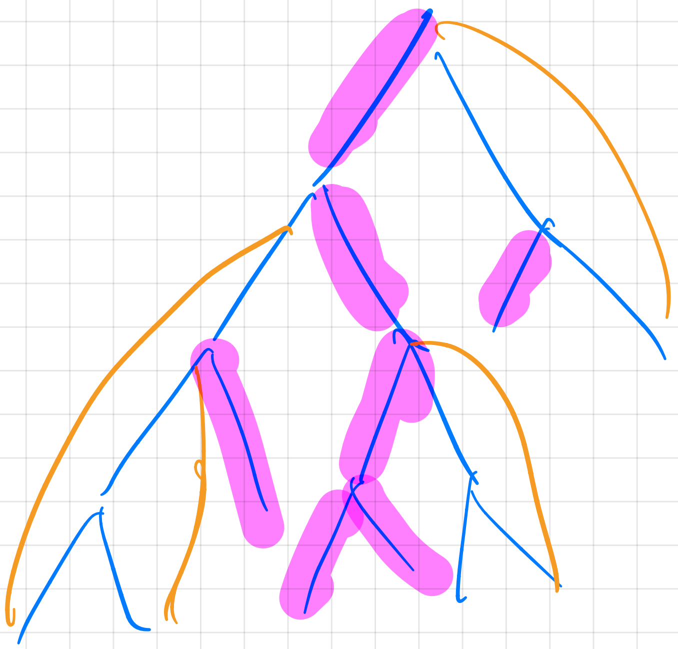
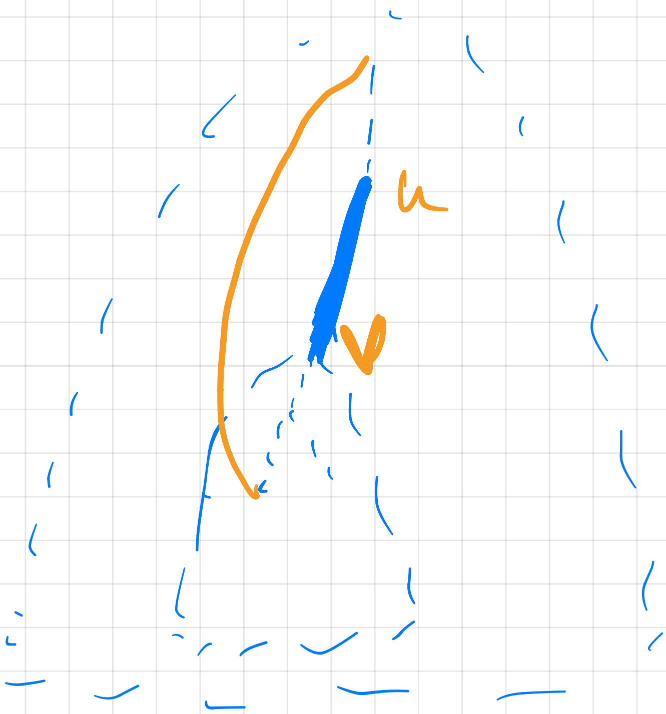
Мост.



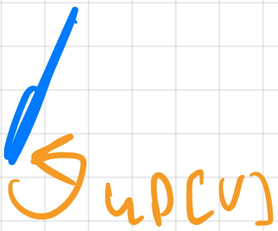
Утв Обратное ребро никогда не мост.

Утв Древесное ребро мост если не покрыто обратным.

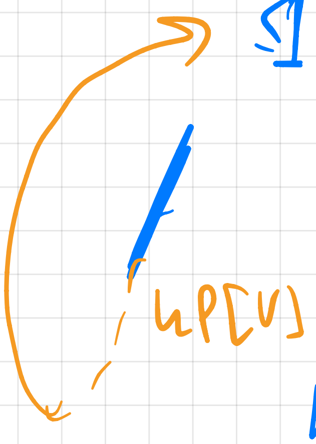


up[v] = самая неглубокая в-на до которой мы можем добраться из поддерева v по $\exists 1$ обр. ребру.

