Weight $w(p)$ of path
$p = (v_0, v_1, \ldots, v_k)$ is the sum:

$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

Define shortest-path weight $\delta(u, v)$ from u to v by

$$\delta(u,v) = \begin{cases} \min\{w(p) : u \overset{p}{\rightsquigarrow} v\} & \text{if there is a path from } u \text{ to } v \\ \infty & \text{otherwise} \end{cases}$$

Алгоритмы нахождения кратчайшего пути

Dijkstra's algorithm
relax each edge exactly once
быстрее

Bellman-Ford algorithm
relaxes each edge $|V|-1$ times
медленнее, но может работать с отрицательными весами

Initialization

```
Initialize-single-source(G, s)
  for each vertex v ∈ G.V
      v.d = ∞
      v.π = NIL
  s.d = 0
```

Relaxation

```
Relax(u, v, w)
  if v.d > u.d + w(u,v)
      v.d = u.d
      v.π = u
```

Путь: $s \rightsquigarrow u$

Ребро: $s \to v$

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅,
Q = G.V
while Q ≠ ∅,
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

ТОЛЬКО ДЛЯ РЁБЕР НЕОТРИЦАТЕЛЬНОГО ВЕСА

$O(n^2)$

кол-во рёбер

```
Bellman-Ford(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
for i = 1 to |G.V| - 1
    for each edge (u, v) ∈ G.E
        RELAX(u, v, w)
for each edge (u, v) ∈ G.E
    if v.d > u.d + w(u, v)
        return FALSE
return TRUE
```

релаксация выполняется n-1 раз

$O(n \cdot m)$

кол-во верш.

кол-во рёбер

Обязательно будут на экзамене:

→ корректность

↳ Прима

↳ Дейкстры

↳ Беллмана-Форда

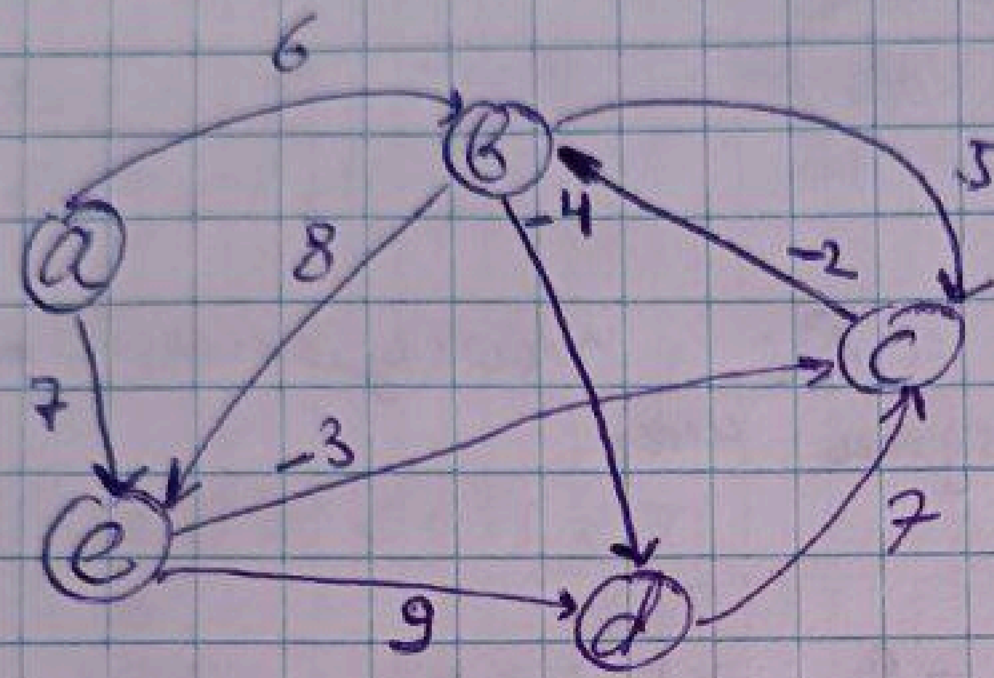
Поиск кратчайшего пути

Алгоритмы

→ Дейкстры

→ Форда-Беллмана

Алгоритм



$\omega: E \to R$

1) Выпишем матрицу весов:

$$
W = \begin{array}{c|ccccc}
 & a & b & c & d & e \\
\hline
a & 0 & 6 & \infty & \infty & 7 \\
b & \infty & 0 & 5 & -4 & 8 \\
c & \infty & -2 & 0 & \infty & \infty \\
d & \infty & \infty & 7 & 0 & \infty \\
e & \infty & \infty & -3 & 9 & 0
\end{array}
$$

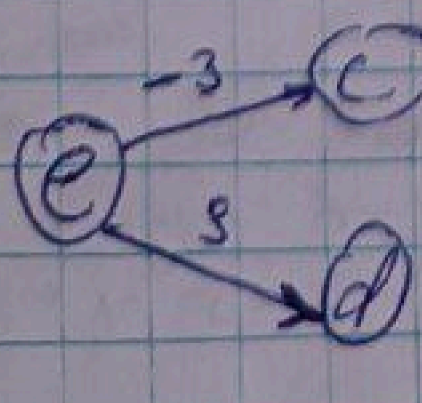
Вес пути равен ∞, если между двумя вершинами нет непосредственного пути

2) Берём любую вершину. Например, e:

$$D_1 = (\infty, \infty, -3, 9, 0)$$

a b

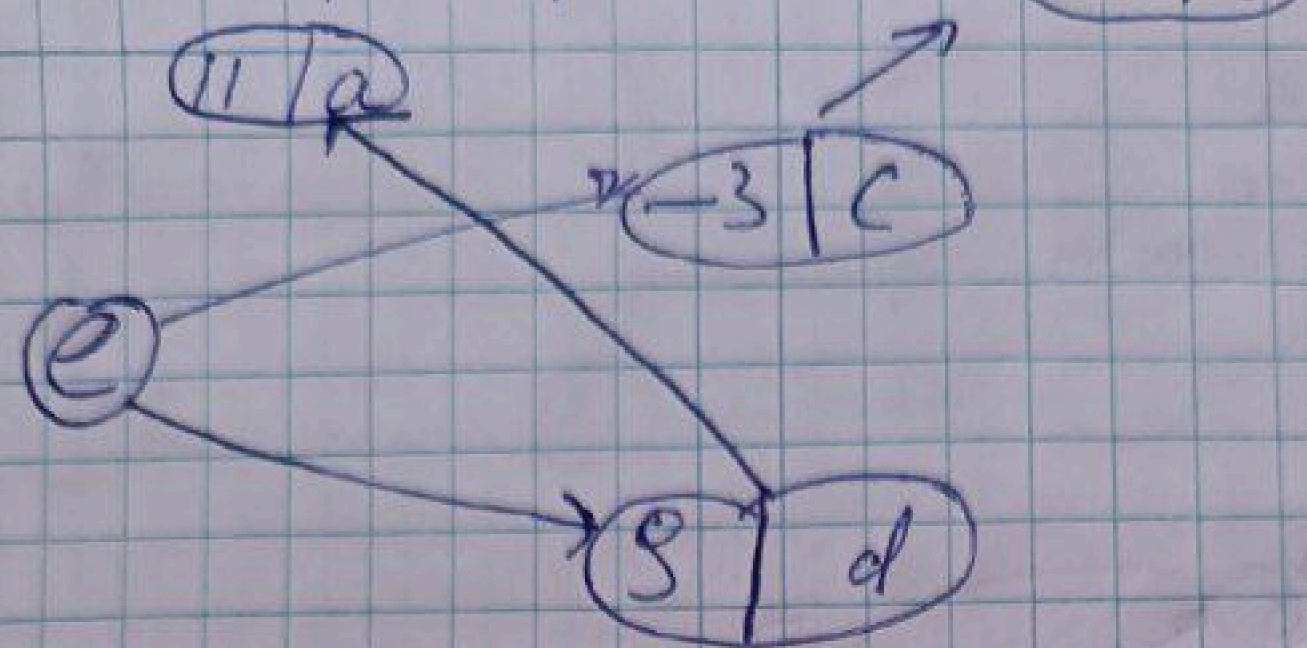
$$D_{i+1}[j] = \min(\{D_i[j]\} \cup \{D_i[k] + W_{kj}\})$$



Параллельно строим остов кратчайших путей. По сути, это вариация поиска в ширину.

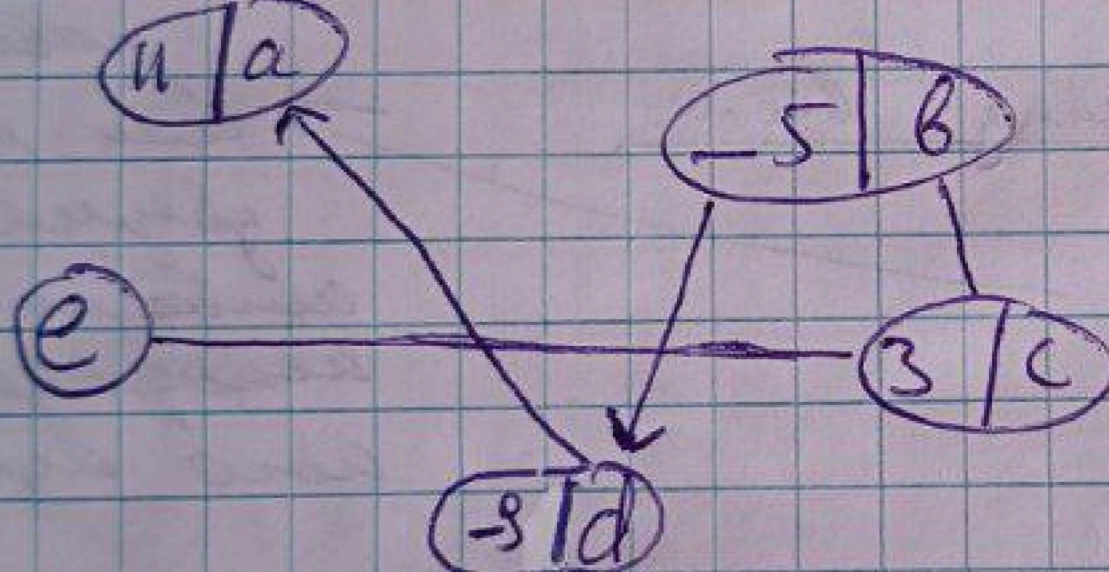
Всего шагов (макс.): $\boxed{n-1}$

$D_2 = (11, -5, -3, 9, 0)$



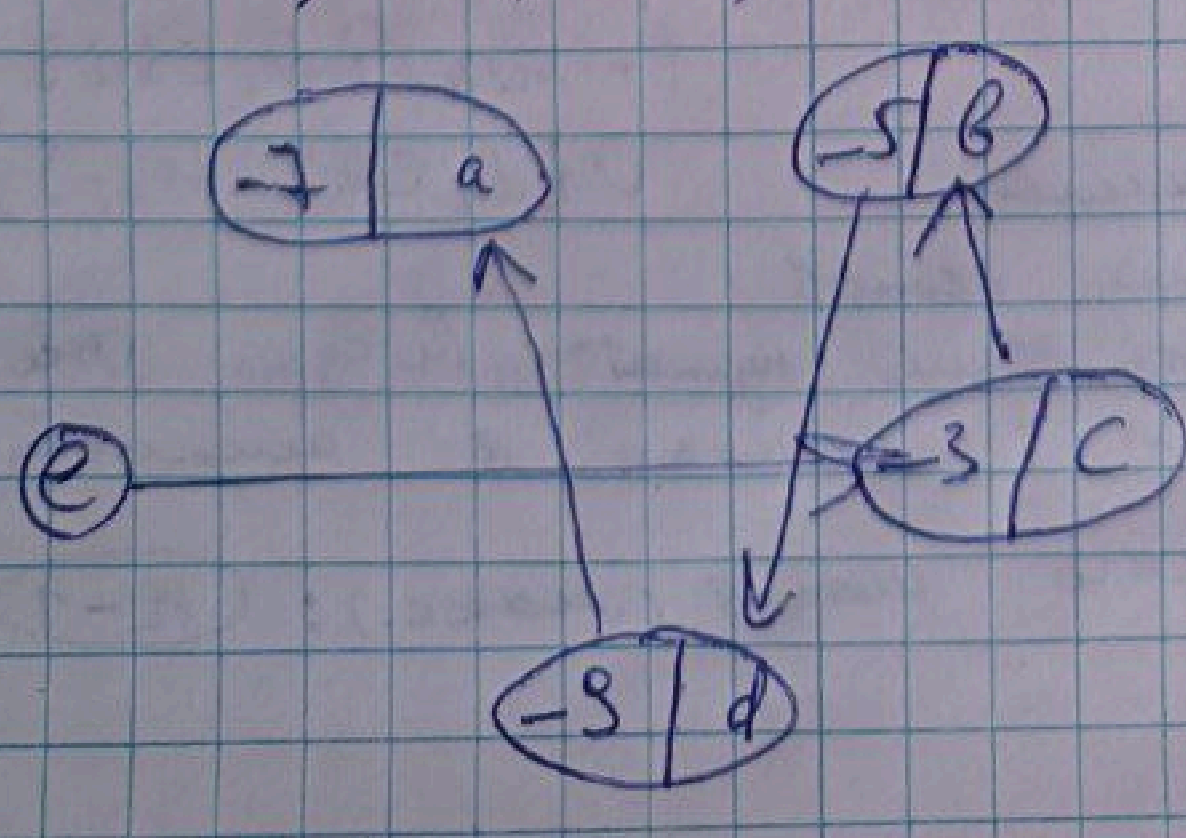
[Изменения между D_1 и D_2 произошли =>
=> делаем ещё один шаг

$D_3 = (11, -5, -3, -9, 0)$



[Изменения между D_2 и D_3 есть =>
=> продолжаем алгоритм]

$D_4 = (-7, -5, -3, -9, 0)$ — Последний шаг (т.к. макс. кол-во шагов будет $n-1=5-1=4$)



Кратчайшие пути. Матрицы смежности
Adjancency - matrix of graph
Input: an $n \times n$ matrix W

$W = \{w_{ij}\}$, where

0, if $i = j$;

w_{ij} = the weight of directed edge(i, j) if $i \neq j$ and $(i, j) \in E$,

∞ if $i \neq j$ and $(i, j) \notin E$.

