

$$L(1) = \{2(4)\}$$

$$L(2) = \{3(7), 4(2), 6(5)\}$$

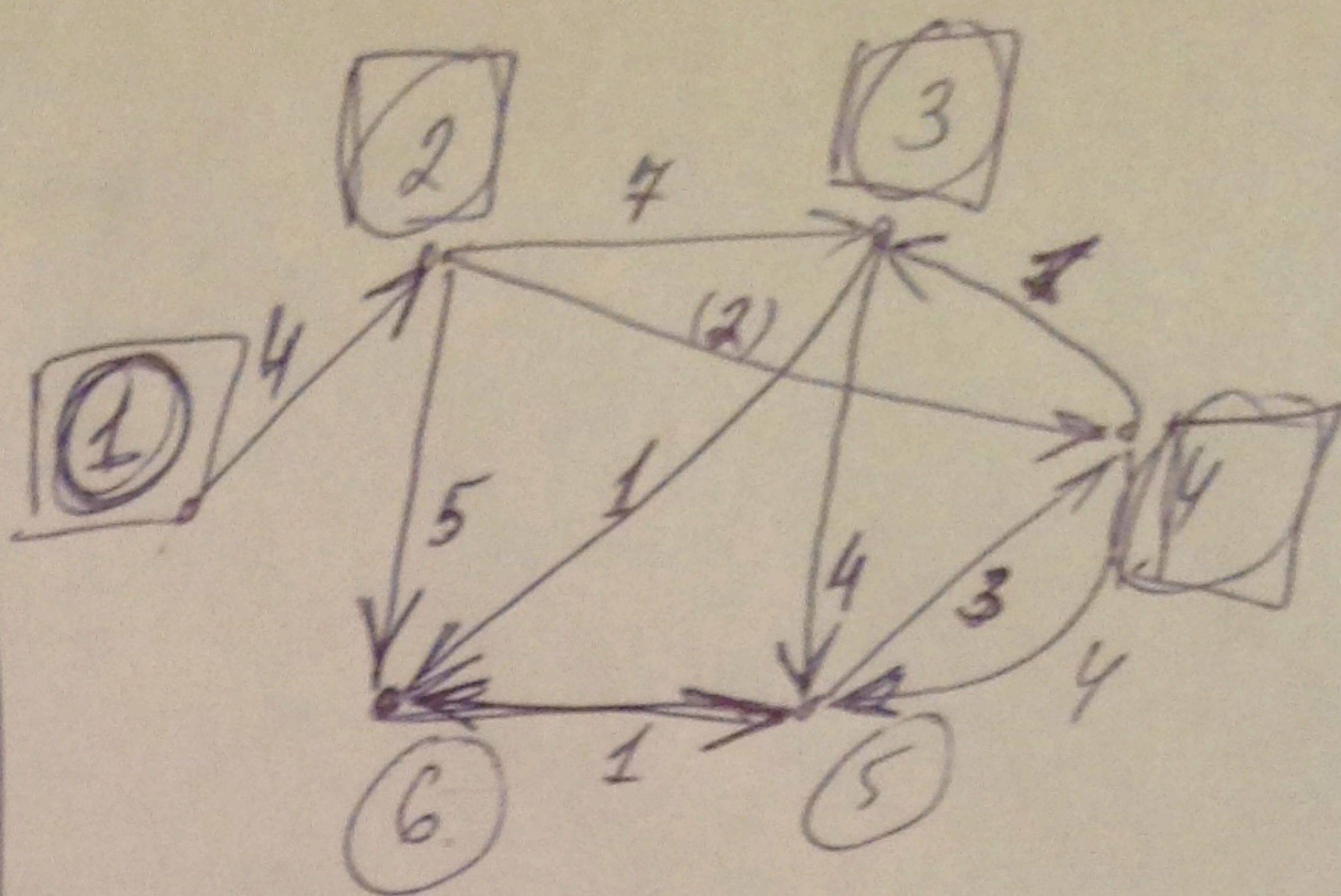
$$L(3) = \{6(1)\}$$

$$L(4) = \{3(1), 5(4)\}$$

$$L(5) = \{4(3)\}$$

$$L(6) = \{5(1)\}$$

$$S = 1$$



S	$\checkmark$ D[1]	$\checkmark$ D[2]	$\checkmark$ D[3]	$\checkmark$ D[4]	$\checkmark$ D[5]	$\checkmark$ D[6]
1	0	4	∞	∞	∞	∞
1, 2	0	4	11	6	∞	9
1, 2, 4	0	4	7	6	10	9
1, 2, 3, 4	0	4	7	6	10	8
1, 2, 3, 4, 6	0	4	7	6	9	8
1, 2, 3, 4, 5, 6	0	4	7	6	9	8

$$Prev(2) = 1$$

$$Prev(3) = 4$$

$$Prev(4) = 2$$

$$Prev(5) = 6$$

$$Prev(6) = 3$$

$$S = 1 \quad t = 8$$

$$L(1) = \{2(4), 3(9)\}$$

$$L(2) = \{3(5), 6(6)\}$$

$$L(3) = \{5(10)\}$$

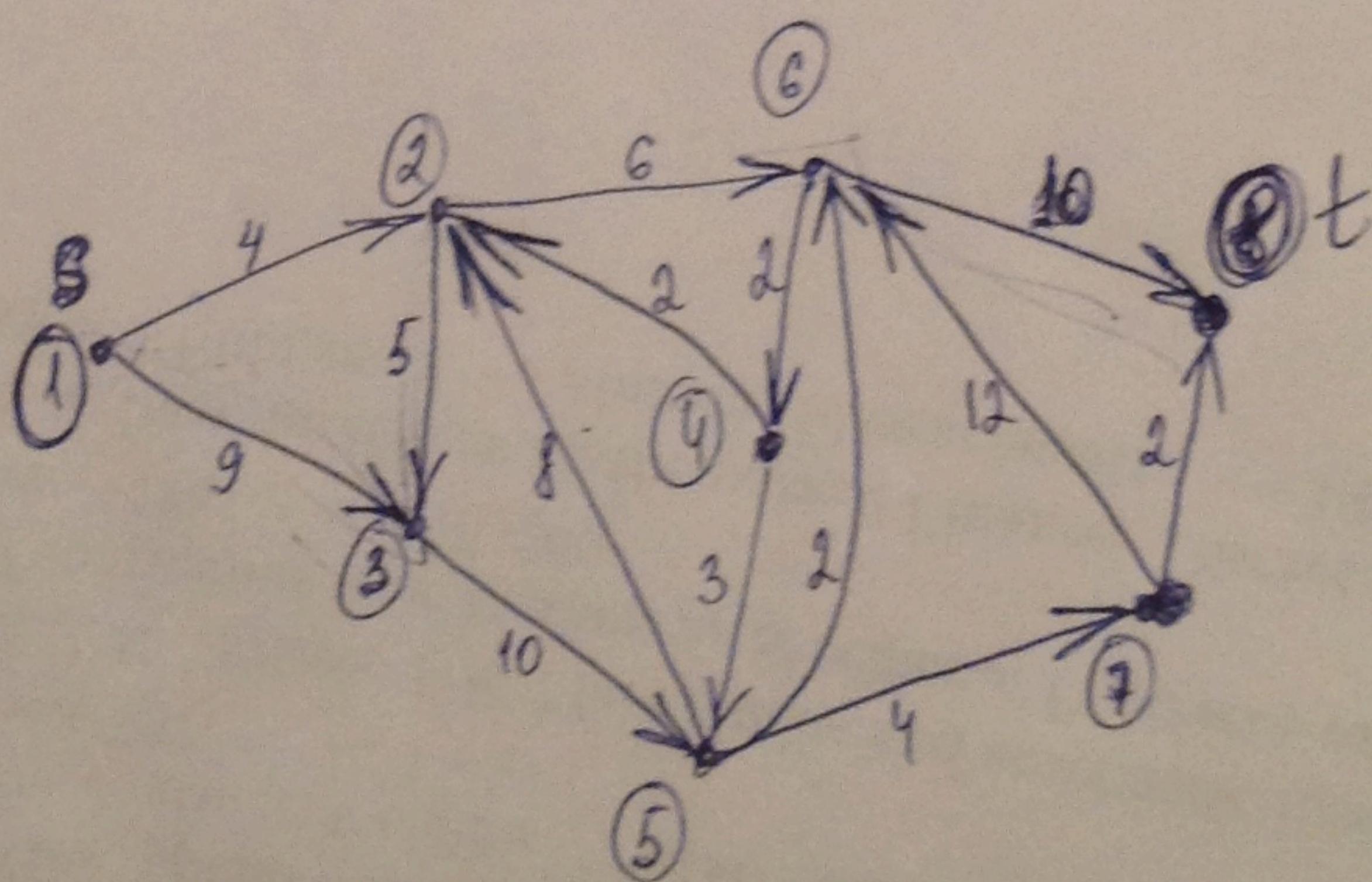
$$L(4) = \{2(2), 5(3)\}$$

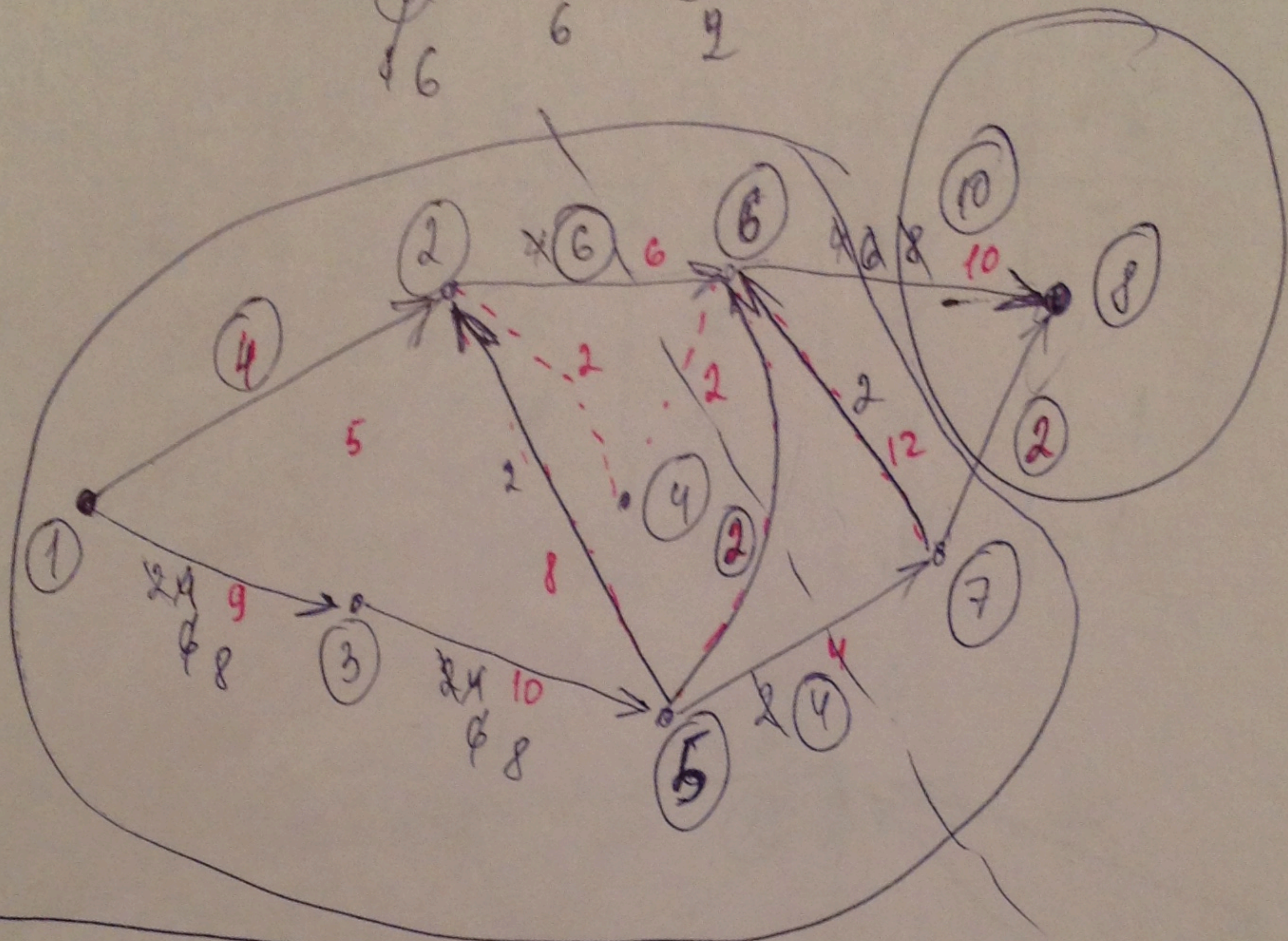
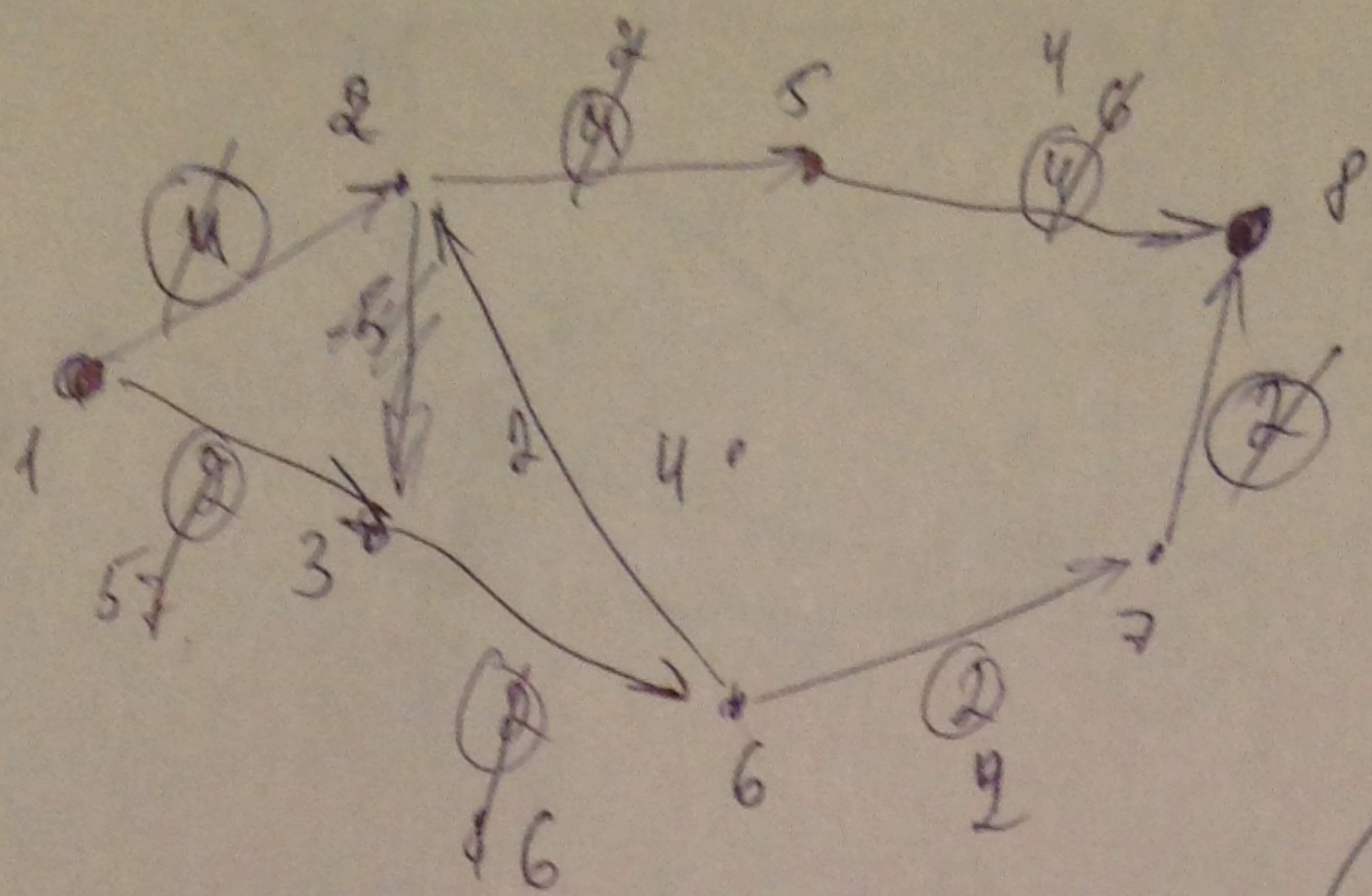
$$L(5) = \{2(8), 6(2), 7(4)\}$$