(её глубина)



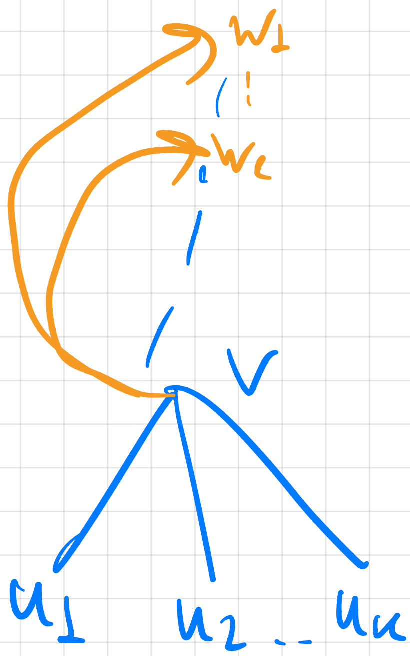
Мост



Не мост

up[v] == depth[v]

up[v] < depth[v].



min

$up[v] \leftarrow depth[v]$

$up[v] \leftarrow up[u_i] \ \forall i$

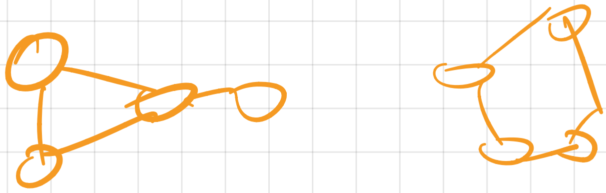
$up[v] \leftarrow depth[w_i] \ \forall i$

```
dfs(v, par) {
    depth[v] = depth[par] + 1
                   или depth[v] = 0
                   если корень.
    up[v] = depth[v]
    for  u ∈ adj[v]:
        if  u == par:
            continue
                            (unused)
        if  u - врем. ребро
            dfs(u)
            up[v] ← up[u]
        else: up[v] ← depth[u]
}
```

Компоненты Сильной Связности

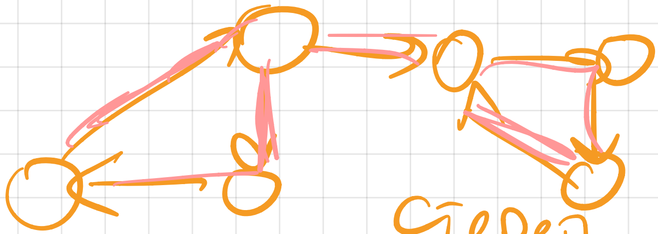
Kosaraju.

Комп. связн.



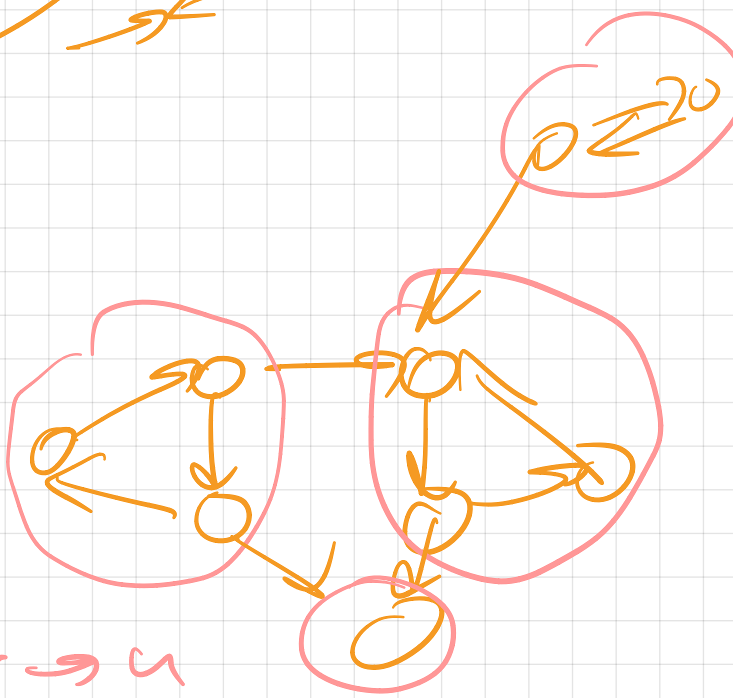
Ор. граф. ???