Predecessor subgraph

For each vertex $i \in V$, we define the predecessor subgraph of G for i as

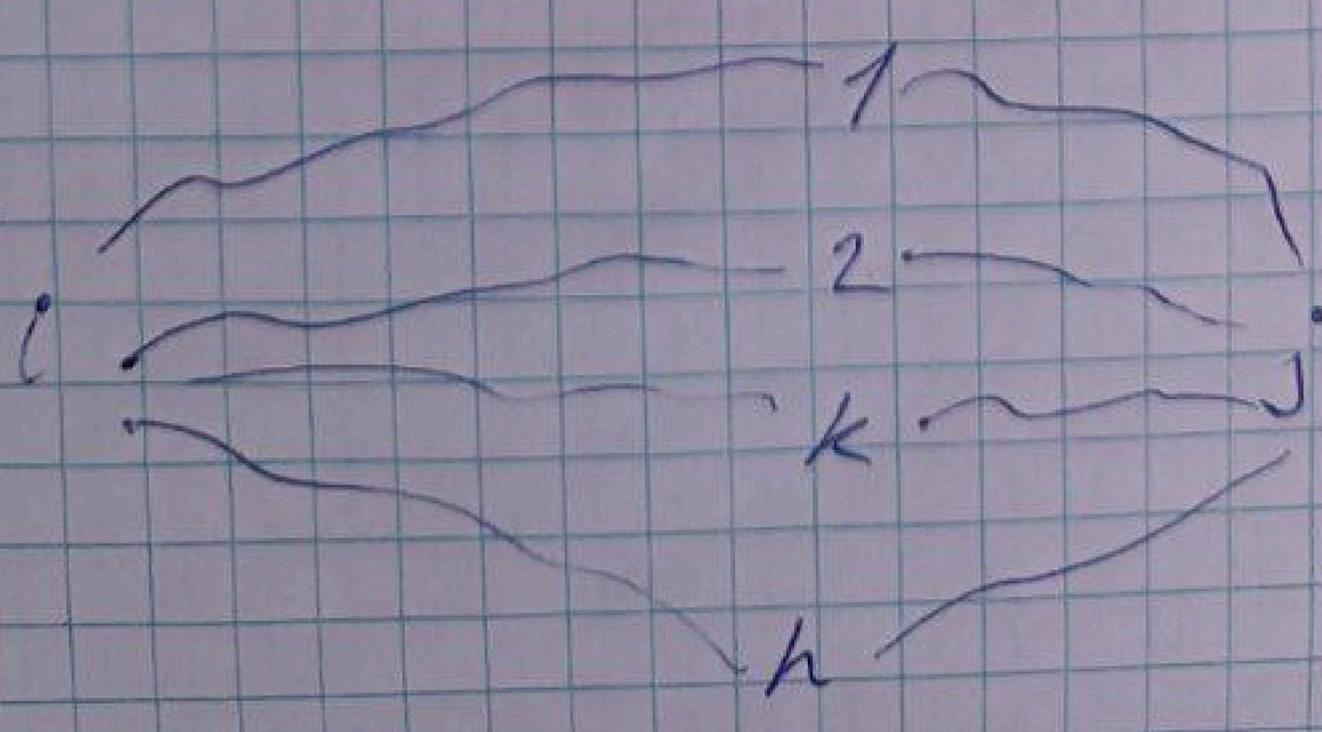
$G_{\pi, i} = (V_{\pi, i}, E_{\pi, i})$,

where

$V_{\pi, i} = \{ i \in V_{\xi} : \pi_{ij} \neq NIL \} \cup \{i\}$

and

$E_{\pi, i} = \{ (\pi_{ij}, j) : j \in V_{\pi, i} - \{i\} \}$.



EXTEND-SHORTEST-PATHS(L, W)

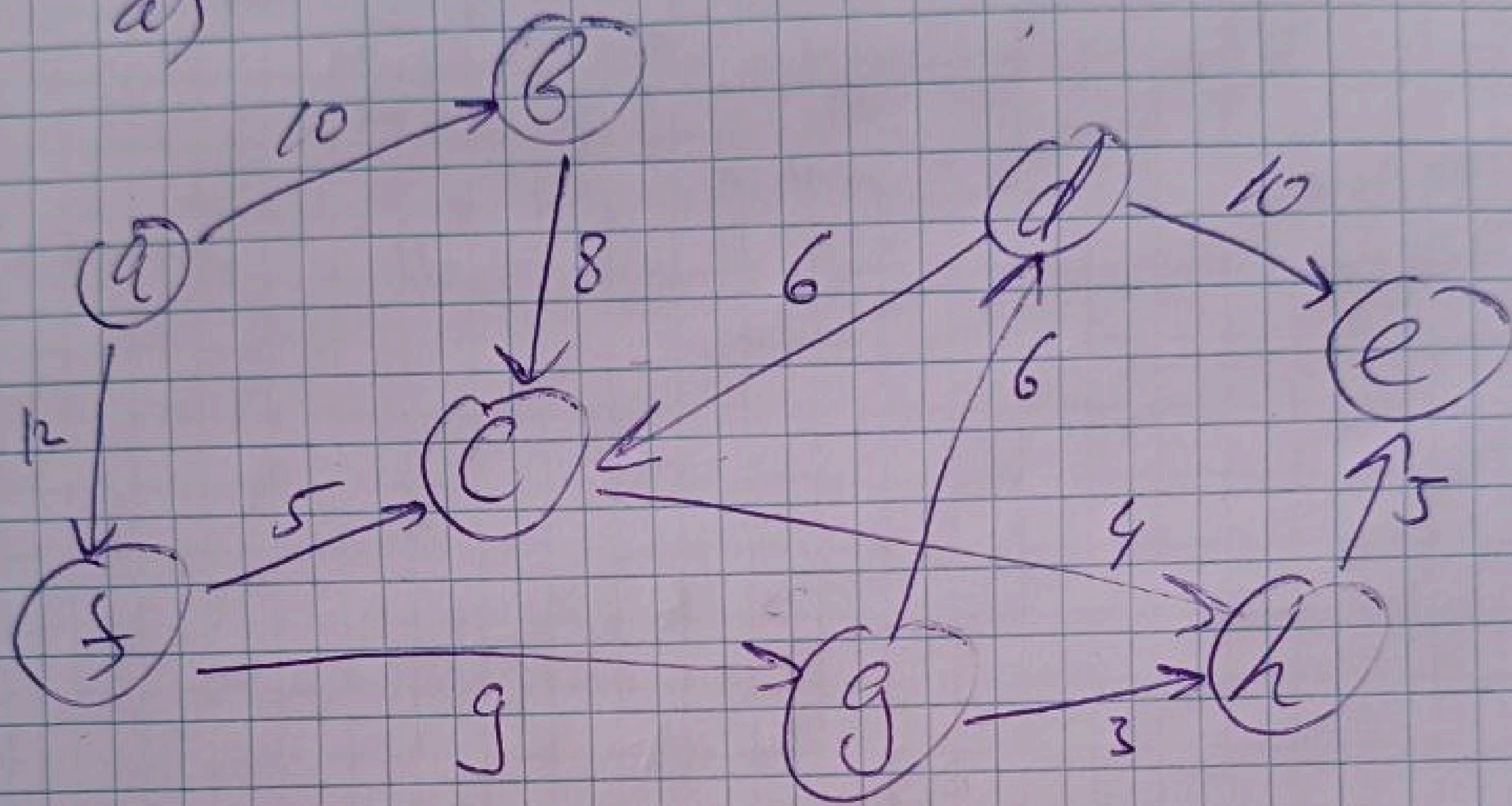
```
n = L.rows
let L' = (l_ij') be a new
   n×n matrix
for i = 1 to n
      for j = 1
            l_ij = ∞
            for k = 1 to n
                  l_ij' = min(l_ij', l_ik + w_kj)
return L'
```