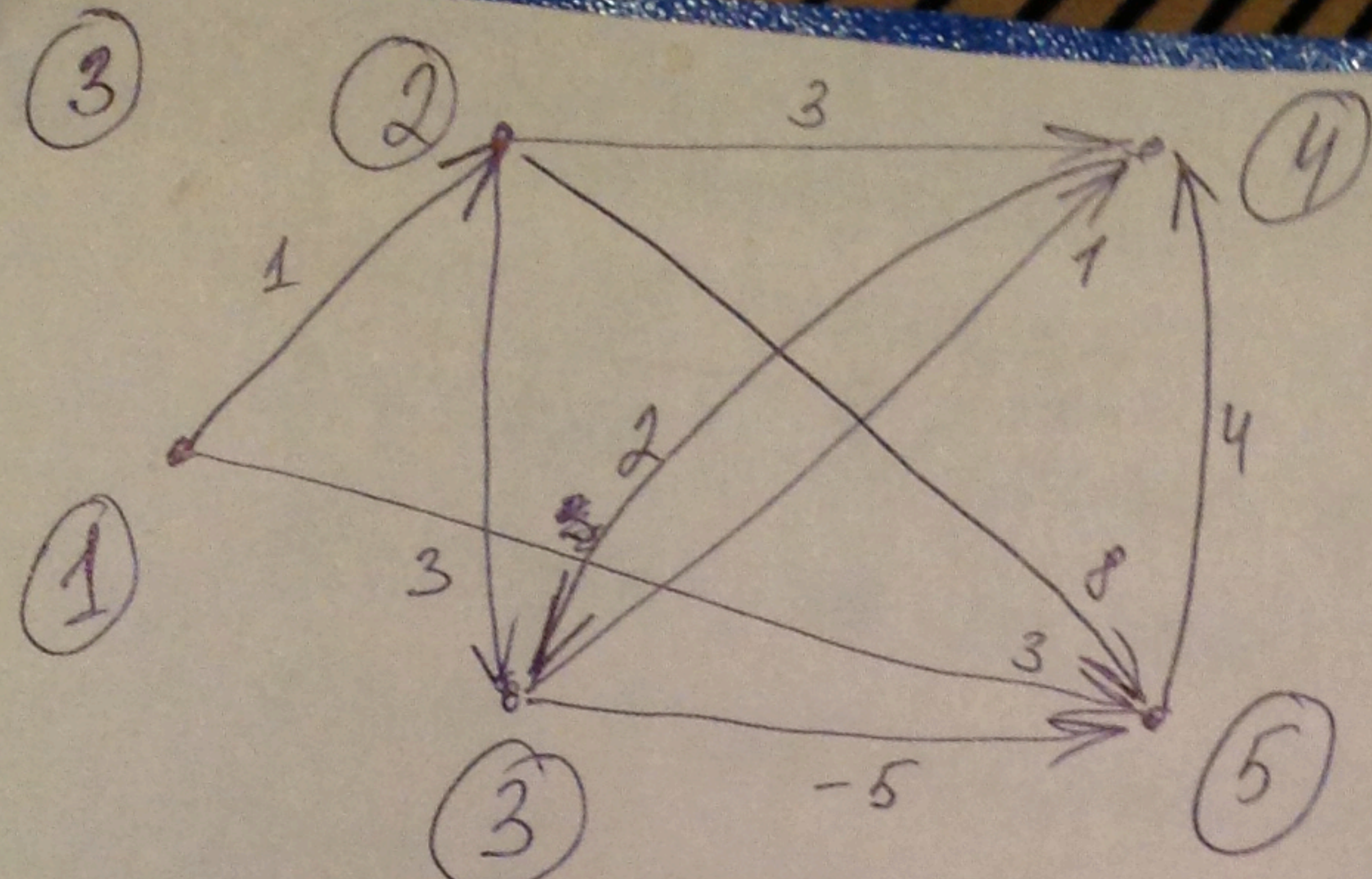
$$L(6) = \{4(2), 8(10)\}$$

$$L(7) = \{6(2), 8(2)\}$$

$$L(8) = \{\emptyset\}$$

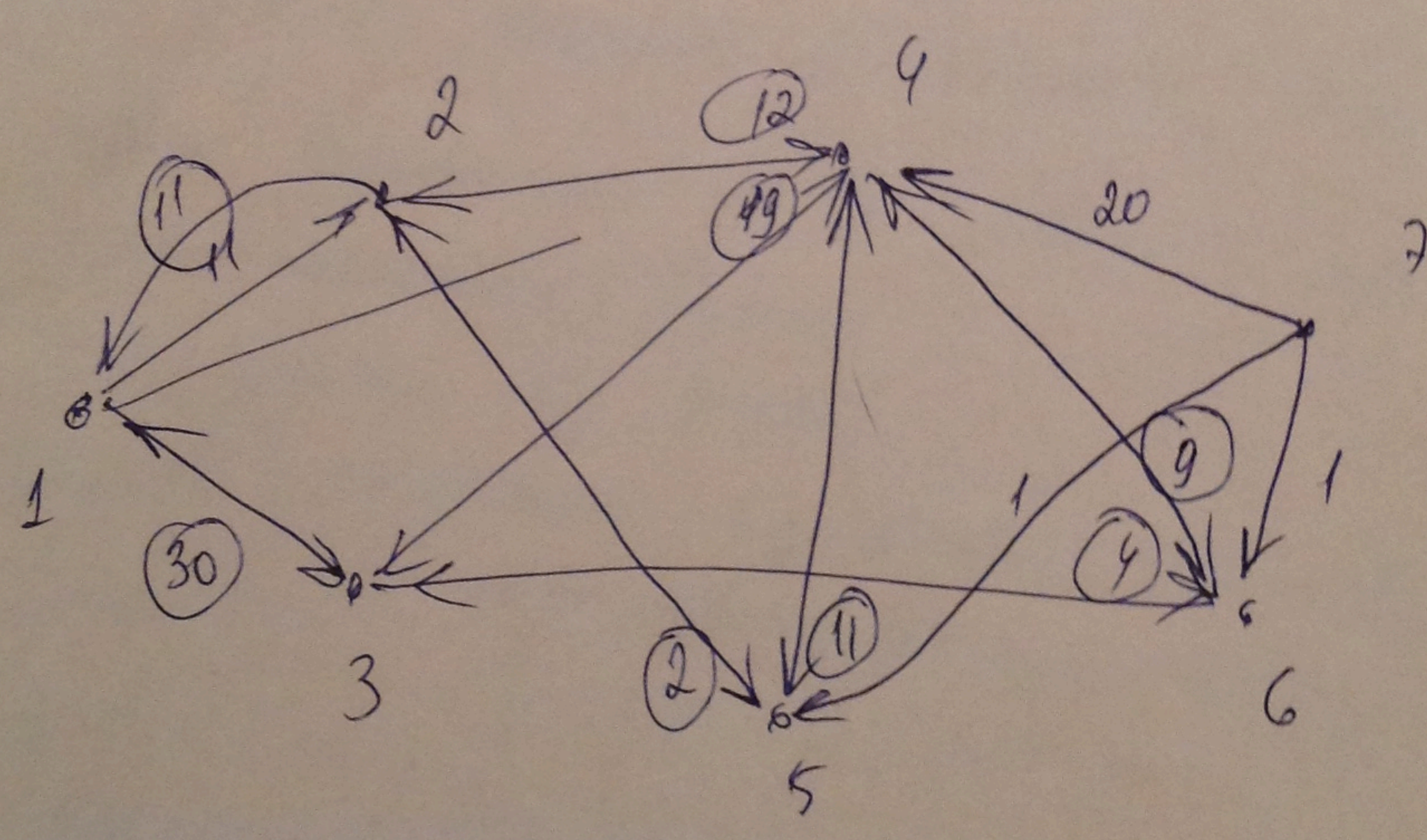






$L(1) = \{2(1), 5(3)\}$
 $L(2) = \{3(3), 4(3), 5(8)\}$
 $L(3) = \{5(-5), 4(1)\}$
 $L(4) = \{3(2)\}$
 $L(5) = \{4(4)\}$

K	D ₁	D ₂	D ₃	D ₄	D ₅
0	0	1	∞	∞	3
1	0	1	4	4	3
2	0	1	4	4	-1
3	0	1	4	3	-1



Поиск в ширину

$G = (V, E)$ - связный простой граф

S - стек, x - y реб. - верш. графа G

$r \in V$ - корень (нач. верш. поиска)

$r \rightarrow$ в стек, r - посещ. верш.

u - последняя посещ. вершина - верш. стека
($u = r$ в начале поиска)

1. Среди вершин, смежных с u есть ещё непосещенные вершины v

$v \rightarrow$ в стек

$u = \text{prev}(v)$

v - посещ. верш.