Комп. Слабой Связности



стереть ориент.

Комп. Сильной Связность



$\forall v, u \in$ группы $\exists\, v \to u$

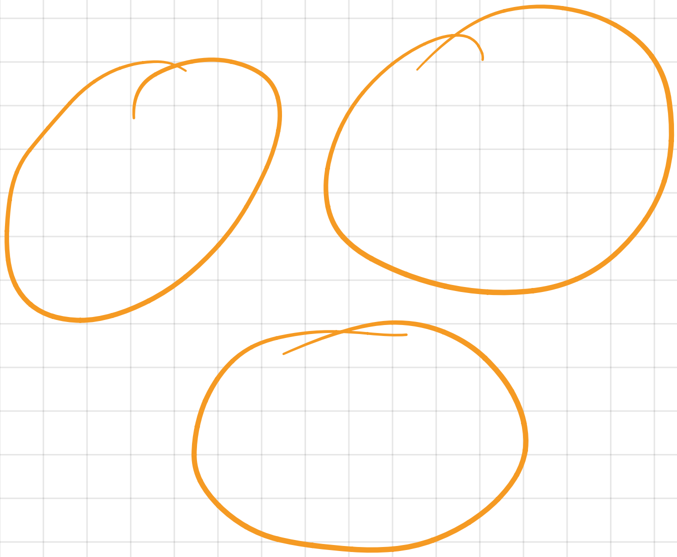
$\forall v, u \quad \exists\, v \to u$ и $u \to v$

Отн. эквив.

$$a \sim a$$

$$a \sim b \Rightarrow b \sim a$$

$$a \sim b \text{ и } b \sim c \Rightarrow a \sim c$$



Утв Пусть G – неор. граф,

и пусть $v \sim u$, если $\exists$ путь $v \to u$.

Тогда $\sim$ – отн. эквив.

Утв 2 Пусть G – ор. граф

и пусть $v \sim u$ если

$\exists$ путь $v \to u$

и $\exists$ путь $u \to v$.

Тогда $\sim$ — отн. экв.

$$v \rightleftarrows u \rightleftarrows w$$

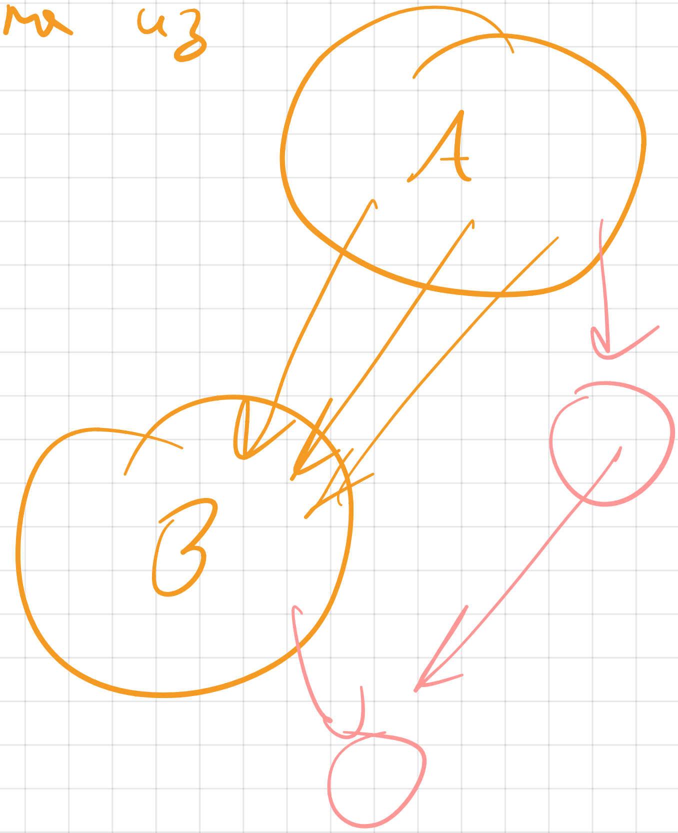
Def $\quad tout[S] = \max\limits_{v \in S} (tout[v])$

S — мн-во в-н.