Алгоритм Дейкстры.

1 Жадный алгоритм

1.

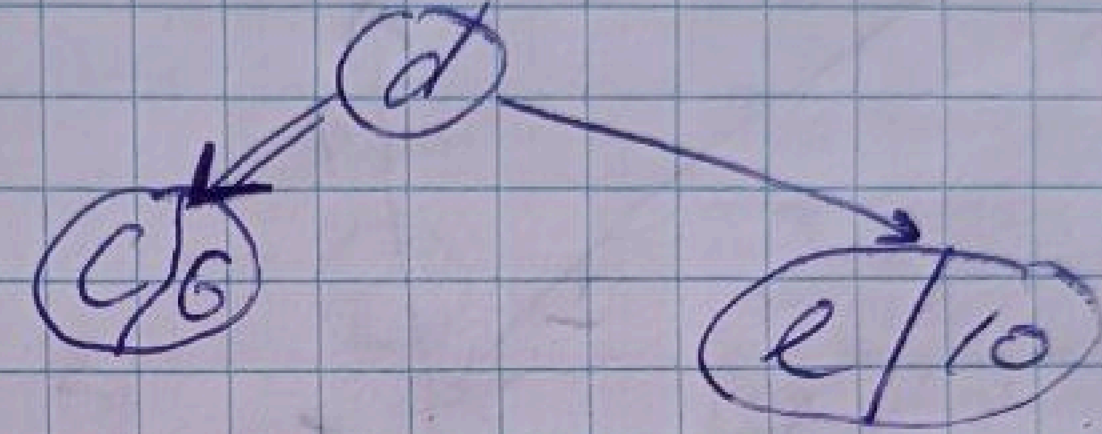
а)



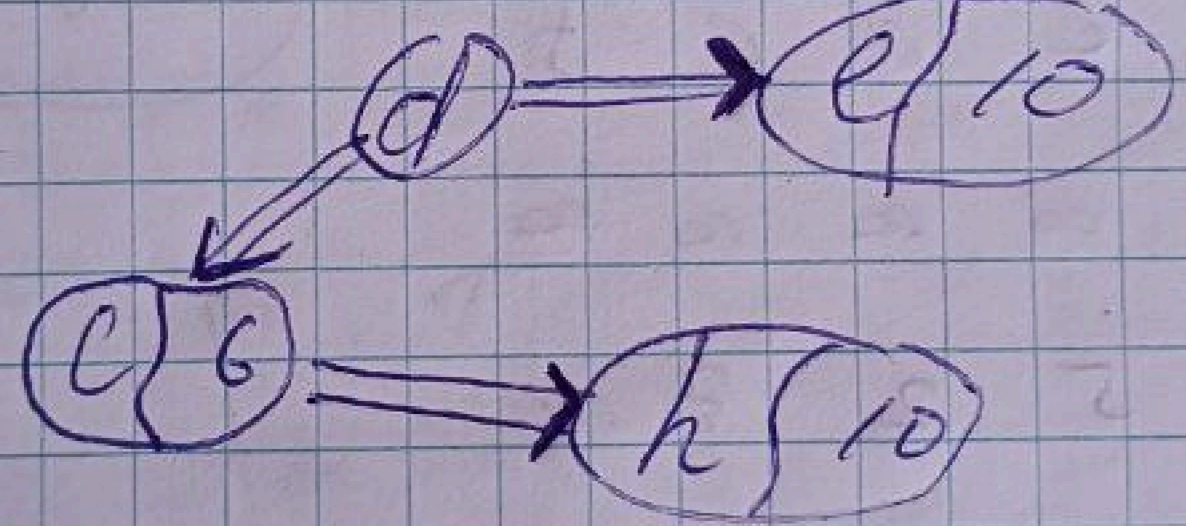
	a	b	c	d	e	f	g	h
a	0	10	∞	∞	∞	12	∞	∞
b	∞	0	8	∞	∞	∞	∞	∞
c	∞	∞	0	∞	∞	∞	∞	4
d	∞	∞	6	0	10	∞	∞	∞
e	∞	∞	∞	∞	0	∞	∞	∞
f	∞	∞	5	∞	∞	0	9	∞
g	∞	∞	∞	6	∞	∞	0	3
h	∞	∞	∞	∞	5	∞	∞	0

$S = \{d\}$ $\quad D_1 = (\infty, \infty, \underline{6}, \underline{0}, 10, \infty, \infty, \infty)$

