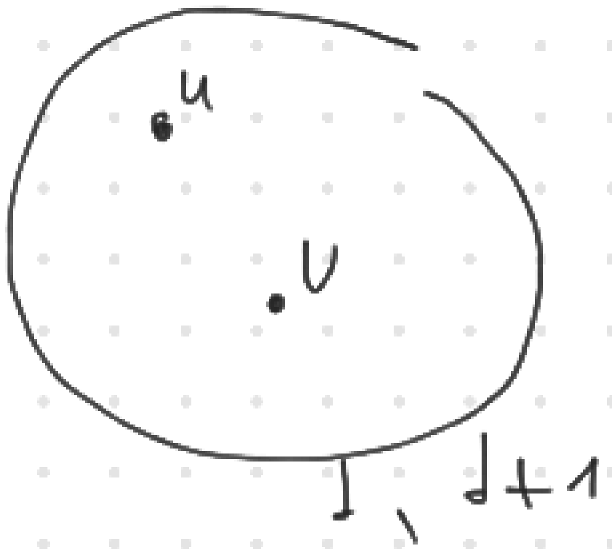
Кратчайшие пути, часть 1

DFS стэк / рекурсия

Времени входа / выхода

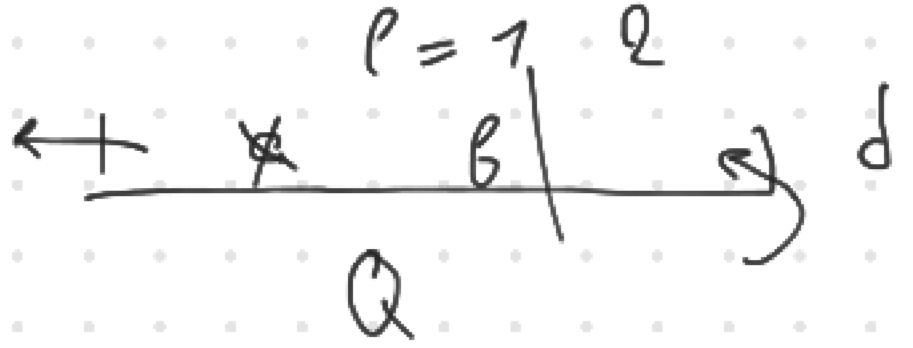
previsit / postvisit

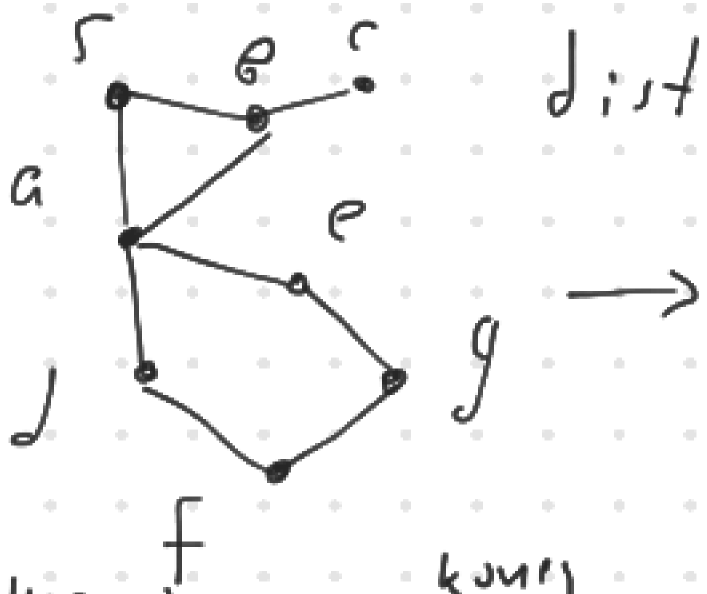
connected comp.



