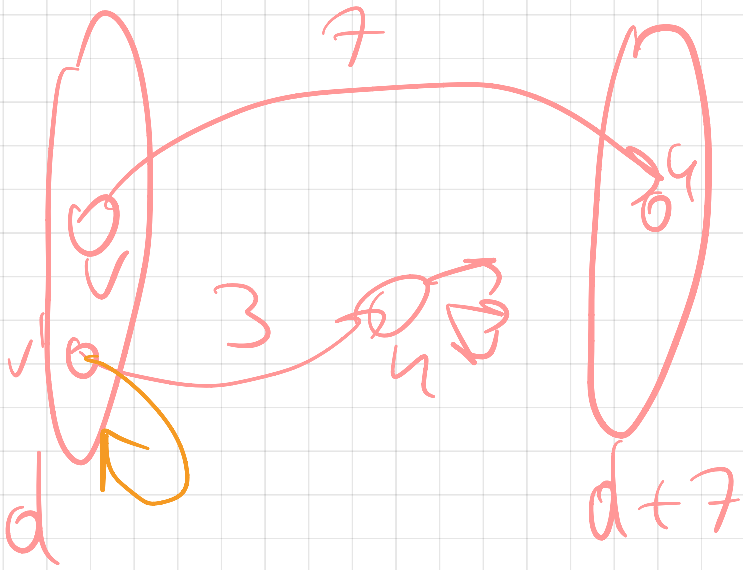
все рёбра имеют вес $\in [1, 10]$

Можно размножить



$O(V+E)$

BFS со слоями



$A = [[\,]\ ----\ ]$ ← kV
↑ макс. длина ребра

$A[0] = S$

$dist[S] = 0$

$\cancel{O(kV + kE)}$

$O(kV + E)$

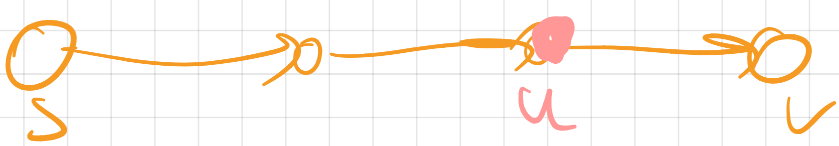
```
for d = 0..n-1
    for v ∈ A[d]:
        if dist[v] == d:
            for (u, w) ∈ adj[v]
                if dist[u] > d + w:
                    A[d+w].add(u)
                    dist[u] = d+w
```

<u>Утв</u> Нельзя найти расст. меньше настоящего.

<u>Утв:</u> расст. найдутся верно.

<u>Д-во</u> Пусть не найдутся в-но

Выберем v: у v посчитано неверно и $dist(s, v) \to \min$



Асимптотика: $O(kV + kE)$ или $O(kV + E)$

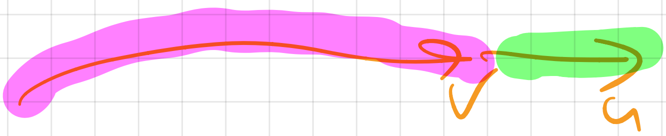
Bfs: $[0; k]$.

0-1 bfs: $[0; 1]$

bfs: восстановление пути.


```
q = Queue()
q ← {s}
dist[s] = 0

while v ← q:
    for u in adj[v]:
        if dist[u] is None:
            dist[u] = dist[v]+1
            par[u] = v
            q ← u.
```



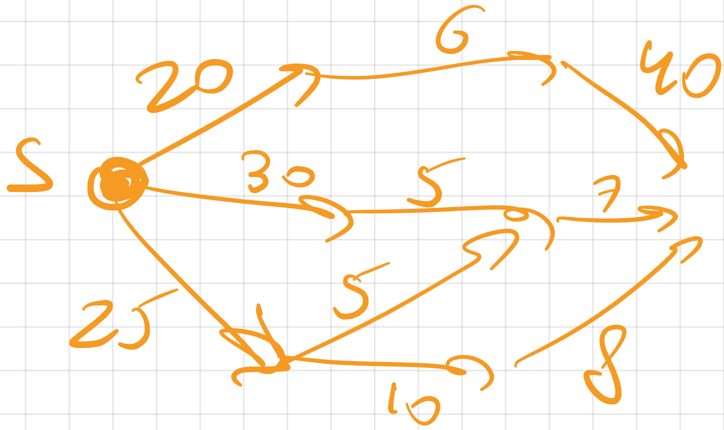
SSSP (single source)