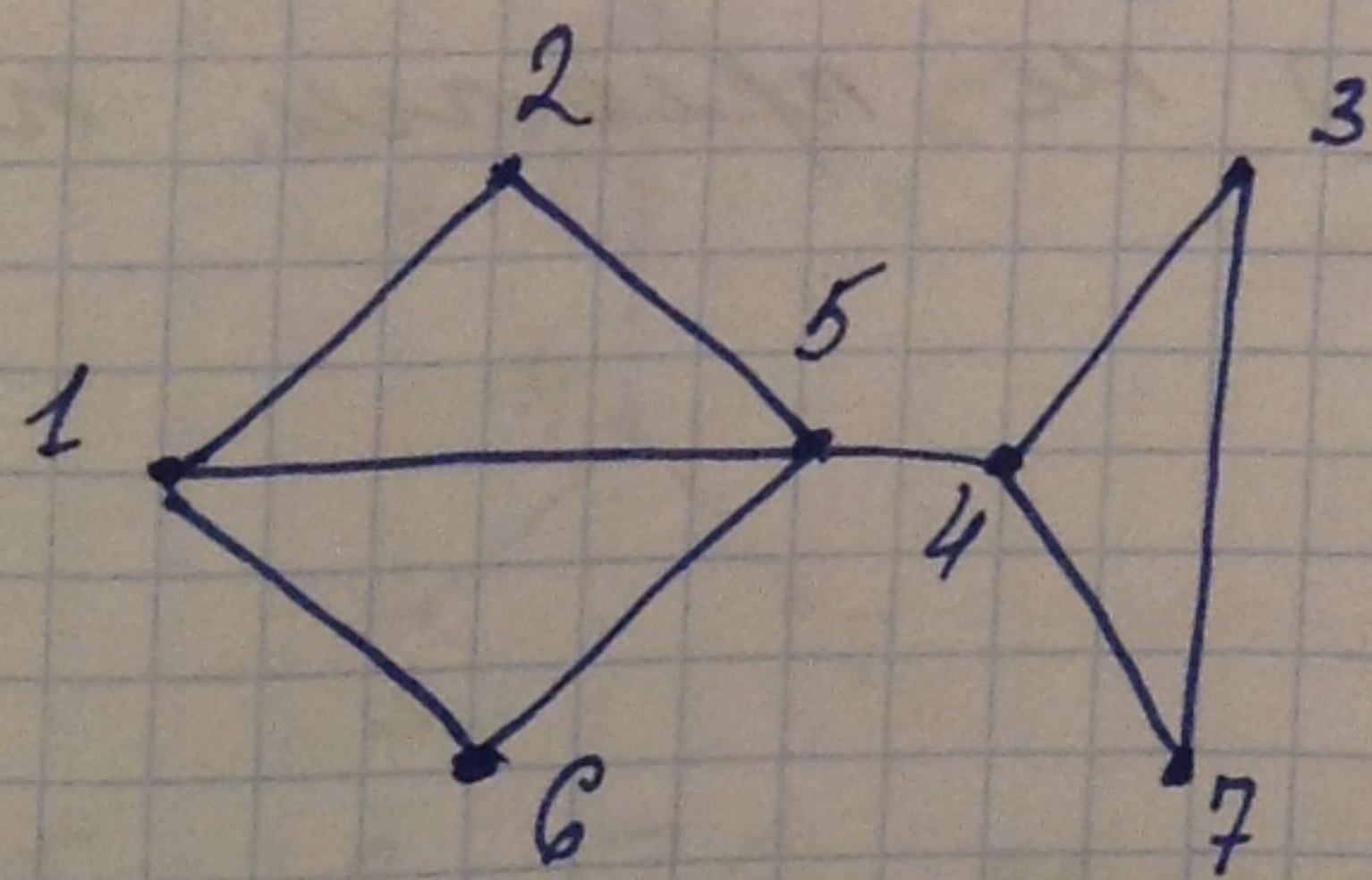
Ребро (u, v) включ. в дерево поиска

2. Все смежные с u верш. - посещённые
 u - исполз. верш.

u удач. из стека

Поиск продолжит из верш. w - верш. стека
(из w мы перейдем в u)

$\text{num}[v] = \begin{cases} 1, & v = r \\ \text{num}[u] + 1, & \text{если } u = \text{prev}[v] \end{cases}$



$h = 1$ $S = 1 \ 2$

$L(1) : 2, 5, 6$

$L(2) : 1, 5$

$L(3) : 4, 7$

$L(4) : 3, 5, 7$

$L(5) : 1, 2, 4, 6$

$L(6) : 1, 5$

$L(7) : 3, 4$

2 Теор. $\exists G = (V, E)$ - связный простой (n, m) граф
 $n = |V|, m = |E| \Rightarrow$

- 1) поиск в глубину посещает $\forall$ верш.
ровно 1 раз
- 2) граф $T = (V, E_T)$ - дерево
- 2) поиск в глубину имеет
сложность $O(m+n)$

Д-во: 1) $\forall$ верш. посещена не более 1 раз

$X \subseteq V$ - мн-во посещ-х вершин

$Y = V \setminus X$ - мн-во непосещ. верш.

G связн. $\Rightarrow x \in X, y \in Y \exists$ ребро $(x, y) \in E \Rightarrow$
верш. y будет посещ. из x

- 2) Условие присв. номера вершин во мн.
($n-1$) раз

$|E_T| = n-1$ циклов отсутств.

$T = (V, E_T)$ - дерево

- 3) К-во записей в стек не превышает
сумму степеней всех вершин
 $O(m)$

К-во записей в мн $[v]$ не превышает n
 $O(m+n)$

Поиск в ширину

Q - очередь, n, m кот. - вершины графа
 $r \in V$ - нач. верш. поиска - корень дерева
 r записывается в очередь, обьект. посещ.

u - первая вершина в очереди

$u_1, \dots, u_r$, смежные с u непосещ. верш.

Верш. $u_1, \dots, u_r$ запис. в очередь в порядке
 их следования в списке смежности
 (смежности)

для $i = 1$: $u = prev[u_i], i = \overline{1, r}$

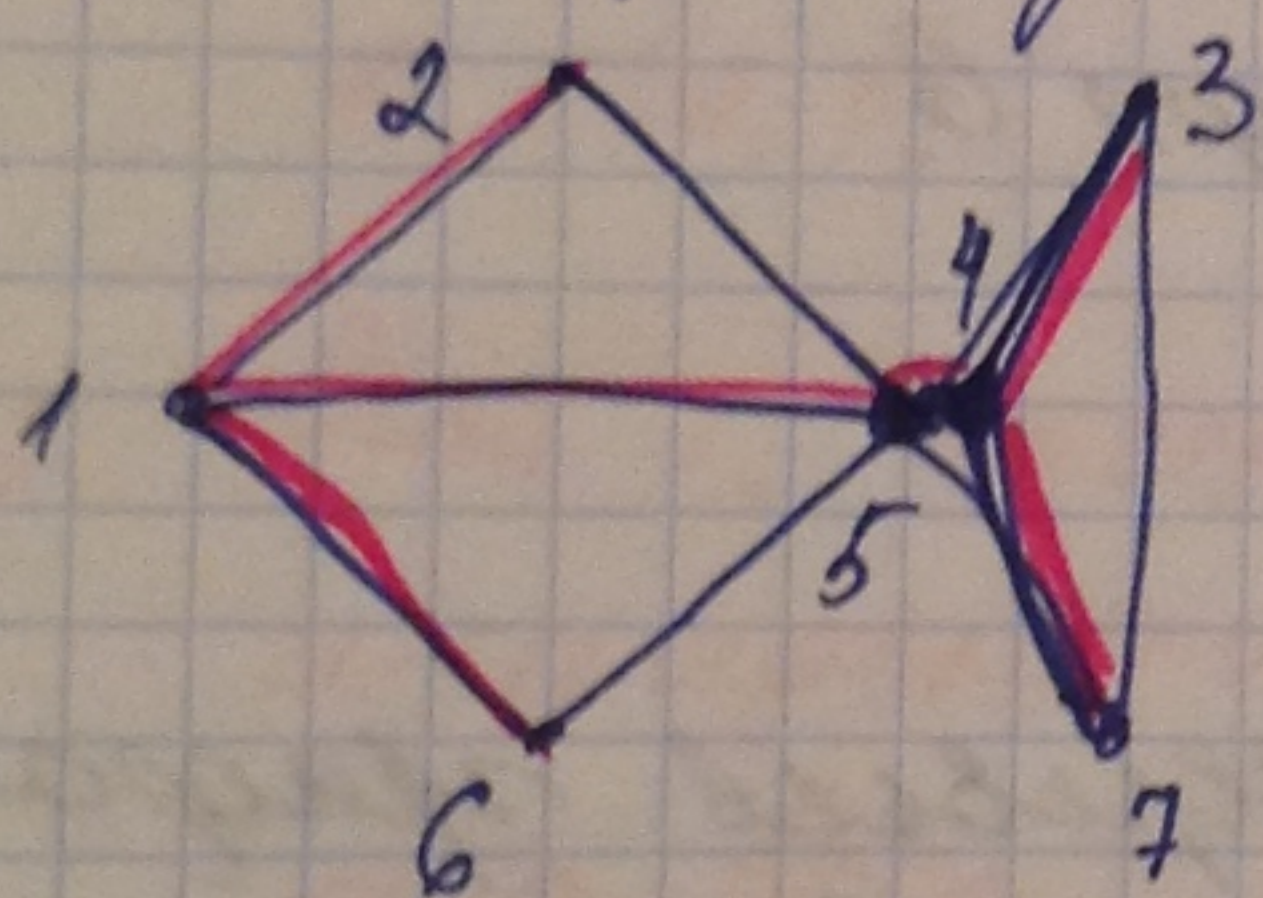
(u, u_i) обьект.
 ребрами дерева
 поиска

$num[u_i] = num[u] + 1$

$num := num + 1$

Верш. u - использованная и удал. из
 очереди

Очередь пуста - поиск заканчивается



$L(1) = \boxed{2, 5, 6}$

$L(5) = \boxed{1, 2, 4, 6}$

$L(2) = \boxed{1, 5}$

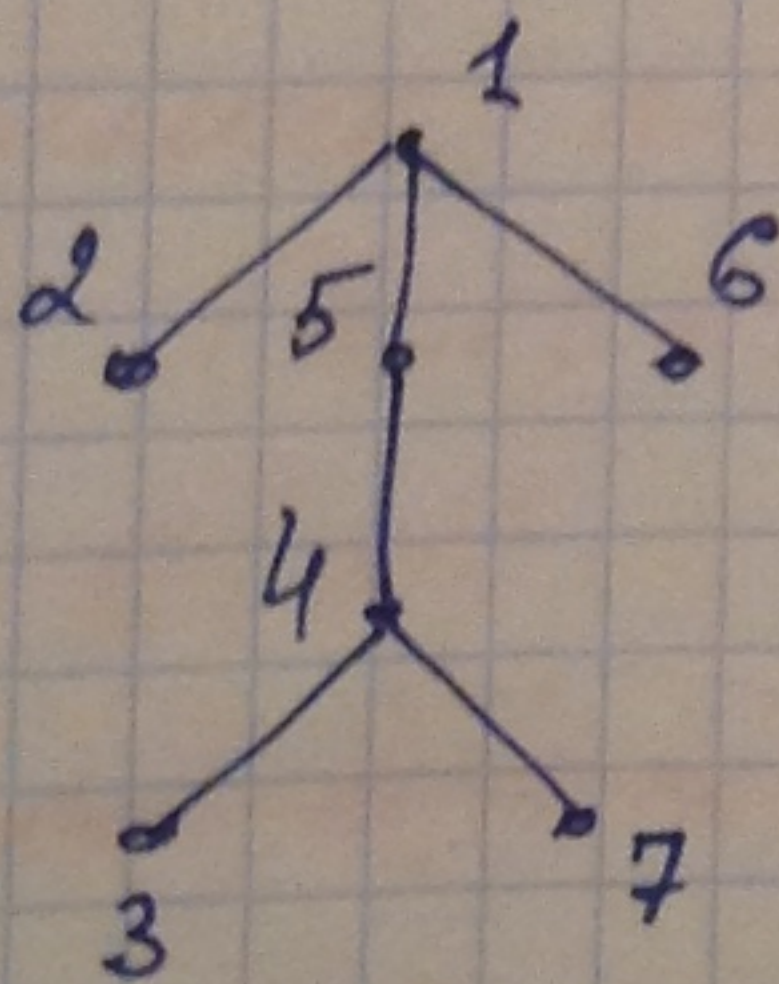
$L(6) = \boxed{1, 5}$

$L(3) = \boxed{4, 7}$

$L(7) = \boxed{3, 4}$

$L(4) = \boxed{3, 5, 7}$

$r = 1$ $Q = \boxed{1, 2, 5, 6, 4, 3, 7}$



Теор. 2. $[G]$ - связный простой (n, m) -граф

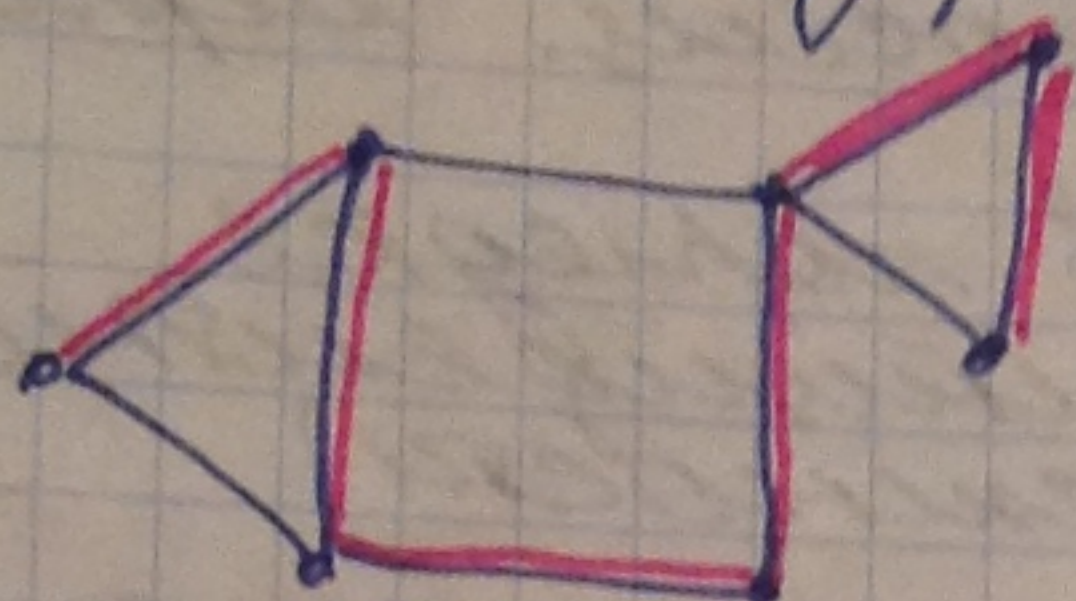
Тогда 1) поиск в ширину посещает
 $\forall$ верш. в точности 1 раз