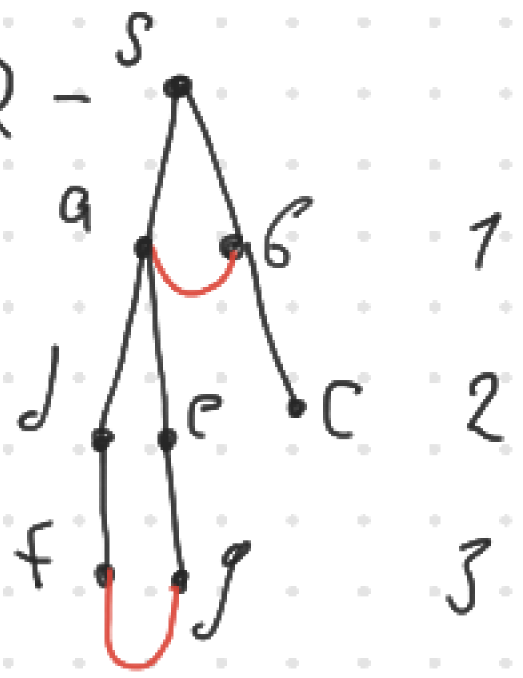
dist(u,v) — # рёбер в кр. пути $u \to v$

$d \to d+1$





dist(s,x) Q –



начало

конец

	Q	
0	~~s~~	
1	~~a~~ б	1
2	~~б~~ de	1, 2
3	dec	2
4	ecf	
5	cfg	
6	fg	

Утв. BFS (breadth-first search) корректен — т.е. все расст. верны

Док. для всех вершин с расст $\le l$ алг. корр.

$l \to l+1$

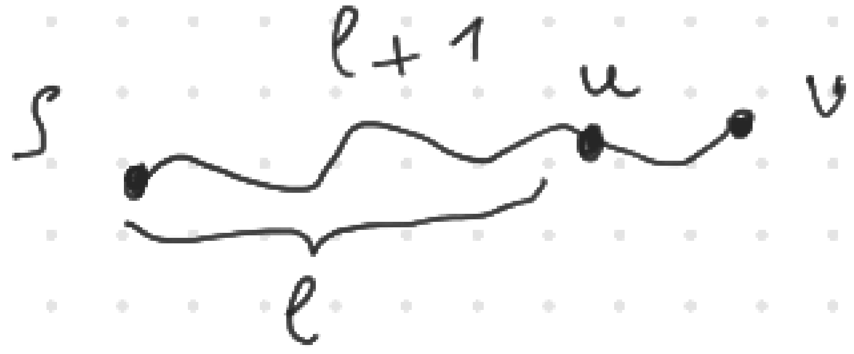


Figure 4.3 Breadth-first search.

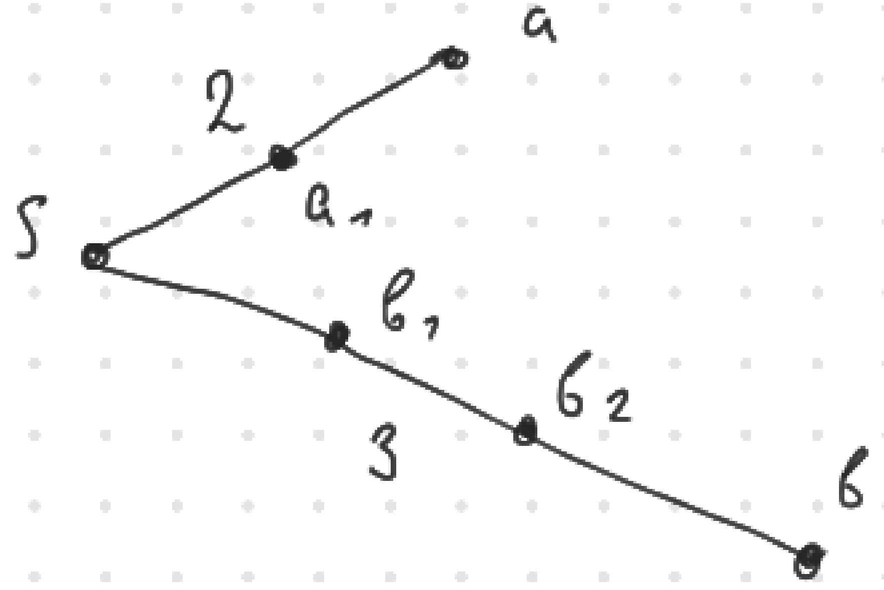
```
procedure bfs(G, s)
Input:    Graph G = (V, E), directed or undirected; vertex s ∈ V
Output:   For all vertices u reachable from s, dist(u) is set
          to the distance from s to u.

for all u ∈ V:
   dist(u) = ∞

dist(s) = 0
Q = [s] (queue containing just s)
while Q is not empty:
   u = eject(Q)
   for all edges (u, v) ∈ E:
      if dist(v) = ∞:
         inject(Q, v)
         dist(v) = dist(u) + 1
```