$tin[S] = \min$

Утв: пусть A и B — к.с.с.

и B достижима из A.



Тогда

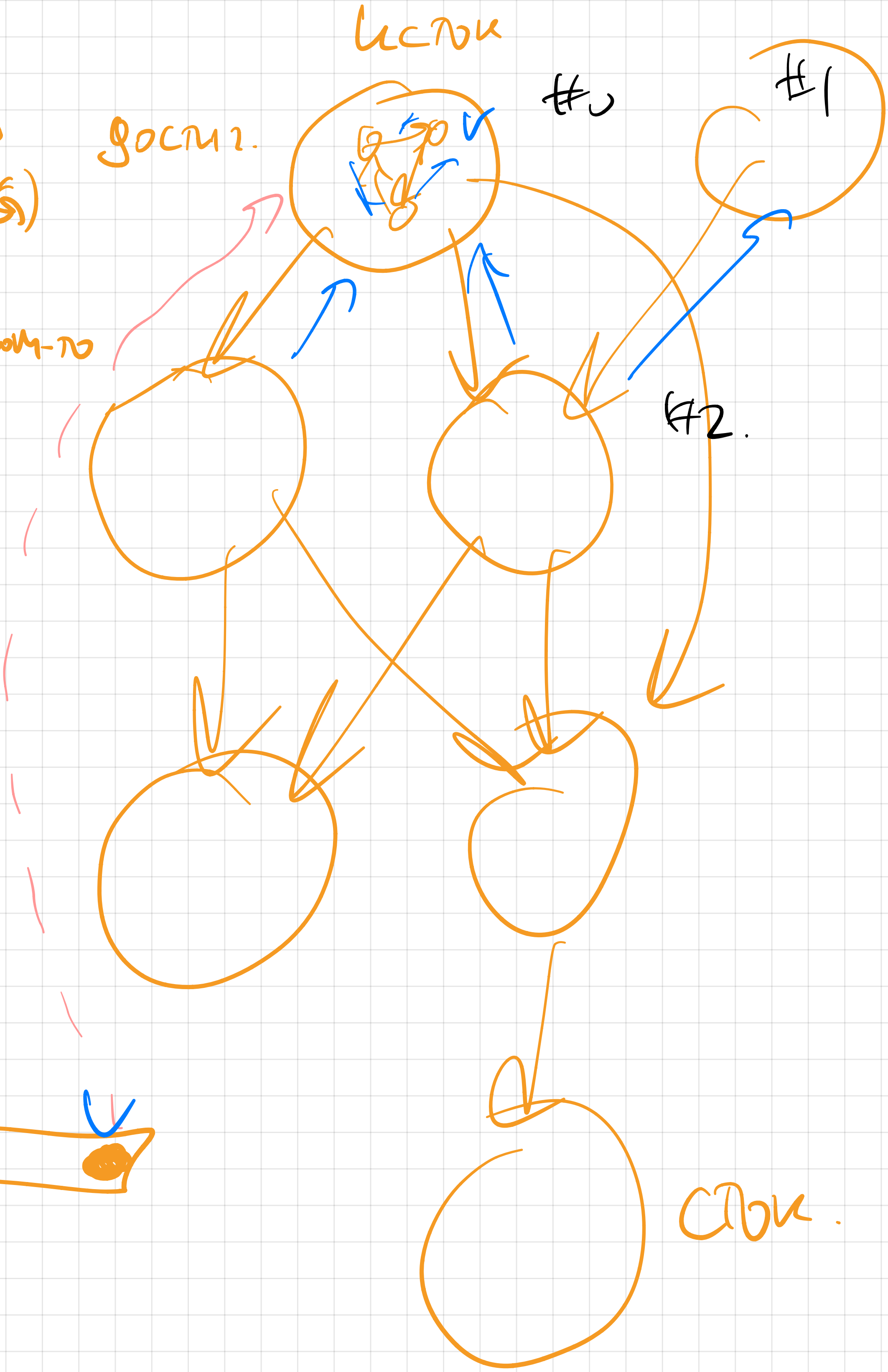
$tout[A] > tout[B]$

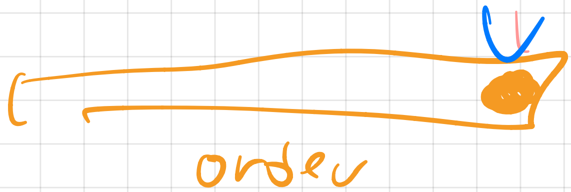
1) tin[A] < tin[B]

2) tin[A] > tin[B]

ЧТБ: tout[C]
C ∈ КСС(G)

Max на компон-те
исток.





Алгоритм

Дано G.

шаг 1: построить order ==
(dfs) = сортировка по tout.

шаг 2: построить G^{rev}

шаг 3:

used = [false . . . . false]

for v in reversed(order)

if !used[v]:

выделить комп. в которой v.

т.е. dfs(used, G^{rev}, v)

$$O(V+E)$$

2-SAT (SATisfyability)

Def SAT: $\underbrace{(x \vee y \vee \neg z)}_{\text{клозы}} \wedge \underbrace{(x)}\wedge\underbrace{(y \vee z)}$

литералы, или, и

$x, \neg x$ = литералы.

Дана ф-ла, проверить $\exists$ ли решение (может быть найти само решение)

Def k-SAT: та же самая задача но известно что в каждом клозе $\le k$ лит.

SAT — NP-полная задача
(скорее всего не решается за poly(n))

3-SAT — NP-полная задача

2-SAT $\in$ P
(решается за полином).

2-SAT

$$(x \vee \neg y) \wedge (z \vee w) \wedge (\ldots)$$

Def импликация

$v \to u$

v	u	$v \to u$
0	0	1
0	1	1
1	0	0
1	1	1

$$(v \to u) = (\neg v) \vee u.$$