2) граф $T = (V, ET)$ - дерево

3) сложность поиска в ширину $O(n+m)$

Остовное дерево минимального веса

$G(V, E)$ - простой связный граф
Дерево T , содержащее все вершины графа, ребрами которого являются ребра графа G , назыв. остовным деревом графа
Остовн. дерево: каркас, остов



$$G = (V, E)$$

$$c: E \rightarrow \mathbb{R}$$

$c(e)$ - вес ребра $e \in E$

$$C(G) = \sum_{e \in E} c(e)$$

$T = (V, E_T)$ - остовн. дерево графа G

$$C(T) = \sum_{e \in E_T} c(e)$$

Задача. Среди всех остовных деревьев графа найти остовное дерево min веса

$$|V_1| \geq |V_2| \Rightarrow |V_1| \leq |V| - 1 = n - 1$$

$$|V_2| \leq |V| / 2$$

name [v] сокр., если $v \in V_1$,
иначе, если $v \in V_2$

$$r(v) \leq \log(n-1) \leq \log n$$

$$\sum_{v \in V} r(v) \leq (n-1) \log n \leq n \log n$$

то, что ушло на перемешивание

$$O(m \log m + m + n \log n) \ominus, \quad n-1 \leq m \leq n^2$$

$$\ominus O(m \log n) - \text{сложность алгоритма}$$

Алгоритм Крускала

$$G = \langle V, E, c \rangle$$

1. Выбираем e , min веса; $T_1: V_1 = \{a, b\};$
 $E_1 = \{e\}$

2. Если T_i построено, $i < n-1$, то среди ребер, инцидентных вершинам T_i , выбрать ребро min веса, не образующего цикла

$\Rightarrow T_{n-1}$ - дерево min веса (остовное дерево)

сложность алг.:

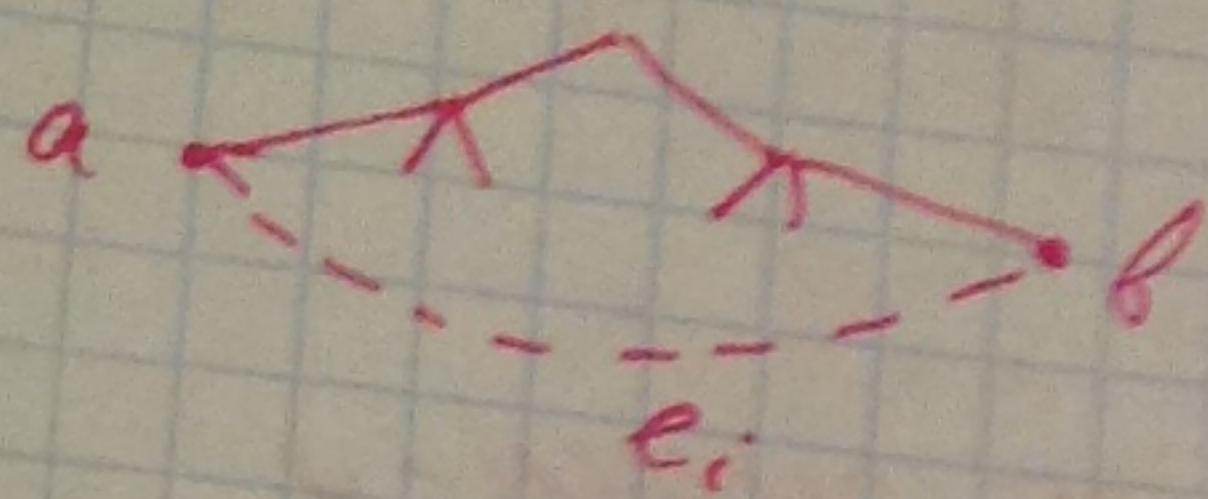
$$(n-1) + (n-2) + \dots + 1 = \frac{n(n-1)}{2}$$

$$O(n^2)$$

~~Решение Крускала~~

6 T - дерево min веса, имеющее max число ребер, отличных
 $\exists e_i \in T_{n-1}, e_i \notin T, e_i$ имеет min номер среди
 всех таких ребер.
 $(e_1, e_2, \dots, e_{i-1}) \in T$

$e_i = (a, b)$ в дереве T $\exists$ путь из $\underline{a}$ в $\underline{b}$



Добавим ребро e_i - получим цикл
 e - ребро цикла $e \notin T_{n-1}$

$$T' = T + e_i - e$$

$$c(T') \geq c(T)$$

$$c(T') = c(T) + c(e_i) - c(e)$$

$$c(e_i) - c(e) \geq 0$$

Кредит, это $c(e_i) > c(e)$, тогда при постр. T_i к графу
 добавимось от ребро e

На сортировку идет массив

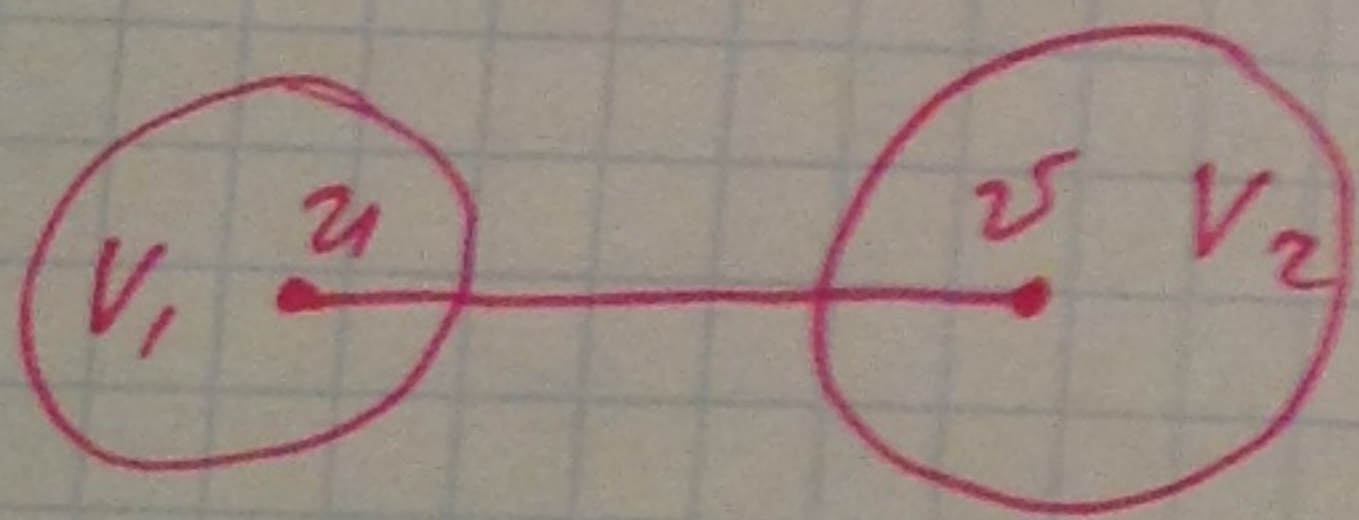
name - имя компоненты связности, которой
 принадлежит верш. v

name[v]

массив size - кол-во вершин в комп. связи

$$T_1 = \langle E_1, V_1 \rangle \quad \text{и} \quad \exists e = (u, v) \quad u \in V_1$$

$$T_2 = \langle E_2, V_2 \rangle \quad v \in V_2$$



$$|V_1| = |V_2|$$

$r(v), v \in V$ - кол-во измен.

name[v] при работе алг

Лемма: $v \in V, G = \langle V, E, c \rangle$

$$h(v) \leq \log |V| = \log n$$

$$|V| = 1 = n \quad \text{-- кер-во}$$

Переход от T_{n-2} к T_{n-1}

$$V = V_1 \cup V_2, \quad V_1 \cap V_2 = \emptyset$$

7

Поиск кратчайших путей и расстояний в ориентированных графах

$G = \langle V, E, c \rangle$ - ориент. граф

$c: E \rightarrow R$, $c(u, v)$ - вес ребра

$P = \langle u_0, u_1, \dots, u_n \rangle$ $u_i \in V$

$l(P) = \sum_{i=1}^n c(u_{i-1}, u_i)$ - длина пути P

$d(a, b)$ - длина кратчайшего пути
 $a, b \in V$ из a в b - расст.

1. Граф без циклов отрицат. длины $O(n^3)$
2. Граф с неотр. весами ребер $O(n^2)$
3. Граф без циклов $O(n^2)$

22.03.13

1. $S := \{v_0\}$

2. $D[v_0] = 0;$

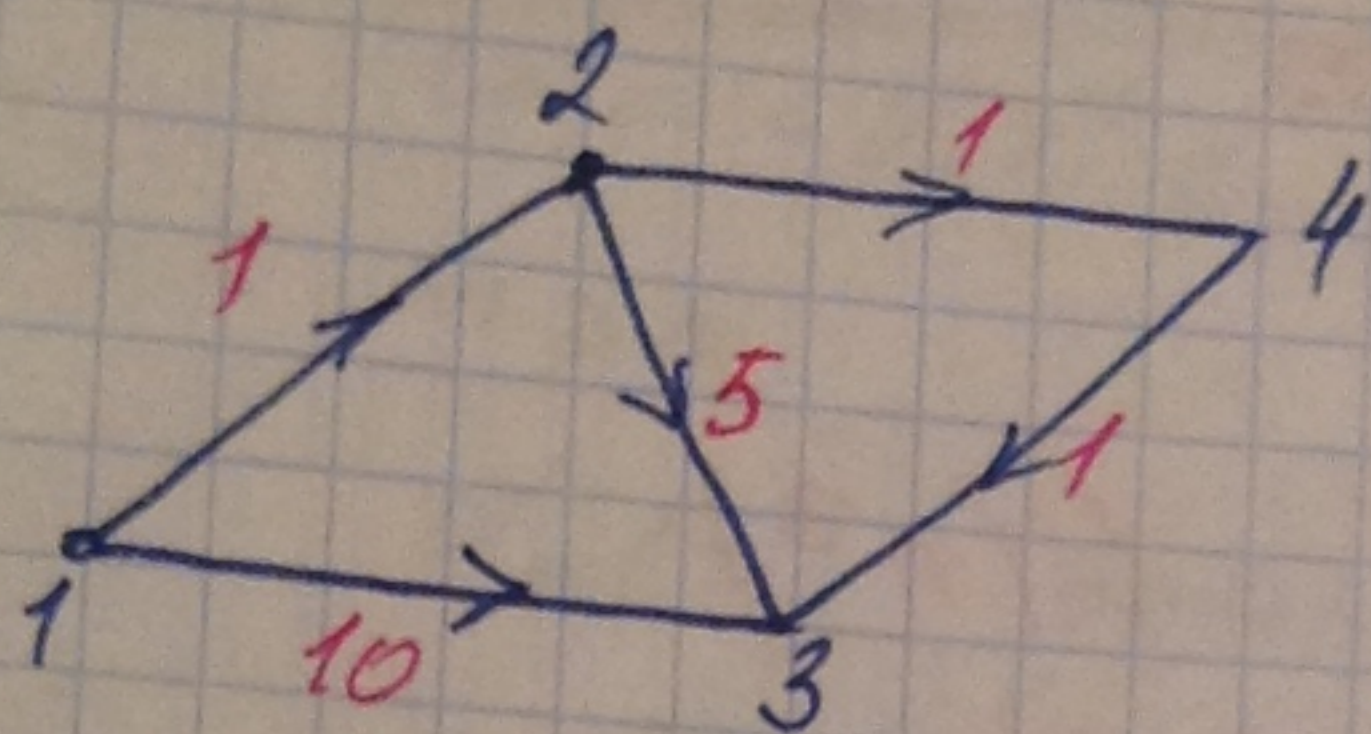
$v \in S: D[v] := c(v_0, v), \text{ Prev}[v] := v_0$

3. Пока $V \neq S$: выбрать $u \notin S$ так, что $D(u) = \min_{v \notin S} D[v]$

4. $S := S \cup \{u\}$

5. $v \notin S: D[v] = \min \{D[v]; \underbrace{D[u] + c(u, v)}_{*}\}$

если $\min - *$, то $\text{Prev}[v] = u$



$v_0 = 1$

S	D[1]	D[2]	D[3]	D[4]
1	0	1 ^{min}	10	∞
1, 2	0	1	6	2 ^{min}
1, 2, 4	0	1	3	2
1, 2, 4, 3	0	1	3	2