$\underline{O}(n+m)$

Q

u

соседи u

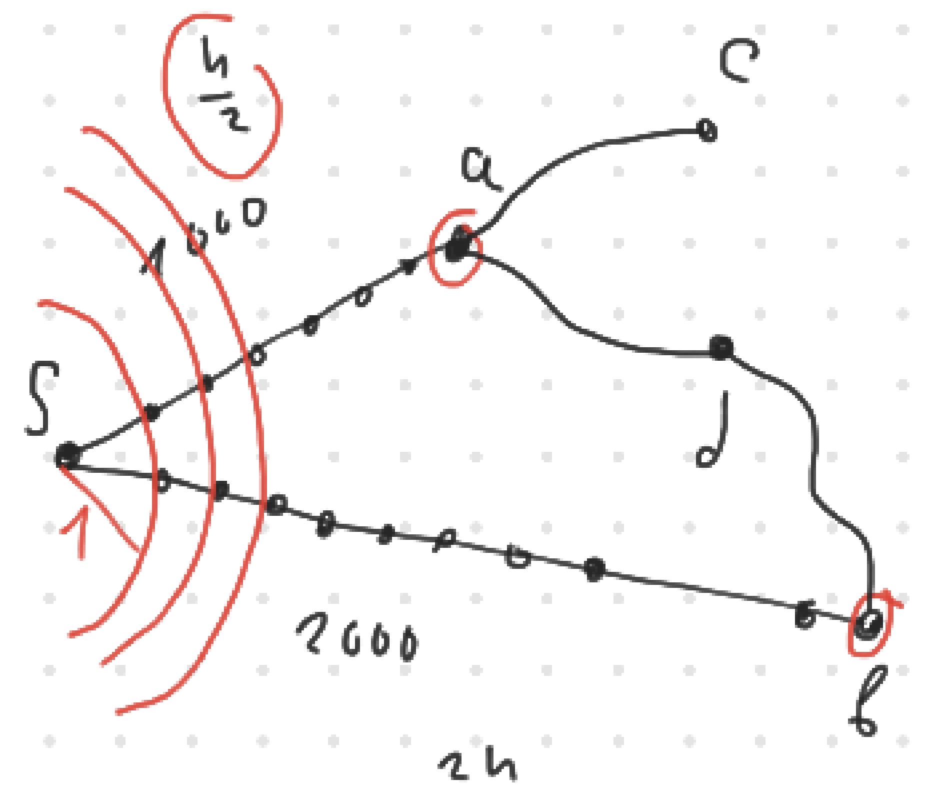


$l : E \to \mathbb{N}$

$l(e)$ – длина

$l(e) \le k$ – const

$O(n+m)$



„нашли s в $t=0$"

события „нашли вершину x"

„нашли s в момент $t=0$"

„нашли a в момент l_{sa}"

„нашли b в момент l_{sb}"

1) задать $dist(s)=0$
$dist(x \ne s) = \infty$

2) пока есть события:

(u,t) – ближ. соб.