$S = \{d, c\}$

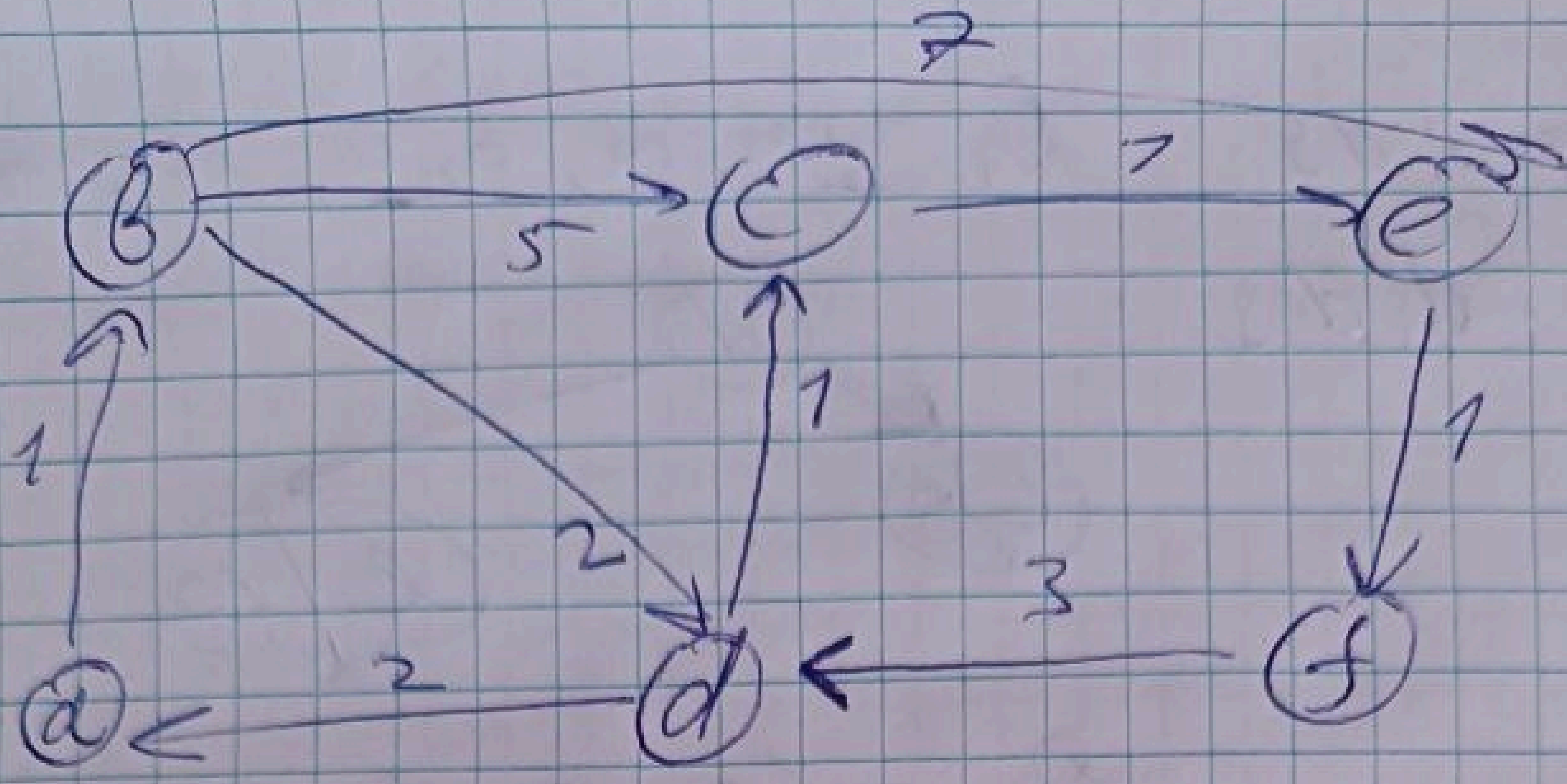


$D_2 = (\infty, \infty, \underline{6}, \underline{0}, \underline{10}, \infty, \infty, \underline{10})$



$S = \{d, c, e, h\}$ $\quad D_3 = (\infty, \infty, 6, 0, 10, \infty, \infty, 10)$

на этом шаге D_{i-1} и D_i совпали ($D_2 = D_3$) поэтому дальше по алгоритму можно не идти б)



Пойдём из c.

	a	b	c	d	e	f
a	0	1	∞	∞	∞	∞
b	∞	0	5	2	7	∞
c	∞	∞	0	∞	1	∞
d	2	∞	1	0	∞	∞
e	∞	∞	∞	∞	0	1
f	∞	∞	∞	3	∞	0

$S = \{c\}$, $D_1 = (\infty, \infty, \underline{0}, \infty, \underline{1}, \infty)$

$S = \{c, e\}$, $D_2 = (\infty, \infty, \underline{0}, \infty, \underline{1}, \underline{2})$

$S = \{c, e, f\}$, $D_3 = (\infty, \infty, \underline{0}, \underline{5}, \underline{1}, \underline{2})$

$S = \{c, e, f, d\}$, $D_4 = (\underline{7}, \infty, \underline{0}, \underline{5}, \underline{1}, \underline{2})$

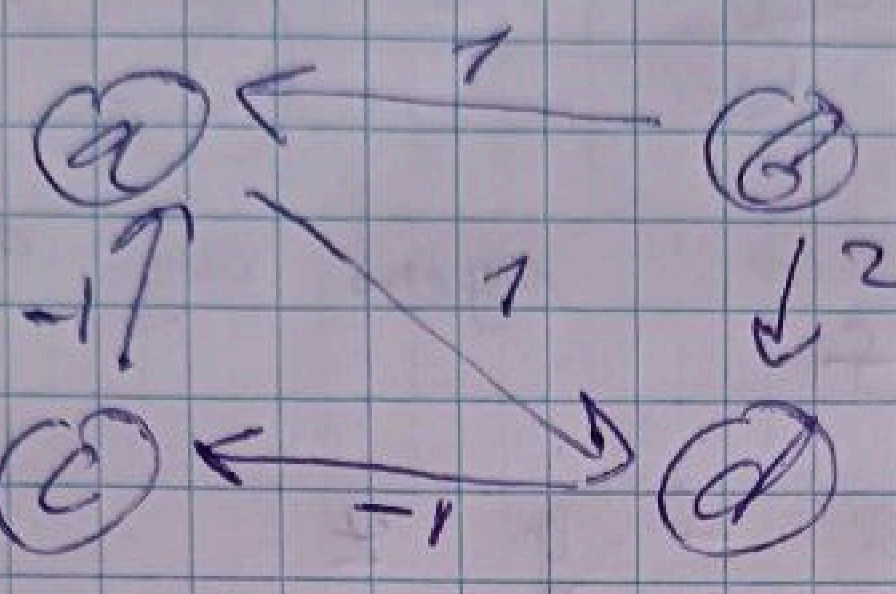
$S = \{c, e, f, d, a\}$, $D_5 = (\underline{7}, \underline{8}, \underline{0}, \underline{5}, \underline{1}, \underline{2})$

Алгоритм Дейкстры не работает на отрицательных весах.