получили кратч. расст. от v_0 до всех остальных

$v \in S \Rightarrow D[v]$ есть расстояние от v_0 до v

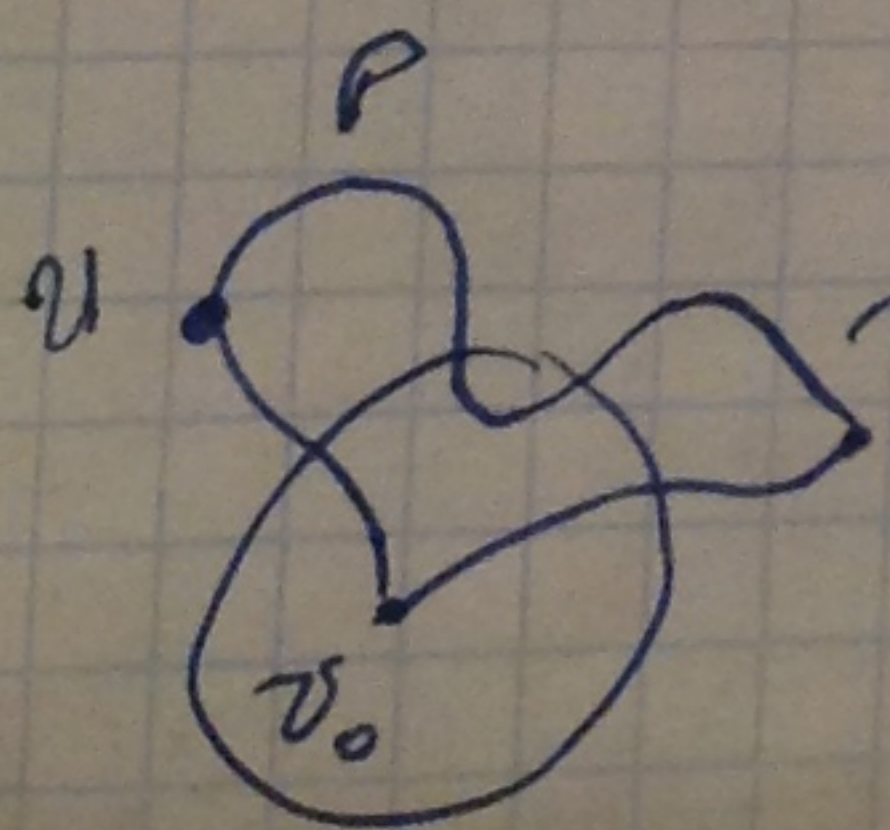
кратч. (v_0, v) -путь целиком лежит в S

$v \notin S \Rightarrow D[v]$ - длина кратч. пути из v_0 в v ,
целиком лежащего в S , кроме
последней вершины

Базис. $|S| = 1$

кр. путь из v_0 в v целиком лежит в S

и $D[v]$ есть длина кр. пути из v_0 в v ,
целиком лежащего в S , кроме
послед. вершины



$D[u] = \min_{v \in S} D[v]$

$D[u]$ - более коротк. путь из v_0 в u

и - первая на пути P вершина, не примарч. 39
 $D[v] \leq D[u] \rightarrow$ против с вооброот, и
 Сложность

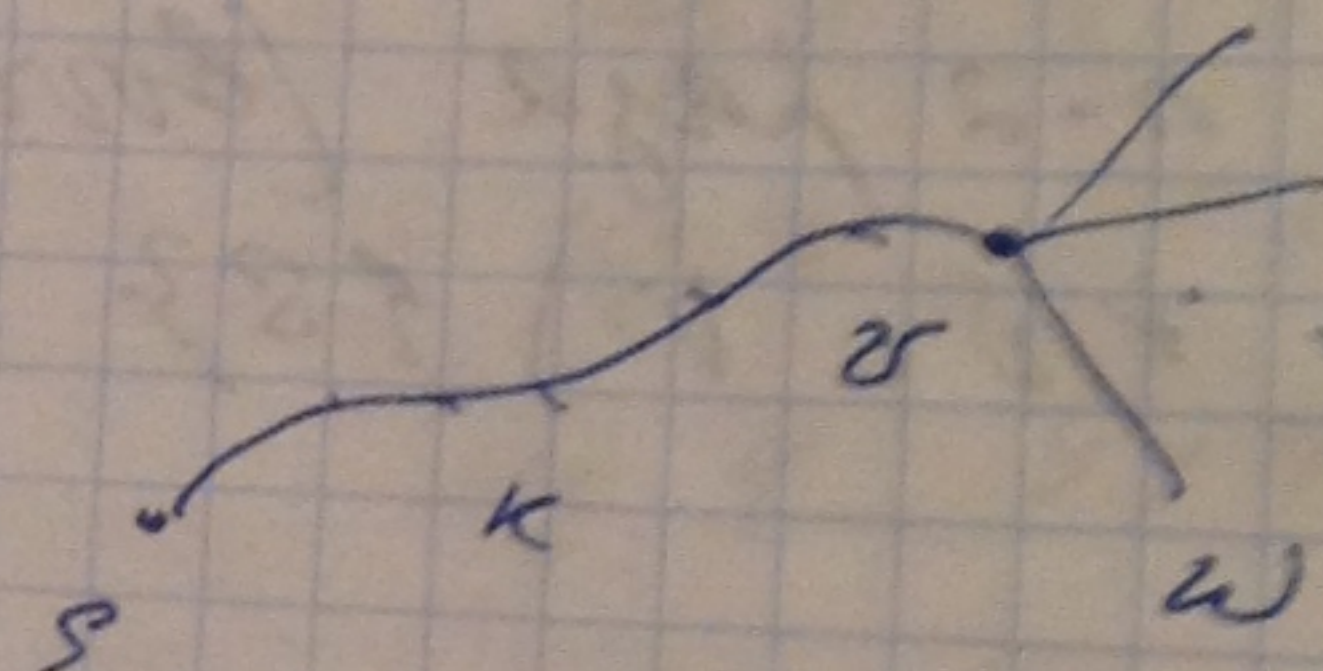
$S \neq v$ вон $(n-1)$ раз
 в числе $O(n)$ макс. min и пересел $D[v]$
 $O(n^2)$

Сеть без чисел отрицат. длинот
 Алгоритм Форда - Беллмана !

$G = \langle V, E, c \rangle$ - сеть, s - источник
 $m \geq n$

$d_k(v)$ - длина кратк. пути из исх. s в верш. v ,
 состоит не более k ребер
 $d_1(v) \geq d_2(v) \geq \dots \geq d_{n-1}(v)$

$d_1(v), d_2(v), \dots, d_{n-1}(v)$
 $d_1(v) = c(s, v), v \in V$



$d_k(v)$ - бучисленна. Найди $d_{k+1}(v)$
 $d_{k+1}(v) = \min\{d_k(v), d_k(w) + c(w, v) \mid w \in V\}$
 $D[v]$

$D[v] = d_1[v] = c(s, v), v \in V$
 $D[v] = \min\{D[v], D[w] + c(w, v) \mid w \in V\}$
 после первого пересел

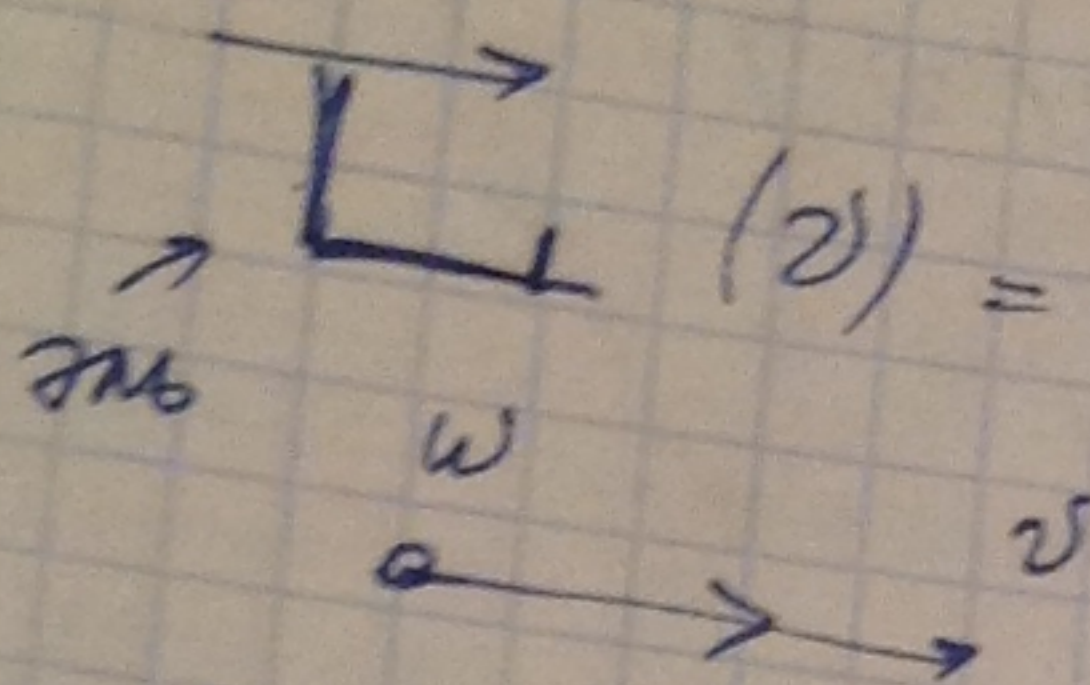
$D[v] \leq d_2(v)$

пусть пересел наг. с верш. v_1 . Тогда $D[v_1] = d_2(v_1)$
 пусть еще верш. v_2
 $D[v_2] = D[v_1] + c(v_1, v_2) \Rightarrow D[v_2] \leq d_2(v_2)$

10 После $n-1$ шагов:

$$D[v] = d_{n-1}(v)$$

Вход $G = \langle V, E, c \rangle$ - сеть
 s - источник



$\vec{L}(v) = \{w \in V \mid \langle w, v \rangle \in E\}$ - множество ребер, входящих в вершину v

Результат: расст. $D[v]$ от s до всех $v \in V$
 $Prev[v]$ - верш. предш. v

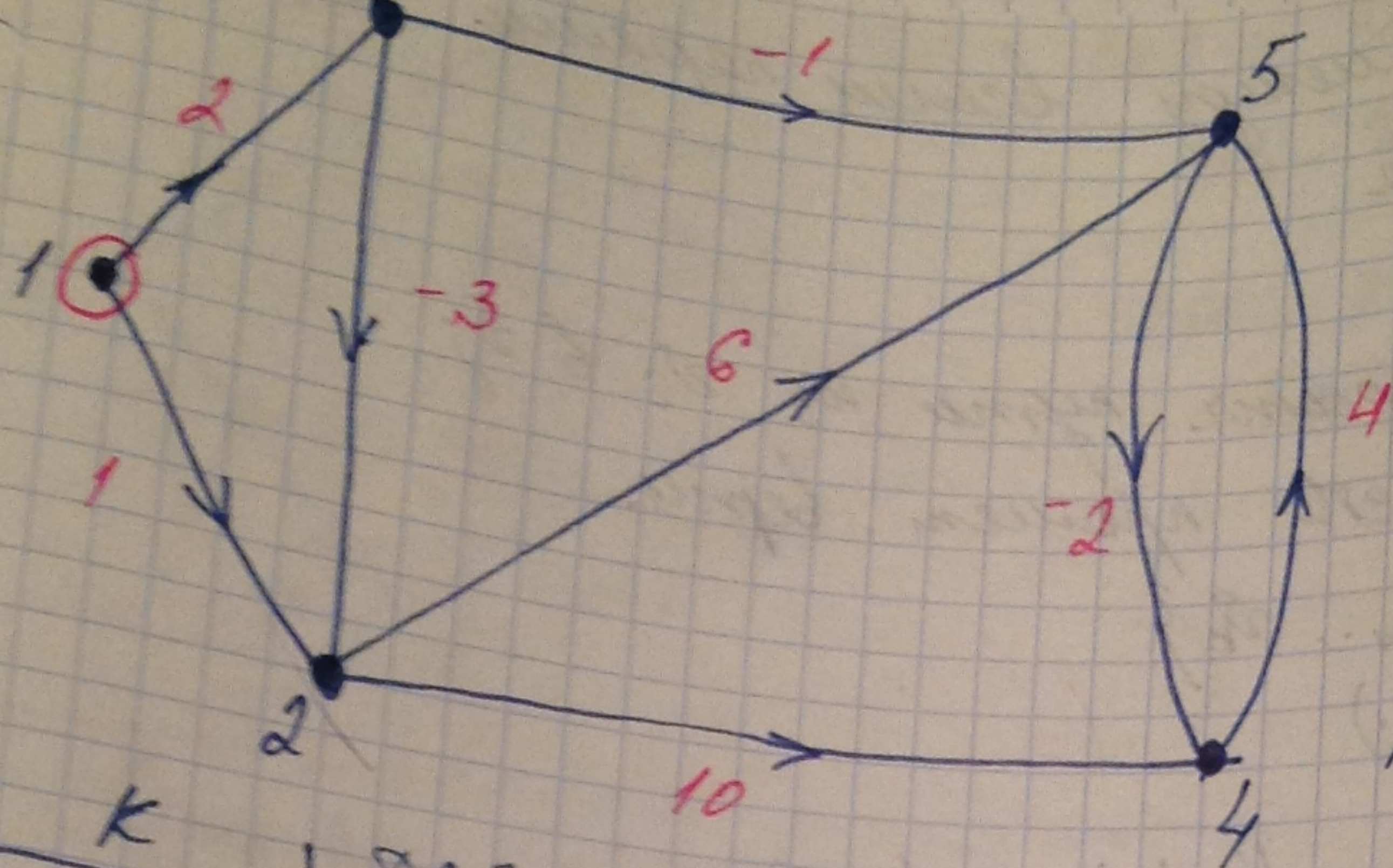
1. $D[s] = 0$; $Prev[s] = \emptyset$

2. $\forall v \in V \setminus \{s\}$: $D[v] = c(s, v)$; $Prev[v] = s$;

3. $n-2$ раза [вычислять] повторить:

$\forall v \in V \setminus \{s\}$: $D[v] = \min \{D[v], D[w] + c(w, v) \mid w \in \vec{L}(v)\}$