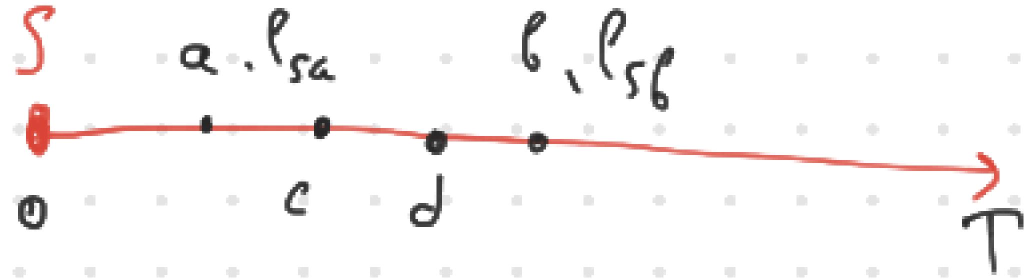
для $v \in$ соседи u:

если v не встр.

добав $(v, —)$

если v встреч:

обновить расст.



$dist(s,a) + l_{ac} < l_{sb}$

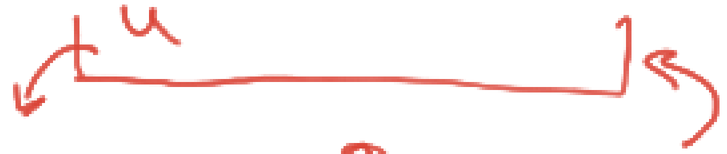
$\Rightarrow$ c раньше b

priority queue
delete_min()
insert(v,t)
decrease_key(v,t)
(v, ∞)

Figure 4.8 Dijkstra's shortest-path algorithm.

```
procedure dijkstra(G, l, s)
Input:    Graph G = (V, E), directed or undirected;
          positive edge lengths {l_e : e ∈ E}; vertex s ∈ V
Output:   For all vertices u reachable from s, dist(u) is set
          to the distance from s to u.

for all u ∈ V:
   dist(u) = ∞
   prev(u) = nil
dist(s) = 0

H = makequeue(V)   (using dist-values as keys)
while H is not empty:
   u = deletemin(H)
   for all edges (u, v) ∈ E:
      if dist(v) > dist(u) + l(u, v):
         dist(v) = dist(u) + l(u, v)
         prev(v) = u
         decreasekey(H, v)
```

