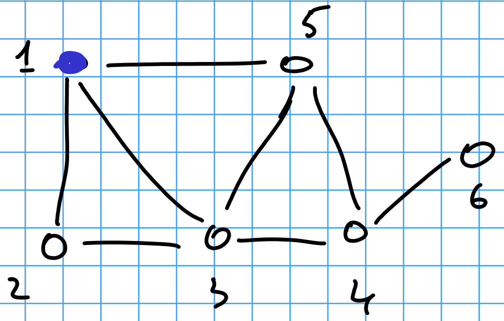
Кратчайшие пути в графах

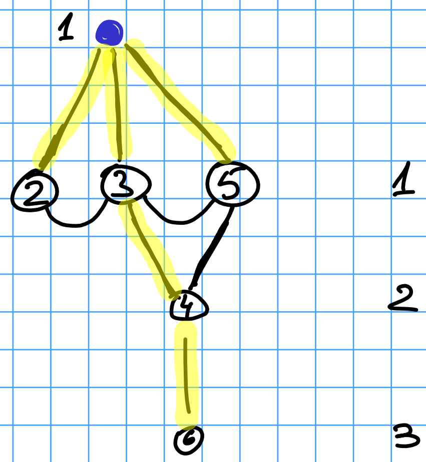
≡ Путь в графе — это посл-ть вершин $v_1 \dots v_k$: $(v_i, v_{i+1}) \in E$

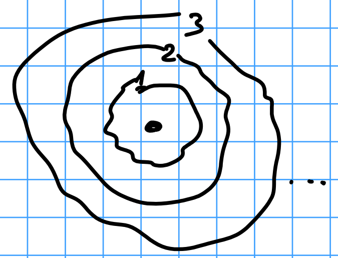
≡ Кратчайший путь — это мин по # верш. путь м/у 2-мя вершинами u и v.

≡ Простой путь — все вершины различные

Ут-в: $\forall$ кратчайший путь — простой путь.







Поиск в ширину (Breadth-first search)

```
BFS(v):
    for u ∈ V:
        dist[u] = ∞
        prev[u] = 0
dist[v] = 0
Q = {v}
while not Q.empty():
    u = Q.dequeue()
    for (u, w) ∈ E:
```

$O(V + E)$

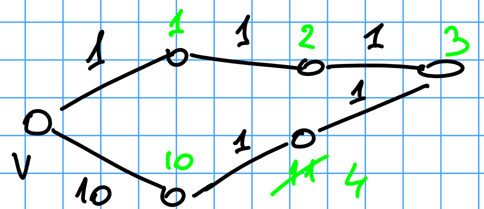
```
if dist[w] == ∞:
    dist[w] = dist[u] + 1
    Q.enqueue(w)
    prev[w] = u
```

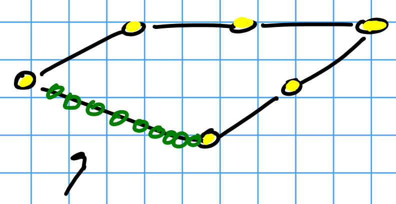
Утв (корректность):

$\forall w \quad dist(v, w) = k \qquad dist[w] = k$

1. База $k = 1$
2. Предположение: верно для $k = d$
3. Переход: докажем для $k = d + 1$. $\triangleleft$

Кратчайшие пути во взвеш. графах





Алгоритм Дейкстры:

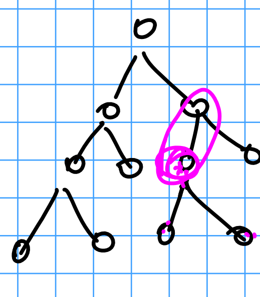
Очередь с приоритетами:

- Make-priority-queue()
- Delete-min()
- Insert(v, d)
- Decrease-key(v, d)

Реализации:

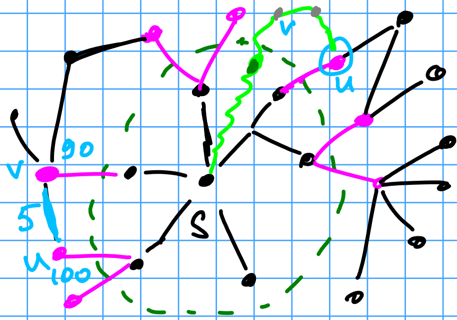
- на массиве
- на куче

В куче $O(\log n)$



```
Dijkstra(s)
  for v ∈ V:
      dist[v] = ∞
      prev[v] = 0
  dist[s] = 0
  P = Make-priority-queue((s, 0))
  while not P.empty():
      (v, d) = P.Delete-min();
      for (v, u) ∈ E:
          if dist[u] == ∞:
              P.Insert((u, dist[v] + w(v, u)))
              dist[u] = dist[v] + w(v, u)
              prev[u] = v
          else if dist[v] + w(v, u) < dist[u]
              dist[u] = dist[v] + w(v, u)
              P.Decrease_key(u, dist[u])
              prev[u] = v
```

dist[v]



уже вычислили

Утв: (Корректность алг. Дейкстра)

Алгоритм Дейкстра разбивает V на три множества

S – уже обработанные

R – вершины в очереди (на границе)

T – вершины с $dist = \infty$.

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

Д. 1. База

$S = \{s\}$

2. Пусть для $\forall$ вершин в S вычислено правильное расстояние

3. Пусть на след. шаге $S \leftarrow S \cup \{v\}$

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

$\vartheta \to$ $\cancel{dist(s,v) > dist[v]}$

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

$\exists$ путь $v_1, v_2, \dots, v_k$, где $s = v_1$, $v = v_k$:

длина этого пути $< dist[v]$

$\angle$ v_i – последняя вершина на этом пути из S.

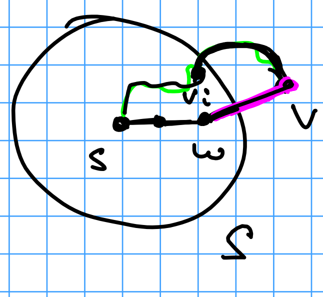
$\Rightarrow$ $v_{i+1} \in R$

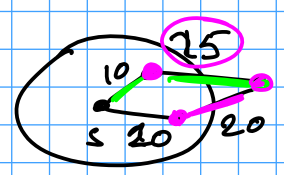
$dist[v_{i+1}] < dist[v]$

$(v_1 \dots v_i)$ – имеет длину $dist[v_i]$

$dist[v_{i+1}] \le dist[v_i] + w(v_i, v_{i+1}) \le$

$\le$ длина $(v_1, \dots, v_k) < dist[v]$





$\triangleleft$

Сложность: 1 Make-priority-Queue()

V × Insert

V × Delete-Min

E × Decrease-Key

I На массиве:

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

II на куче:

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

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

III

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

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

$$d \approx \frac{E}{V} \qquad E \approx V^2$$