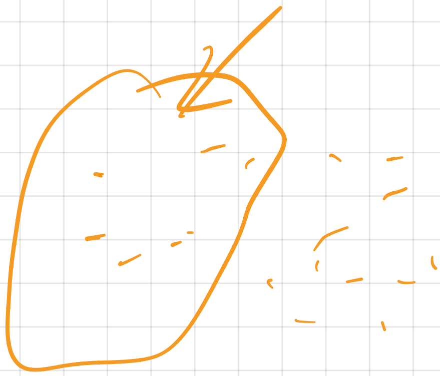
$w=1$	bfs
$w\in[0,k]$	0-k BFS
$w\geq 0$	dijkstra.
$w\in\mathbb{R}$	?

APSP (all pairs) ? ?

Dijkstra (1956)

помеченные





$S\subseteq V$

d_v ← кандидат на расст. $dist(s,v)$

соотв. какому-то пути, по возм. есть путь лучше.

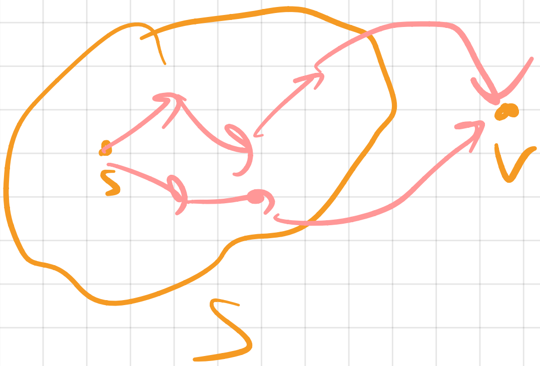
Инварианты:

0) $s \in S$

1) $v \in S \Rightarrow d_v = dist(s, v)$

(расст. найдено в-но)

2) $v \notin S \Rightarrow d_v =$ кр. путь от s до v среди путей делающих ровно один шаг снаружи S.



Алгоритм.

1) $d = [\infty; \infty; \ldots \infty]$

иниц. 2) $S = \{s\}$, $d_s = 0$

3) for $(u, w) \in adj[s]$:

$d_u = w$

а) while $S \neq V(G)$:

Пусть v: $v \notin S$ и

$d_v \to \min$.

Тогда $\boxed{S += \{v\}}$

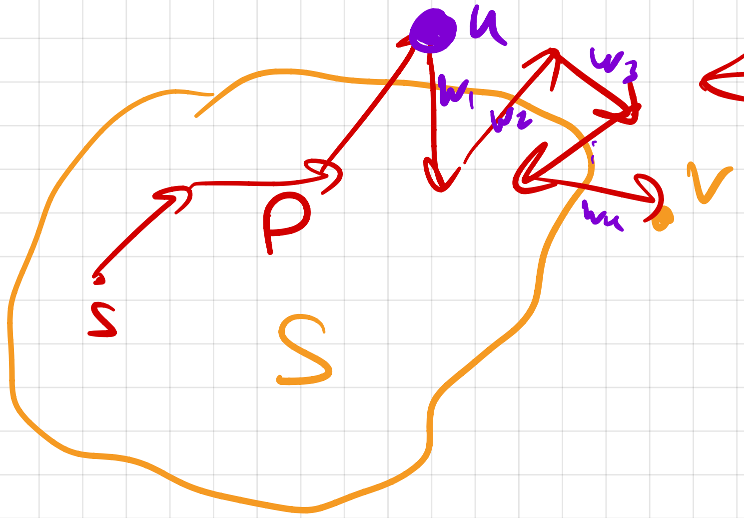
(Утв: d_v посчитано верно)

Выполнили инв. (2)

```
for (u, w) in adj[v]:
    if u ∉ S:
        d_u = min(d_u, d_v + w)
```

<u>Утв:</u> d_v посчитано верно.

Пусть неправда (u это самый ранний момент в алгоритме когда стало неправда)



как будто бы применили утв

$$d_u \geq d_v$$

$$w_1 \ldots w_k \geq 0$$

$$w(P) \geq d_u + \underbrace{w_1 + \ldots + w_k}_{\geq 0}$$

$$\geq d_u$$

$$\geq d_v$$

Реализация Дейкстры.