BFS



Q

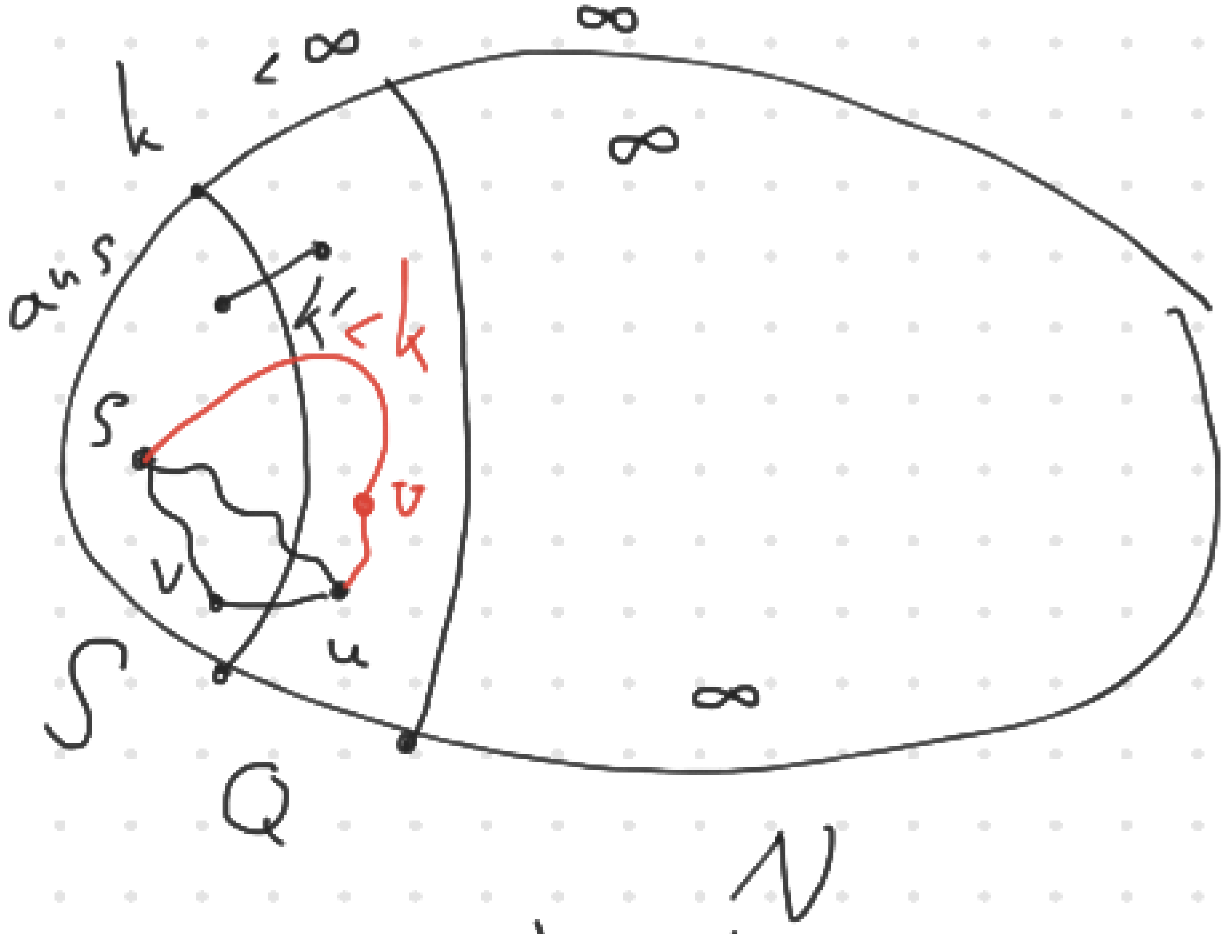
$l_{uv} \in \mathbb{R}_{\geq 0}$

нашли и фикс dist (s,u)

V (S = {удалённые из pr. q. вершина}
Q = {верш в pr. q.}
N = {верш, кот не были в pr. q.}

