Rem. В BFS заменить очередь q на стек s, то получится нерекурсивный DFS.

(2) Алгоритм Дейкстры

Взвеш. граф, $\forall w_e \geq 0$.

Интуиция:

BFS, но из очереди берём вершину с min $d[v]$ („жадный" алгоритм)

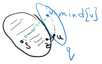
```
Dijkstra(s):
    d = [+∞] * |V|;  q = new PriorityQueue()        1). (d, v)
    d[s] = 0;  q.insert((d[s], s))                   2). keys = d
                                                         values = v
    while not q.empty():
        dv, v = q.extract_min()
        for (vu) ∈ E:                                 v --w_vu--> u
            if d[u] > d[v] + w_vu:                    d[v]+w_vu
                [ q.remove((d[u], u))   // u -> где хранится      (d[u],u) -> (d'[u],u)
                [ d[u] = d[v] + w_vu
                [ q.insert((d[u], u))

                [ d[u] = d[v] + w_vu
                [ q.change_key(u, d[u])   // все (d[v], v) изначально д.б. положены в q
```

Th. Алгоритм Дейкстры корректен: когда вершина v достаётся из q, для неё $d[v] = \rho(s,v)$. **Rem.** Верш. обраб. ≤ 1 раз.

Lm 1. В любой момент $d[v] \geq \rho(s,v)$.

Индукция по количеству обр. вершин. Ув: у всех обр. вершин $d[v] = \rho(s,v)$.

„Обработ." = достали из q и обр. все исх. рёбра



Достаём из q вершину v с min $d[v]$.

Утв: $d[v] = \rho(s,v)$.

От противного. Кратч. путь $p: s \to v$.

(*) $(d[v] > \rho(s,v))$

z – последняя вершина в p, кот. обр.

u – следующая за ней в p

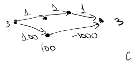
1). $d[v] \leq d[u]$ $\quad v, u$ – не обр. $\quad v$ имеет min $d[v]$.

2). длина $p = \rho(s,v) = \underbrace{\rho(s,z)}_{=d[z]} + w_{zu} + \rho(u,v)$

3). вершина z обработана $\Rightarrow d[u] \leq d[z] + w_{zu} = \rho(s,z) + w_{zu}$

$$\rho(s,v) \overset{2)+3)}{\geq} d[u] + \rho(u,v) \overset{1)}{\geq} \underbrace{d[v]}_{> \rho(s,v)} + \underbrace{\rho(u,v)}_{\geq 0} > \rho(s,v) + 0 \qquad (?!)$$

(*) по предполож. [пользуемся тем, что $\forall w_e \geq 0$]



$w'_e = w_e + p_v - p_u \quad w'_e \geq 0$, $e = vu$

(потенциалы Джонсона)

Время работы:

(Память: $O(|V|)$)

Списки смежности

1). Очередь с приоритетом – массив $\quad d[v]$, $used[v]$ – обр. ли вершина

extract_min: for $u \in V$:
    if not $used[u]$:
        $m = \min(m, (d[u], u))$

$O(|V|)$

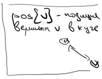
$\forall$ вершины: extract_min + проходим по рёбрам

$O(V) + O(\deg v)$

плотные графы: $|E| \sim |V|^2$

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

2). Очередь с приоритетом – куча



$$Time = \sum_v (\log|V| + \deg v \cdot \log|V|) =$$
$$O(|V| \cdot \log|V| + E \log|V|) =$$
$$= O(|E| \log|V|)$$

разреж. графы: $|E| \ll |V|^2$

---

(3) w_e – произвольные

Алгоритм Форда – Беллмана

Состояние ДП: (v, # рёбер до неё)

- $d[v][k]$ = min длина пути из s в v, кот. состоит из $\leq k$ рёбер
  $v \in V$
  $k = 0..|V|-1$
- $d[v][0] = \begin{cases} 0, & v = s \\ +\infty, & v \neq s \end{cases}$
- $d[v][k+1] = \min(d[v][k], \min\limits_{u:(uv) \in E} d[u][k] + w_{uv})$
- Ответ: $d[v][|V|-1] = \rho(s,v)$ (если нет отрицательных циклов)

```
for k = 0..|V|-2
    d[v][k+1] = d[v][k]  ∀ v ∈ V
    for e = (uv) ∈ E:
        d[v][k+1] = min(d[v][k+1], d[u][k] + w_e)
```

$Time = O(|V| \cdot |E|)$

$Mem = O(|V|^2) \to O(|V|)$

```
d[s] = 0; d[v≠s] = +∞
for k = 0..|V|-2
    for e = (uv) ∈ E
        d'[v] = min(d'[v], d'[u] + w_e)
```

$s \to$ все вершины

$\forall v, u \quad \rho(v,u) = ?$

$w_e \geq 0 \to O(V \cdot V^2)$
$\quad \to O(V \cdot E \log V)$

w_e – произв. $\to V \cdot O(VE) = O(V^2 E)$

Алгоритм Флойда – Уоршелла

$d[k][v][u]$ – min путь $v \to u$, исп. вершины с номерами $1..k$

(*) $d[k][v][u] = \min(d[k-1][v][u], d[k-1][v][k] + d[k-1][k][u])$


```
for k = 1..V
    for v = 1..V
        for u = 1..V
            (*)
```

$Time = O(V^3) \qquad Mem = O(V^3)$

$d[v][u] = \min(d[v][u], d[v][k] + d[k][u])$