29.03.13. 3



$S=1$
 $\vec{L}(2) = \{1, 3\}$
 $\vec{L}(3) = \{1\}$
 $\vec{L}(4) = \{2, 5\}$
 $\vec{L}(5) = \{2, 3, 4\}$

n-2 пересчётов
пересчёт всего списка

k	D[1]	D[2]	D[3]	D[4]	D[5]
0	0	1	2	∞	∞
1	0	-1	2	9	1
2	0	-1	2	-1	1
3	0	-1	2	-1	1

$n-2=3$

б 4: уз 2 9

уз 5 $-2+1=-1$

б 5:

$$D[v] = d_{n-1}(v)$$

$$D[v] = d_1(v)$$

k-е итер.: $D[v] \leq d_k(v)$

$$D[v] \leq d_{k+1}(v)$$

$$D[v] \leq D[w] + e(w, v) \quad \forall w \in \vec{L}(v)$$

$$D[v] \leq d_k(w) + e(w, v) \quad \forall w \in \vec{L}(v)$$

б 2: уз 1 1
уз 3 -1 ^{min}

б 3: 2

б 4: ~~уз 2~~ $2 - 3 + 10 = 9$

б 5: уз 2 $-1 + 6 = 5$

уз 3 $-1 + 2 = 1$

уз 4 $9 + 4 = 13$

12 Нахождение раскр-о между всеми парами вершин
Алгоритм Флойда

$G = \langle V, E, c \rangle$ - сеть

$d_k(v)$ - длина кратч пути из v_i в v_j ,
сод. в качестве промежу. верш.
 $\{v_1, v_2, \dots, v_k\}$

$$d_0(i, j) = c(v_i, v_j)$$

$d_k(i, j)$ вычислено $\forall i, j$ и $k \geq 0$

$$d_{k+1}(i, j) = \min \{ d_k(i, j), d_k(i, k+1) + d_k(k+1, j) \}$$

P - кратч. (v_i, v_j) - путь с пром. верш. $\{v_1, v_2, \dots, v_{k+1}\}$

1. $v_{k+1} \notin P$ $d_{k+1}(i, j) = d_k(i, j)$

2. $v_{k+1} \in P$

Разобьем путь P на 2 пути:

$P_1 : (v_i, v_{k+1})$ -

$P_2 : (v_{k+1}, v_j)$

$$D_0 = (d_0(i, j))$$

$\vdots$

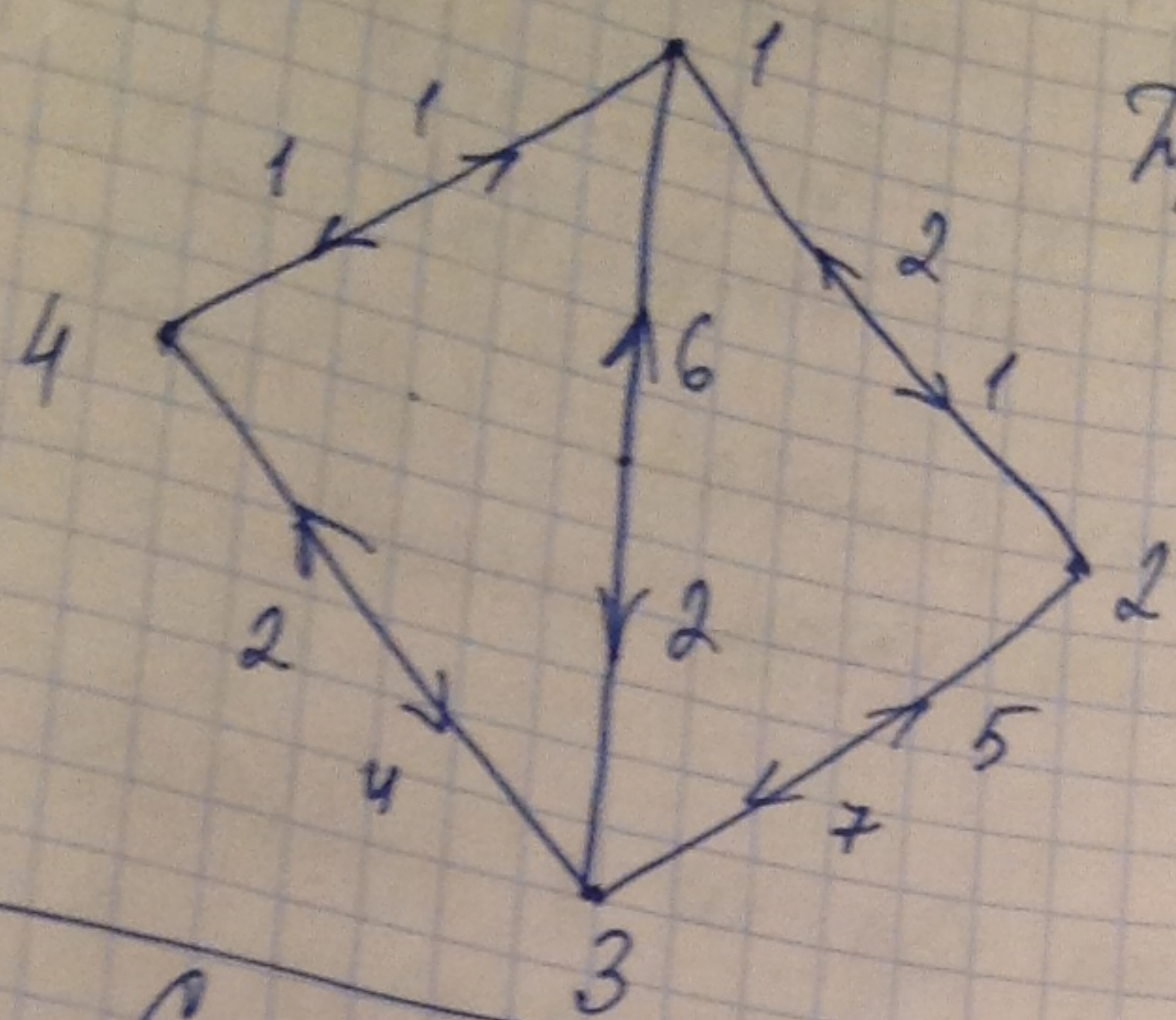
$$D_k = (d_k(i, j) \mid 1 \leq i, j \leq n)$$

$Prev[v_i, v_j]$ верш., предш. v_j на кратч. пути (v_i, v_j) по D_k

$$d_k(i, i) = 0, \text{ иначе все } \infty$$

$$d_{k+1}(i, k) = d_k(i, k)$$

$$d_{k+1}(k, i) = d_k(i, k)$$



Д/з: найти $D_0 \dots D_4$

D_0 - граф
только эд-ти D_k пути при
вошли. D_{k+1}

Сеть без циклов

1. В ор. графе без циклов $\exists$ верш, степен. исхода
ком. = 0.

степен. исхода v :

$$\text{DegOut}(v) = \rho(v) = |\{w \mid \langle v, w \rangle \in E\}|$$

$G = \langle V, E \rangle$ - ор. граф

$v_1 \in V$

$$\rho(v_1) \neq 0 \Rightarrow \exists v_2 \in V: \langle v_1, v_2 \rangle \in E$$

$v_2 \in V$

$$\rho(v_2) \neq 0 \Rightarrow \exists v_3 \in V: \langle v_2, v_3 \rangle \in E$$

цикл $\uparrow$ законч. v_k : $\text{DegOut}(v_k) = 0$.

1.2. В ор. графе без циклов вершины можно
занумеровать так, что $\forall$ ребро идёт из верш.
с меньш. ном. в верш. с больш. ном.

Такой ор. граф имеет топологич. отсортир.

Метод критического пути

$W = \{v_1, \dots, v_n\}$ - проект W сост. в вып. работ $v_1 \dots v_n$
 $t_m(v_k)$ - длительность вып. работ. v_k

$Prev(v_k)$ - работы, непосред. предшествов. работе v_k :
 только после вып.-я работ $Prev(v_k)$ можно
 сделать v_k

Добавим в W две фикт. работы:

S - начало проекта

t - окончание пр.

$Prev(S) = \emptyset$

$Prev(t) = \emptyset \Rightarrow Prev(v) = \{S\}$

$W \Rightarrow G = (W, E, c)$

1. $V = W \cup \{S, t\}$

2. $E = \{ \langle v, w \rangle \mid v \in Prev(w) \}$ - отнош. непосред. предш. работ
 предш. задан ребра

3. $c(v, w) = t_m(v)$

Сеть G - сетевой график вып. работ по пр. W
 циклов отсутств.

Все верш. можно отсортировать по

n	Наим. работы	Предш. раб.	Длит-ть
1.	Закл. фундамента	$\emptyset$	4
2	Воздвиг. коробки	1	8
3.	Электромонтаж	2	2
4	Сан.-тех. оборуд-е	2	3
5	Настил. крыши	2	4
6.	Отдел. работ	3, 4	5
7	Благоустр. террит.	5	2

1. Постр. сетев. график
 в виде ор. графа
 с весами ребер
 и дополн. верш.
 S и t

2/3

$e_{fin}(v)$ - наиб. ранний возм. срок оконч. работы
с начала v

$e_{вер}(v)$

$$e_{вер}(s) = e_{fin}(1) = 0$$

T - максимальный срок окончания работ по проекту.

$$0 = s$$

$$n+1 = t$$

$$e_{fin}(n+1) \leq T$$

$e_{fin}(v)$ - наиб. поздний возможный срок окончания
работы v , т.е. такой,
который не увеличивает время окончания
проекта T

$e_{вер}(v)$ - наиб. поздний допуст. срок начала работ v

$$e_{вер}(v) = e_{fin}(n+1) = T$$

$$e_{fin}(v) = \min \{ e_{вер}(w) \mid v \in \text{prev}(w) \}$$

$$e_{вер}(v) = e_{fin}(v) = t_m(v)$$

$$v \in V: e_{fin}(v) = T$$

$$k: n+1, (-1), 0: \\ \text{от с началом до}$$

$$e_{вер}[k] = e_{fin}[k] - t_m[k]$$

$$i \in \text{prev}[k]$$

$$e_{fin}[i] = \min \{ e_{fin}[i], e_{вер}[k] \};$$

Полный резерв работ v :

$$\text{reserv}(v) = e_{вер}(v) - o_{вер}(v) - \text{насе. задержка в вып. работ } v, \text{ не влияющая на время оконч. проекта } T$$

$$f(v) = \sum_{u \in V^-} f(u, v); \quad f(v) = \sum_{u \in V^+} f(v, u)$$

Работы, имеющие нулевой резерв времени — критические работы

Макс. (s, t)- путь, проходящий через критич. работы — критический путь

Пример из 9/3

	Работ	e beg	e fin	l beg	l fin	резерв
s	0	0	0	0	0	0
	1					
	2					
	3					
	4					
	5					
	6					
	7					
t	8					

0 - s
8 - t

отметить критич. работы и критич. путь

2/3

Задача о макс потоке в сети

сеть: $G = \langle V, E, c \rangle$ (ор. граф)

$s \in V$: - источник единств. верш. $\vec{L}(s) = \emptyset$ (←)

$t \in V$: - сток $\vec{L}(t) = \emptyset$ (→)

$c(e) > 0$ пропускная способ-ть ребра $e \in E$

$v_+ = \vec{L}(v)$ - лев. во вершин, к кот. идут ребра v

$v_- = \vec{L}(v)$ - лев. верш., ребра кот. выходят в v

-		+
то, что выходит		то, что входит

$f: E \rightarrow R$ - поток в сети G, если:

1) $0 \leq f(e) \leq c(e) \quad \forall e \in E$ (усл. ор-о по пропускн. способ-ти)

2) $f(v_-) = f(v_+) \quad \forall v \in V \setminus \{s, t\}$

$f(v_-) = \sum_{u \in v_-} f(u, v); \quad f(v_+) = \sum_{u \in v_+} f(v, u)$ (усл. сохр-н потока в вершине)

сколько
вошли
столько
вышло

18 $\|f\| = f(s^+)$ - величина потока (сколько вытекает из верш. s)

Задача о max потоке.