У76 алг. Дейкстра корректен

k — известная область
(k=0) время события в обработке

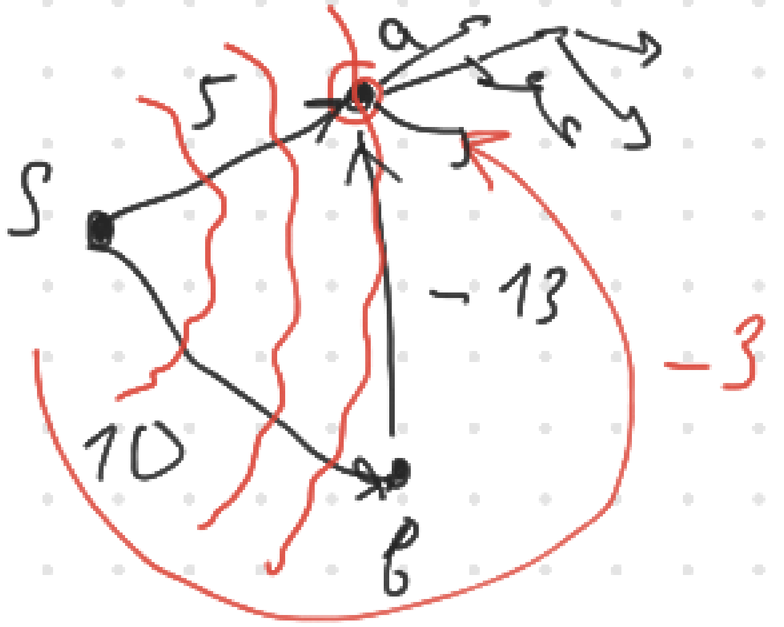


Переход: для всех k'<k верно

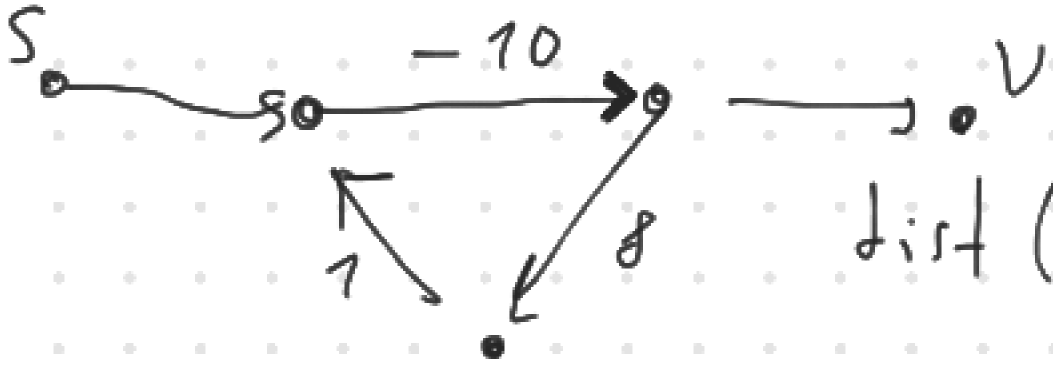
Q —› u dist(s,u) = k

1) v ∉ S dist(s,v) ≥ k

2) v ∈ S ⇒ dist(s,u) < k ?!



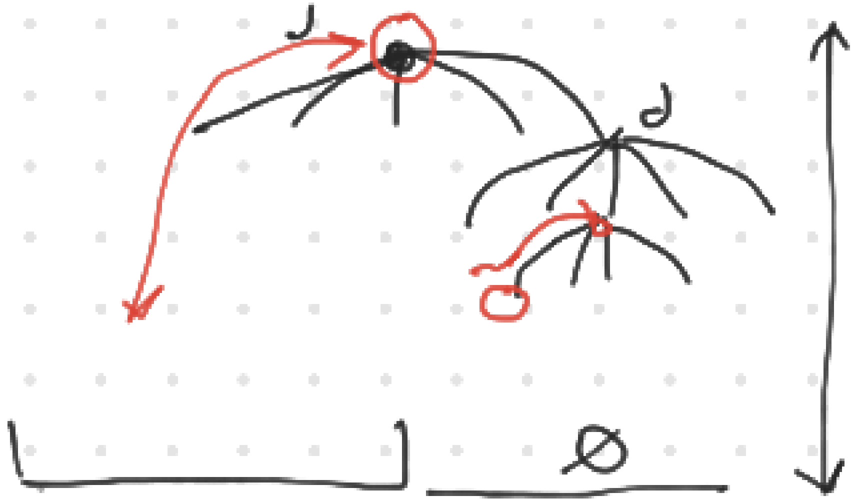
min



$dist(s,v) = -\infty$

0-k BFS

Implementation	deletemin	insert/ decreasekey	$\lvert V\rvert$ × deletemin + ($\lvert V\rvert + \lvert E\rvert$) × insert
Array	$O(\lvert V\rvert)$	$O(1)$	$O(\lvert V\rvert^2)$
Binary heap	$O(\log \lvert V\rvert)$	$O(\log \lvert V\rvert)$	$O((\lvert V\rvert + \lvert E\rvert)\log \lvert V\rvert)$
d-ary heap	$O(\frac{d \log \lvert V\rvert}{\log d})$	$O(\frac{\log \lvert V\rvert}{\log d})$	$O((\lvert V\rvert \cdot d + \lvert E\rvert)\frac{\log \lvert V\rvert}{\log d})$
Fibonacci heap	$O(\log \lvert V\rvert)$	$O(1)$ (amortized)	$O(\lvert V\rvert \log \lvert V\rvert + \lvert E\rvert)$



2 d

$$\frac{E \log \log V}{\sqrt{\log V}}$$

h

$$\log_d h = \frac{\log_2 h}{\log_2 d}$$

$$2^{\log_d h \cdot \log_2 d} = 2^{\log_2 h} = h$$