Пример:

При попытках модификации алгоритма Дейкстры для графов с ребрами отрицательного веса получается алгоритм Беллмана-Форда

3.



	a	b	c	d
a	0	∞	∞	1
b	1	0	∞	2
c	-1	∞	0	∞
d	∞	∞	-1	0

$S = \{b\}, \quad D_1 = (\underline{1}, \underline{0}, \infty, 2)$

$S = \{b, a\}, \quad D_2 = (\underline{1}, \underline{0}, \underline{-1}, \underline{2})$

$S = \{b, a, c\}, \quad D_3 = (\underline{1}, \underline{0}, \underline{-1}, 2)$

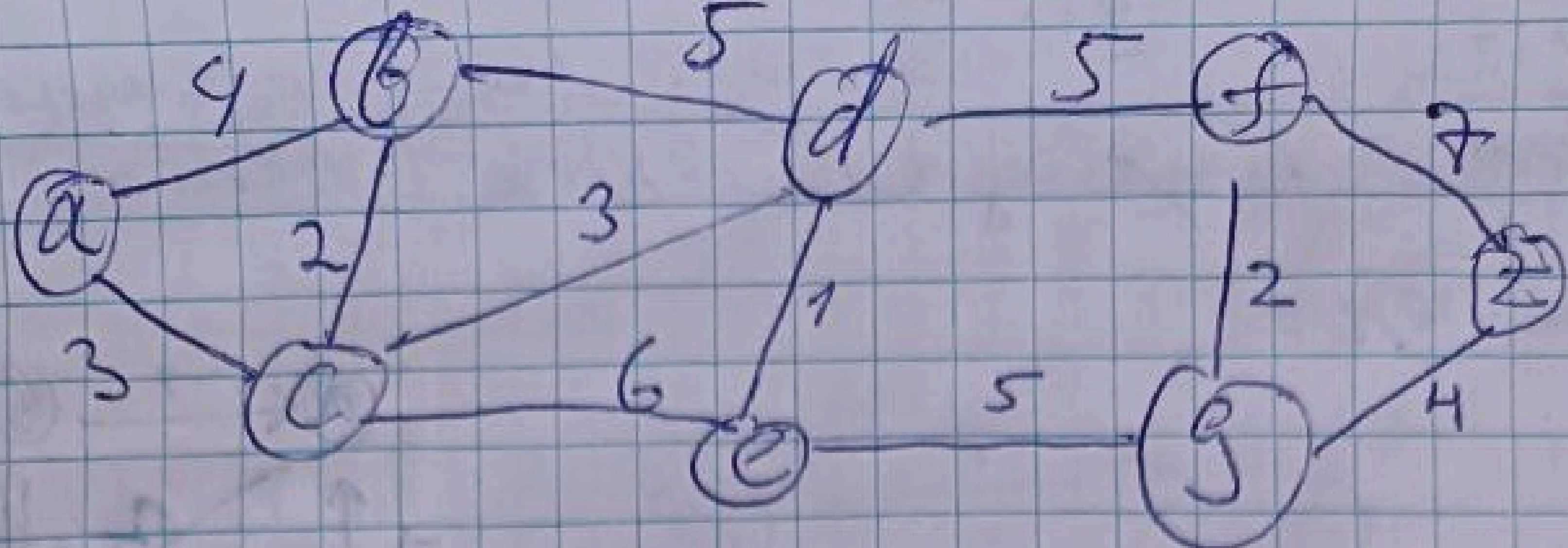
Из b в c есть путь веса 0:

$b \to d \to c \to a.$

Это кратчайший путь.

Алгоритм его не находит (показывает кр. путь длины 1)

3.



Найти мин. путь из a в z через f

	a	b	c	d	e	f	g	z
a	0	4	3	∞	∞	∞	∞	∞
b	4	0	2	5	∞	∞	∞	∞
c	3	2	0	3	6	∞	∞	∞
d	∞	5	3	0	1	5	∞	∞
e	∞	∞	6	1	0	∞	5	∞
f	∞	∞	∞	5	∞	0	2	7
g	∞	∞	∞	∞	5	2	0	4
z	∞	∞	∞	∞	∞	7	4	0

Из a в f:

$S=\{a\}$, $D_1=(0, 4, 3, \infty, \infty, \infty, \infty, \infty)$

$S=\{a, c\}$, $D_2=(0, 4, 3, 6, 9, \infty, \infty, \infty)$

$S=\{a, c, b\}$, $D_3=(0, 4, 3, 6, 7, 11, 14, \infty)$

$S=\{a, c, b, d\}$, $D_4=(0, 4, 3, 6, 7, 11, 12, 18)$

$S=\{a, c, b, d, e\}$, $D_5=(0, 4, 3, 6, 7, 11, 12, 18)$

$S=\{a, c, b, d, e, f\}$, $D_6=(0, 4, 3, 6, 7, 11, 12, 18)$

$\Rightarrow$ кратчайший путь из a в f = 11.

из f в z:

$S=\{f\}$, $D_1 = (\infty, \infty, \infty, 5, \infty, 0, 2, 7)$;

$S=\{f, g\}$, $D_2 = (\infty, 10, 8, 5, 6, 0, 2, 6)$;

$S=\{f, g, d\}$, $D_3 = (11, 10, 8, 5, 6, 0, 2, 6)$;