1) $d = [\infty; \ldots; \infty]$

2) used = [false; ....; false]

($v \in S \Leftrightarrow$ used[v])

3) d[start] = 0

4) loop (n раз):

(*) выбрать v: !used[v] && d[v] → min (d(v))

(if d[v] == ∞: break)

used[v] = 1

$O(E)$ в сумме

for (u, w) in adj[v]:
    if !used[u]:
        (**) d[u] = min(d[u], d[v] + w)

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

Dijkstra + Heap.

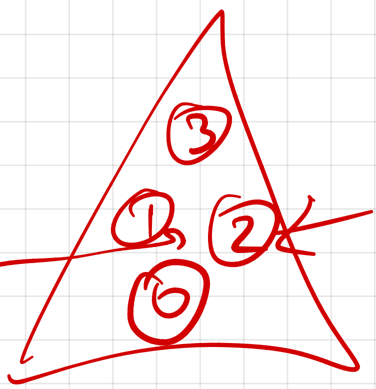
Пусть в куче хранятся в-ны с ключом d[v]

$O(\log V)$

(*) ← извлечь min в куче.

(**)

$O(\log V)$

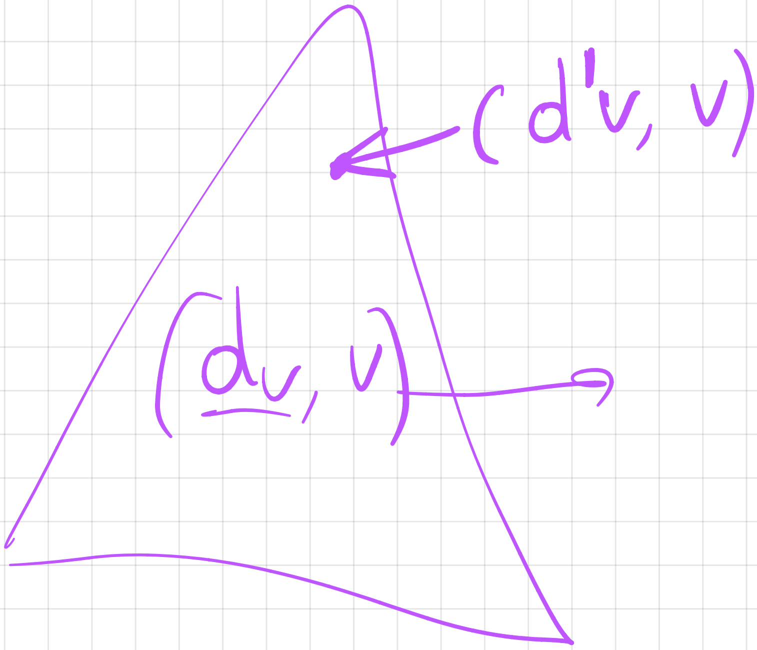


index[2]

ключ -- DecreaseKey (Sift Up)

$O(V\log V + E\log V)$ - бинарная куч

$O(V\log V + E)$ - фиб. куча.



while H не пуста:

(d, v) ← достать из кучи

if $d_v \neq d$:

continue

Bellman-ford

SSSP, w не обяз. ≥ 0.

$dp_{v,d}$ = кр-й путь $s \to v$

V^2 сост. из ровно d рёбер.

$d \in [0; n-1]$

База.

$dp_{s,0} = 0$

$dp_{*,0} = \infty$

v, d

Переходы

$V E$ переход.

$dp_{v,d} \leftarrow dp_{u,d-1} + w$

$(u, v, w) \in E$

$E \geq V$.

$O(VE)$

$dp_{*,*} = \infty$
$dp_{s,0} = 0$

Д.П.

for $d = 0..n-2$:
  for $(u,v,w) \in E$:
    $dp_{v,d+1} = \min(dp_{v,d+1},\ dp_{u,d} + w)$

$dp_{q,d} =$ $= \min$ по путям длины d

Б.Ф.

$dp_{*} = \infty$
$dp_{s} = 0$

for $d = 0..n-2$:
  for $(u,v,w) \in E$:
    $dp_{v} = \min(dp_{v},\ dp_{u} + w)$

$dp_{q,d} =$ $= \min$ по путям длины d

и ещё чем-то