Для заданной сети G найти поток max величины

(V_s, V_t) - (s, t) -разрез, если:

1) $s \in V_s, t \in V_t$

2) $V_s \cup V_t = V$

3) $V_s \cap V_t = \emptyset$

19.04.13.

$$\|f\| = \sum_{v \in V_s} f(v^+) - \sum_{v \in V_s} f(v^-) \quad (2)$$

$$e = (v, w) \in E, \quad v, w \in V_s$$

$f(v, w)$ - входит в $f(v^+)$ как часть вых. потока в v
и в $f(w^-)$ как часть вх. в w потока

$f(v, w)$ - взаимно уничтож. в (2)

(V_s, V_t) - разрез:

$$c(V_s, V_t) = \sum_{e \in E(V_s \rightarrow V_t)} c(e) \quad \text{— пропускная способность разреза}$$

$$0 \leq f(V_s \rightarrow V_t) \leq c(V_s, V_t)$$

Сл. 1 * поток и * разрез

$$\|f\| \leq c(V_s, V_t)$$

Разрез (V_s^*, V_t^*) - min, если

$$c(V_s^*, V_t^*) \leq c(V_s, V_t) \quad \forall V_s, V_t$$

$\exists f$ - max поток, (V_s, V_t) - min разре

$$\|f\|^* \leq \|f\| \leq c(V_s, V_t) \leq c(V_s^*, V_t^*) \Rightarrow \text{все нер-ва} \rightarrow p\text{-ва} \\ (\|f\|^* = c(V_s^*, V_t^*))$$

Цепь из v в w в сети G есть чередующаяся последовательность различных вершин и ребер (без циклов)

$v_0 = v, e_0, v_1, \dots, e_{k-1}, v_k = w$, в которой

$$e_i = \langle v_i, v_{i+1} \rangle \in E, \text{ либо}$$

$$e_i = \langle v_{i+1}, v_i \rangle \in E$$

$e_i = \langle v_i, v_{i+1} \rangle$ - согласованное (прямое) ребро

$e_i = \langle v_{i+1}, v_i \rangle$ - несогласованное (обратное) ребро

P - цепь из v в w $e \in P$

$$\Delta(e) = \begin{cases} c(e) - f(e), & \text{если } e \text{ - прямое ребро} \\ f(e), & \text{если } e \text{ - обратное ребро} \end{cases}$$

$$\delta(P) = \min \{ \Delta(e) \mid e \in P \}$$

(s, t) - цепь P называется увеличивающей потоком f (f -увеличивающей), если

$$\delta P > 0$$

13 (алгор увелич. потока)

$\exists f$ - поток в сети G , P - f -увеличивающая (s, t) цепь. Тогда в сети $\exists$ поток f^* :

$$\|f^*\| = \|f\| + \delta P$$

$$f^*: E \rightarrow R: \begin{cases} f(e) + \delta(P), & \text{если } e \in P \text{ - } e \text{ - прямое ребро} \\ f(e) - \delta(P), & \text{если } e \in P \text{ - } e \text{ - обратн. ребро} \\ f(e), & \text{если } e \notin P \end{cases}$$

20

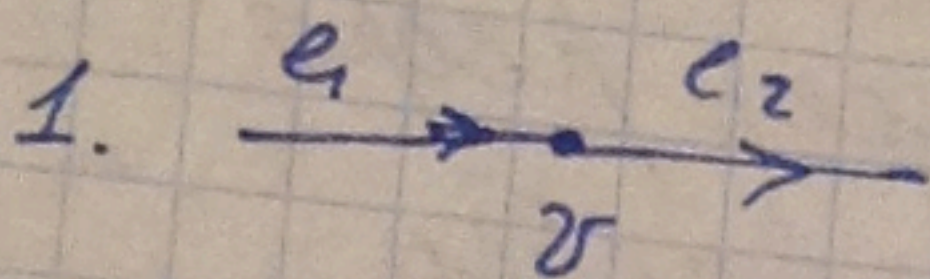
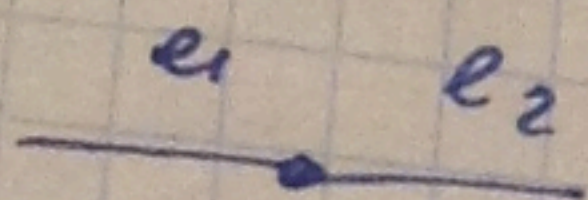
 e - прямое ребро

$$0 \leq f(e) \leq f^*(e) = f(e) + \delta(P) \leq f(e) + c(e) = \\ = f(e) + [c(e) - f(e)] = c(e)$$

 e - обратное ребро

$$f^*(e) \leq c(e)$$

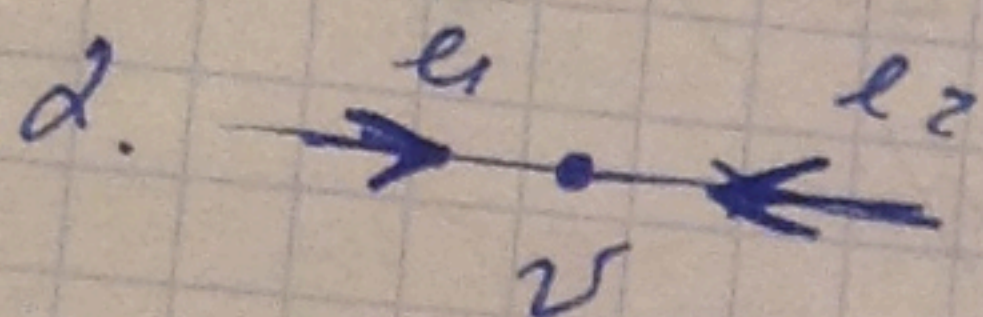
$$f^*(e) = f(e) - \delta(P) \geq f(e) - c(e) = f(e) - f(e) = 0$$

Усл 2 следует проверить только для $v \in P$  e_1, e_2 - прям.

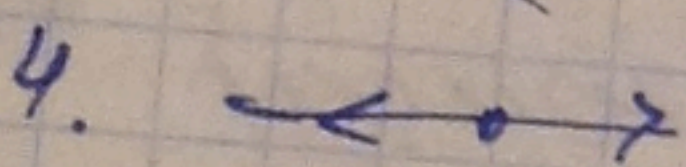
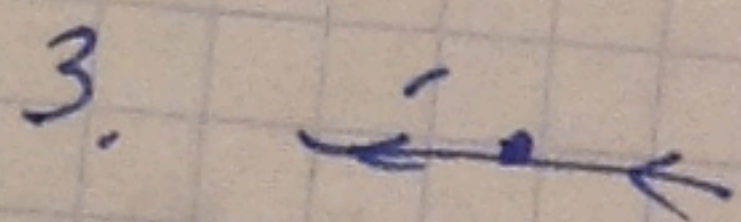
$$f^*(v^-) = f(v^-) + \delta(P)$$

$$f^*(v^+) = f(v^+) + \delta(P)$$

$$f^*(v^-) = f^*(v^+)$$



$$f^*(v^-) = f^*(v^+)$$



3. 4. } по аналогии

Теор. 2 Для потока f в сети G эквив. условия:1) Поток f - max2) Не $\exists$ увеличивающийся f (s, t) - усло3) $\exists$ разрез (V_s, V_t) , где ком. $c(V_s, V_t) = \|f\|$ 1 $\Rightarrow$ 2 - лемма 33 $\Rightarrow$ 1 - лемма 12 $\Rightarrow$ 3

$$V_s = \{v \in V \mid \exists (s, v) \text{ - усло } P: \delta(P) \geq 0\}$$

$f(e) = c(v_s, v_t)$

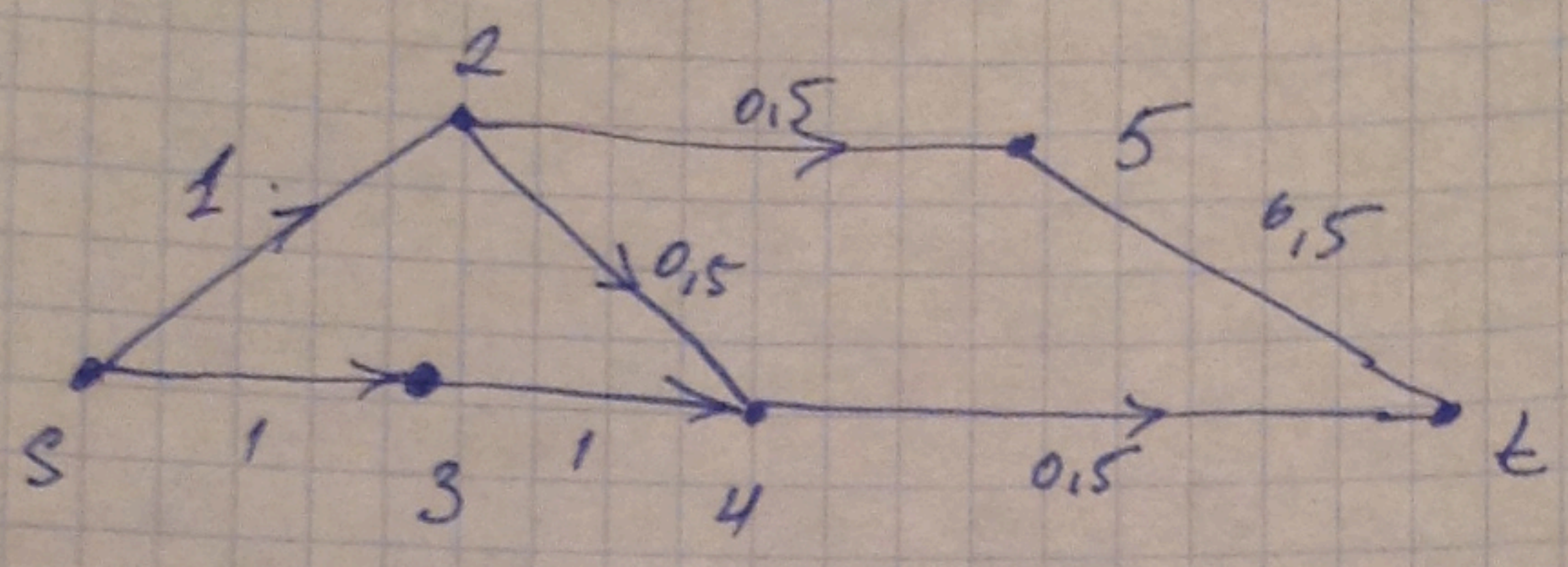
$e = \langle v, w \rangle, e \in E(v_s \rightarrow v_t) \Rightarrow f(e) = c(e)$