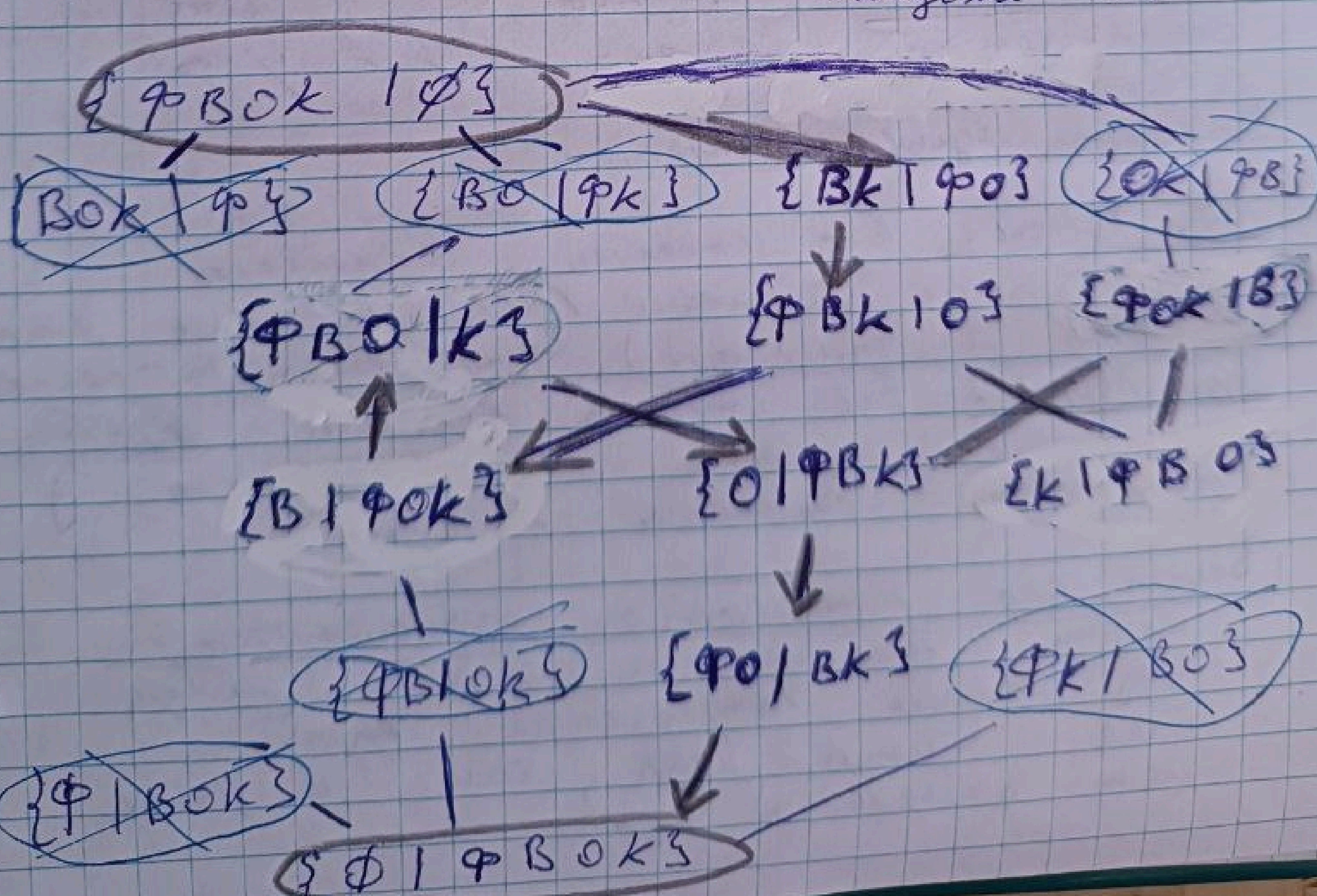
Кр. путь из f в z равен 6.

Путь из a в z через f:

$\underbrace{11}_{a \to f} + \underbrace{6}_{f \to z} = 17$

Задача про козу, капусту и волка

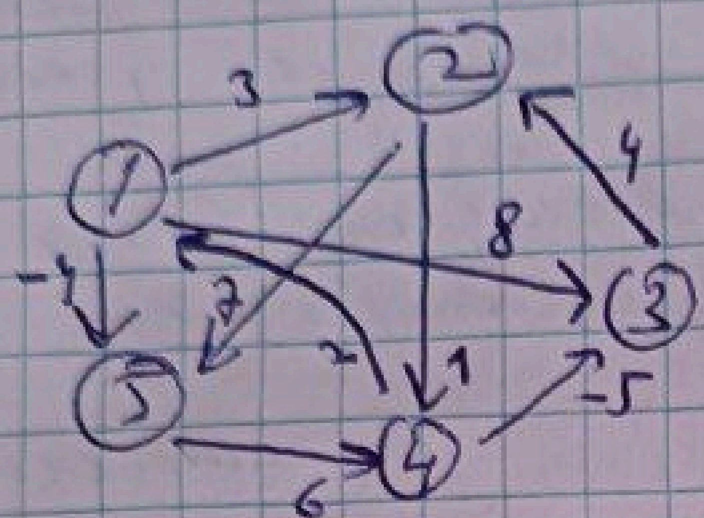
Фермер – Ф, волк – В, коза – О, капуста – К



Поиск кратчайших путей в графе.

→ Беллман – Форд

а)



$$W = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

выполнение за n шагов

$$A \cdot B = C$$

$$c_{ij} = \min\{\{a_{ik} + b_{kj} \mid k = 1 \ldots n\} \cup a_{ij}\}$$

$$D^{(0)} = W$$

$$D^{(1)} = D^{(0)} \cdot W = W^2$$

$$D^{(2)} = D^{(1)} \cdot W = D^{(0)} \cdot W^2 = W^3$$

$$\ldots$$

$$D^{(n-1)} = D^{(0)} W^{n-1} = W^n$$

↳ $O(n^4)$

↓ Медленный, но надёжный

↓ Можно параллельно строить деревья кратчайших путей

Можно стабилизировать алгоритм!

$$D^{(n-1)} = D^{(m)} \quad \forall\, m > n-1$$

$$W^{n} = W^{2^{k+1}} = \left(W^{2^k}\right)^2 = \left(W^2\right)^{\cdot^{\cdot^{\cdot^2}}}{}_{k+1}$$

$$2^k \le n < 2^{k+1}$$

$\hookrightarrow \quad O(n^3 \log(n))$

$$W^2 = WW = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} =$$

$\hookrightarrow$ если по диагонали 0, то в графе нет циклов отрицательной длины

$$= \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix}$$

$$W^4 = W^2 W^2 = \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix} \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$W^8 = W^4 W^4 = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

Совпали (значит, дальше более коротких путей НЕТ)

→ Флойд – Уоршелл

Пример:

$d^{1}_{13} = \min\{d^{0}_{13}, d^{0}_{11} + d^{0}_{13}\}$

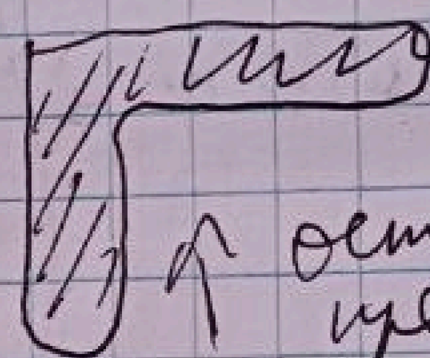
$D^{(0)} = W$

$D^{(k)}: \quad d^{k}_{ij} = \underbrace{\min\{d^{k-1}_{ij}, \; d^{k-1}_{ik} + d^{k-1}_{kj}\}}$

$d^{1}_{53} = \min\{d^{0}_{53}, \; d^{0}_{51} + d^{0}_{13}\}$

выполнение за фиксированную константу (один/два шага) (смотря что называть шагом)

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$



остаётся прежним всегда

стали меньше на этом шаге

если мы уверены в том, что нет циклов отриц. длины

если строка начинается с бесконечности то она гарантировано не изменится

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

В довесок к этому алгоритму предлагается вести матрицу предшественников

$$\Rightarrow \quad R^0_{ij} = \begin{cases} N, & i=j \text{ или } w_{ij}=\infty \\ i, & \text{иначе} \end{cases}$$

$$R^k_{ij} = \begin{cases} R^{k-1}_{ij}, & d^{k-1}_{ij} \le d^{k-1}_{ik} + d^{k-1}_{kj} \\ R^{k-1}_{kj}, & \text{иначе} \end{cases}$$

$$R^0 = \begin{pmatrix} N & 1 & 1 & N & 1 \\ N & N & N & 2 & 2 \\ N & 3 & N & N & N \\ 4 & N & 4 & N & N \\ N & N & N & 5 & N \end{pmatrix}$$

$$R^1 = \begin{pmatrix} N & 1 & 1 & N & 1 \\ N & N & N & 2 & 2 \\ N & 3 & N & N & N \\ 4 & 1 & 4 & N & 1 \\ N & N & N & 5 & N \end{pmatrix}$$

R^1_{42}

$\downarrow$

$d^0_{42} = \infty$

$d^0_{41} + d^0_{12} = 2 + 3 = 5$

$R^1_{42} = R^0_{12}$

$$R^2 = \begin{pmatrix} N & 1 & 1 & 2 & 1 \\ N & N & N & 2 & 2 \\ N & 3 & N & 2 & N \\ 4 & 1 & 4 & N & 1 \\ N & N & N & 5 & N \end{pmatrix}$$

$$R^5 = \begin{pmatrix} N & 3 & 4 & 5 & 1 \\ 4 & N & 4 & 2 & 1 \\ 4 & 3 & 4 & N & 1 \\ 4 & 3 & 4 & N & 1 \\ 4 & 3 & 4 & 5 & N \end{pmatrix}$$

$$R^3 = \begin{pmatrix} N & 1 & 1 & 2 & 1 \\ N & N & N & 2 & 2 \\ N & 3 & N & 2 & N \\ 4 & 3 & 4 & N & 1 \\ N & N & N & 5 & N \end{pmatrix}$$

$$R^4 = \begin{pmatrix} N & 1 & 4 & 2 & 1 \\ 4 & N & 4 & 2 & 1 \\ 4 & 3 & N & 2 & 1 \\ 4 & 3 & 4 & N & 1 \\ 4 & 3 & 4 & 5 & N \end{pmatrix}$$

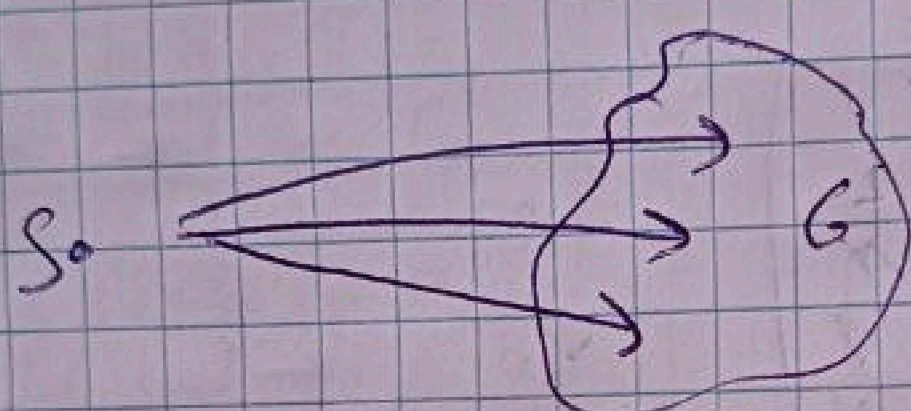
→ Джонсон

У графа есть отрицательные веса.

Если бы их не было, мы бы запустили Дейкстру.

Цель: придумать, как избавиться от отриц. весов.

Идея: добавим вершину s:



$$G' = \langle V \cup \{s\},$$
$$E \cup \{(s, v) \mid v \in V\}\rangle,$$

$$w: \ w_{sv} = 0$$
$$w_{vs} = \infty$$
$$v \neq s$$

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & 0 & 3 & 8 & \infty & -4 \\ \infty & \infty & 0 & \infty & 1 & 7 \\ \infty & \infty & 4 & 0 & \infty & \infty \\ \infty & 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D_0^0 = (0\ 0\ 0\ 0\ 0\ 0)$$
$$D_1^0 = (0\ 0\ 0\ -5\ 0\ -4)$$
$$D_2 = (0, 0, -1, -5, 0, -4)$$
$$D_5 = (0, \underset{h(1)}{0}, \underset{h(2)}{-1}, \underset{h(3)}{-5}, \underset{h(4)}{0}, \underset{h(5)}{-4})$$

$$W': \quad w'_{ij} = w_{ij} + h(i) - h(j)$$
$$h(i) = \sigma(s, i)$$

$$W^1 = \begin{pmatrix} 0 & 4 & 13 & \infty & 0 \\ \infty & 0 & \infty & 0 & \\ \infty & 0 & 0 & \infty & \infty \\ 2 & \infty & 0 & 0 & \infty \\ \infty & \infty & \infty & 2 & 0 \end{pmatrix}$$

Далее запускаем алгоритм Дейкстры

(из любой вершины)

$T_0 = \{2\}$ $D_2^{1,0} = (\infty \; \underline{0} \; \infty \; \underline{0} \; 10)$

$T_1 = \{2, 4\}$ $D = (2 \; \underline{0} \; \underline{0} \; \underline{0} \; 10)$

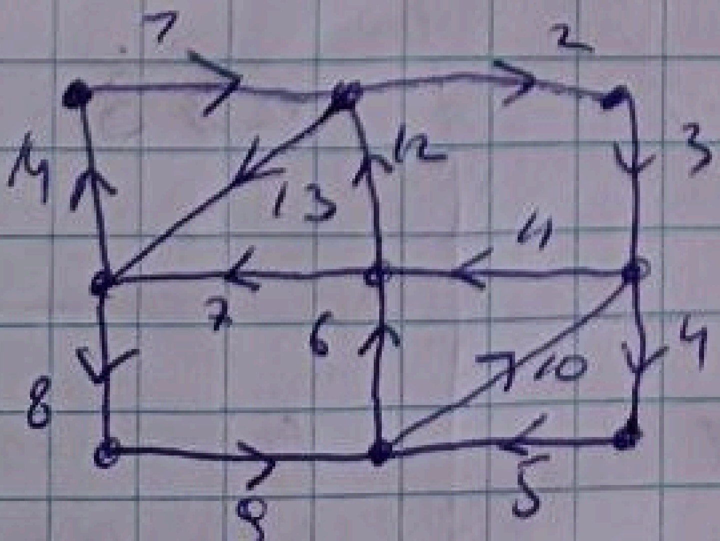
$2 \; 0 \; 0$

Эйлеровы и гамильтоновы графы

Если все вершины чётной степени, то есть эйлеров цикл.

Если все вершины, кроме двух, чётной степени, то есть эйлеров путь.

1. а) Цикл (эйл.): б) Ничего



в)