$f(e) \leq c(e)$ (s, v) -цель P $(\delta(P) > 0)$

может быть дополн. ребром e и верш. v до (s, w) цель Q $w \in V_s'$

Сл. Величина max потока равна пропускной способ-ти min разреза

2/3

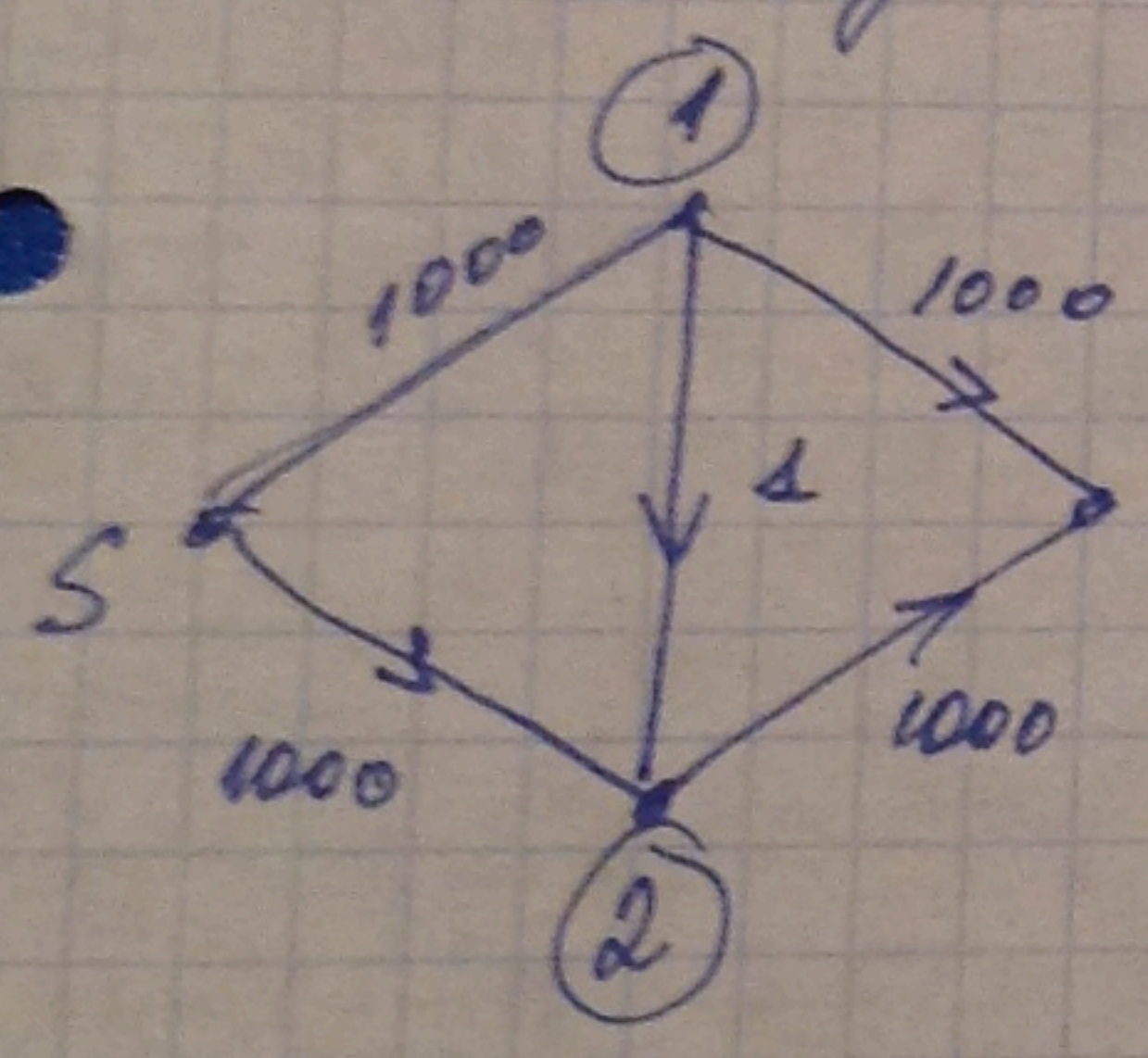


велич. max потока и min разреза

— 0 — e поток

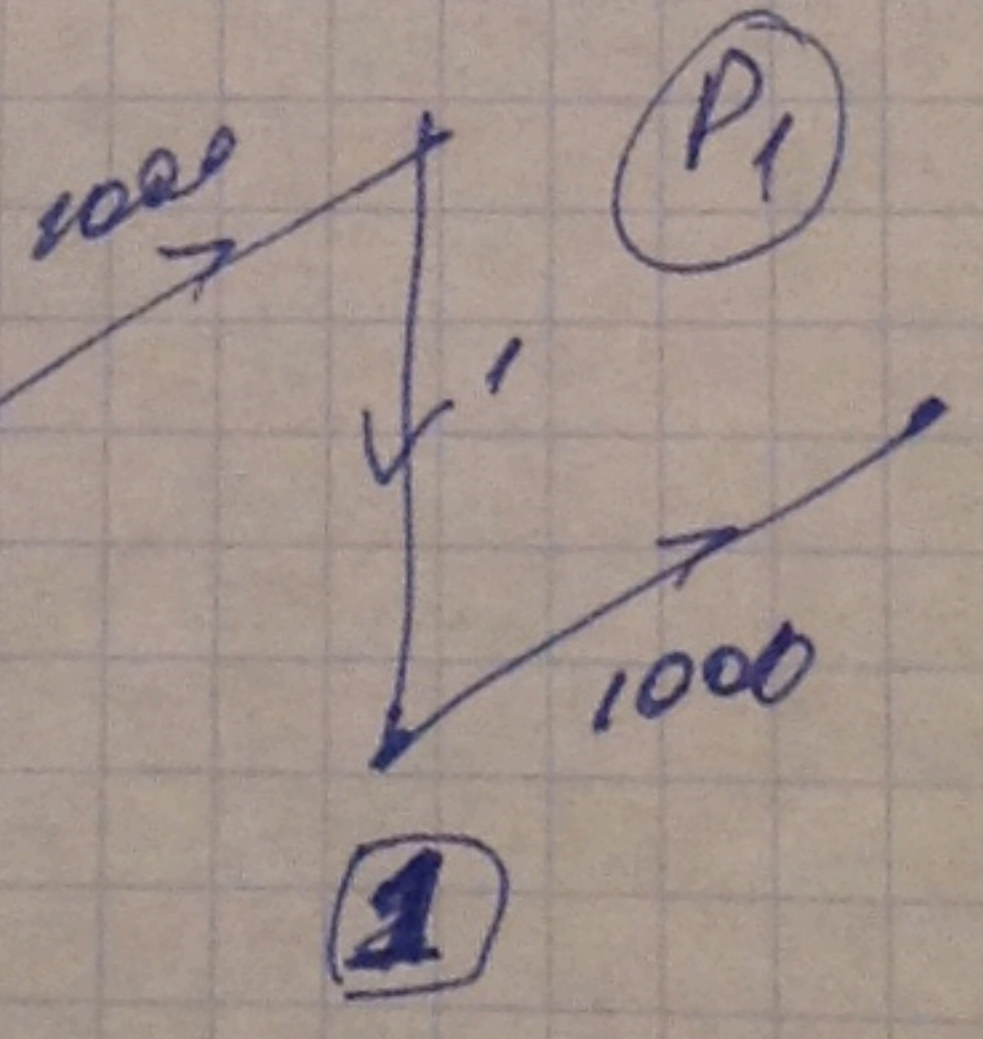
цели { — цель
— увелич

нет увелич. цели $\rightarrow$ max поток

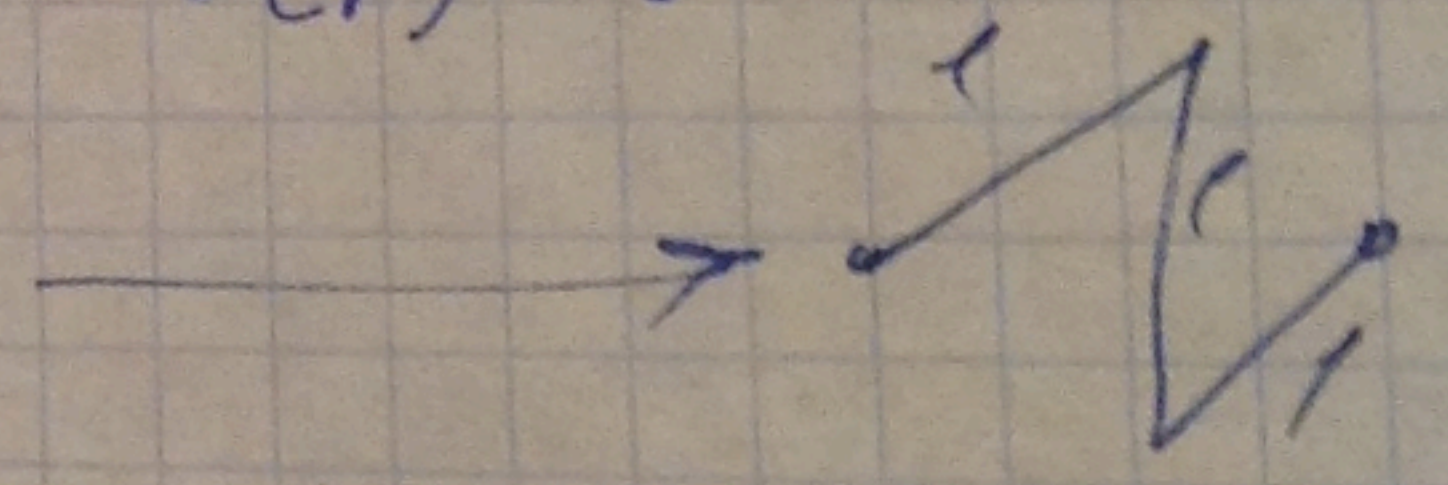


кучев.

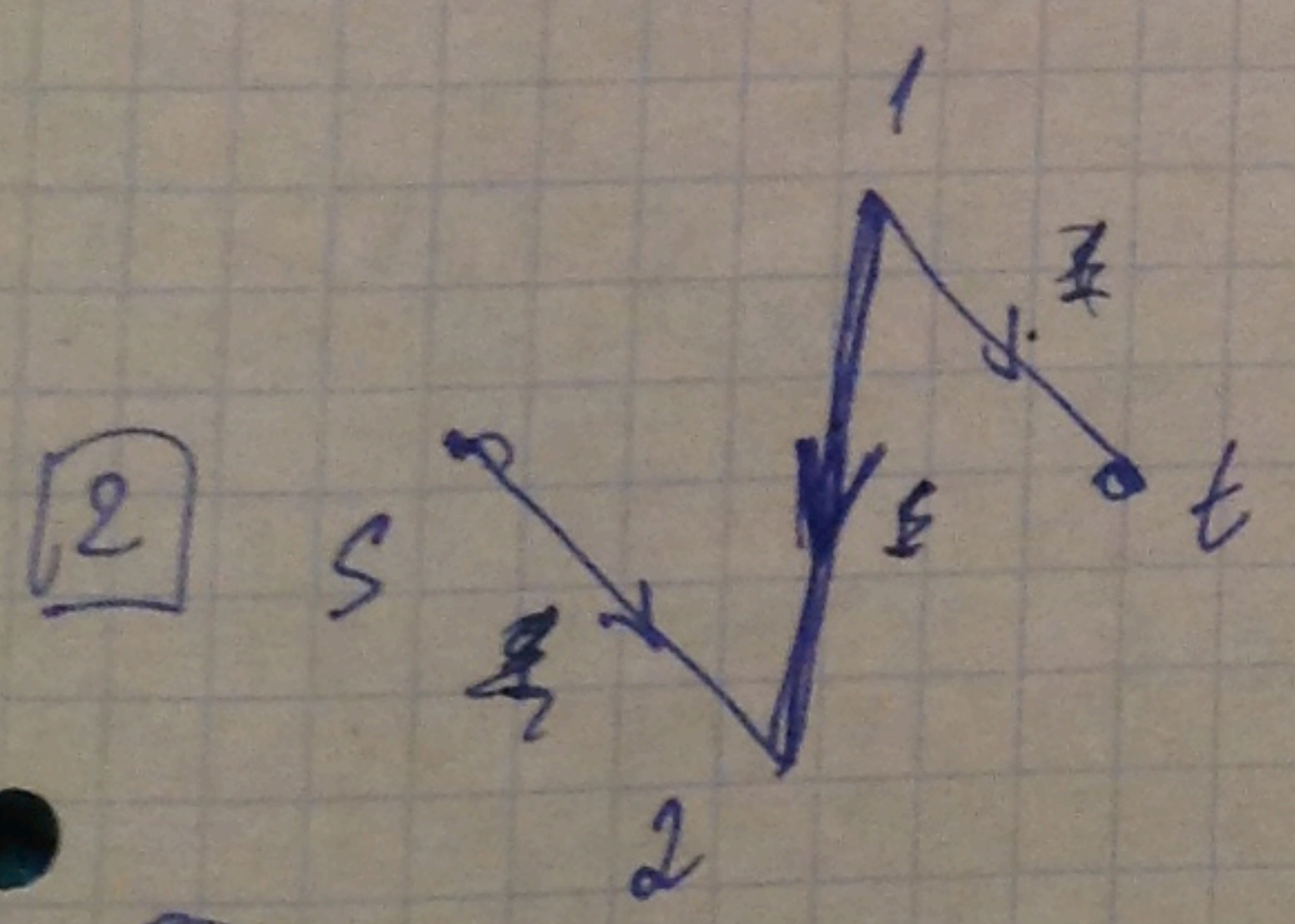
③



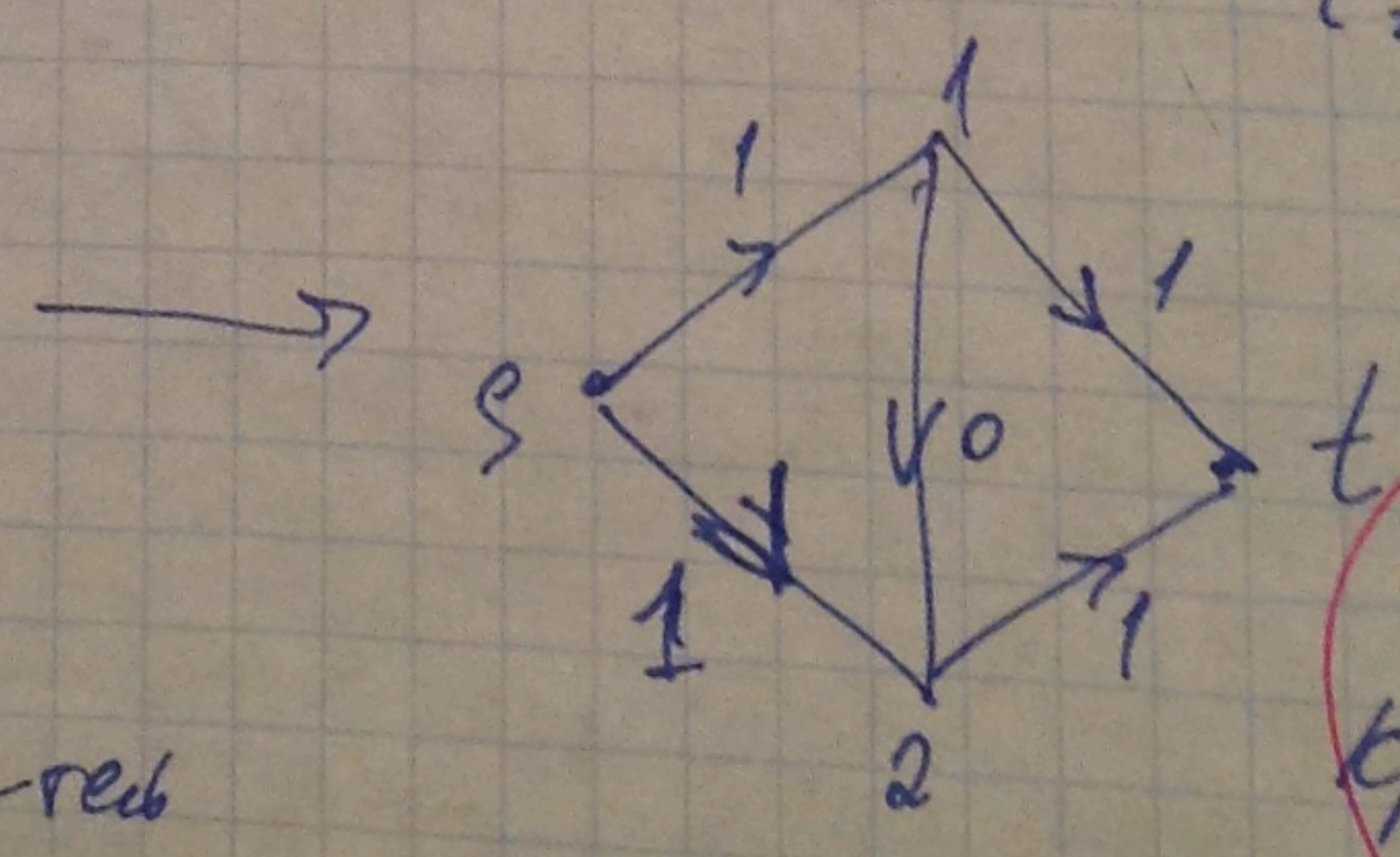
$\delta(P) = 1$



$\delta(P) = \min_{(e) \in P}$



$\delta(P_2)$



$P_2: s \rightarrow 2 \rightarrow t$ δP_2 - min разрез

иначе 2000 просчетов

надо искать кратчайшие пути!