$$(x \vee y) = (\neg x \to y) =$$
$$= (\neg y \to x)$$

Заведём граф, в котором в-ны это $\neg x$ и x (все литералы)

for $(x \vee y)$ in входн. данные

добавить ребро $\neg x \to y$

и ребро $\neg y \to x$

$$a \to b \to c$$

$$\frac{a \to b^{=1} \text{ и } b \to c^{=1}}{a \to c_{=1}}$$

Предположим есть путь

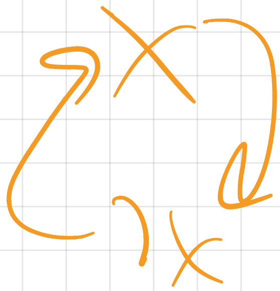
$$x \to (\ldots) \to \neg x$$

$$\frac{x \to \neg x}{\neg x} \quad (\text{т.е. } x = 0)$$

Пусть сущ. путь $x \to \neg x$, тогда $kcc[x] \leq kcc[\neg x]$

Предположим есть.

$$x \to \neg x$$



и $\neg x \to x$

$\Rightarrow$ нет решения.

$x \to \neg x$ и $\neg x \to x$ есть $\Leftrightarrow$ x и $\neg x$ лежат в общей к.с.с.

! если $\exists\, x$, что $KCC[x] == KCC[\neg x]$ $\Rightarrow$ проиграли.

Пусть $x_i = 0$, если $KCC[x_i] < KCC[\neg x_i]$

и пусть $x_i = 1$, если $KCC[x_i] > KCC[\neg x_i]$

<u>Утв</u> x_i — решение

Пусть не решение.

Тогда $\exists\, y \to z$ среди рёбер графа, что $y = 1$, $z = 0$

(y, z — литералы)

$\Theta(\text{input})$

$$\begin{array}{ccc} 1 = y & & \neg y \\ \downarrow & & \uparrow \\ 0 = z & & \neg z \end{array}$$

$y=1 \Rightarrow KCC[y] > KCC[\neg y]$

$z=0 \Rightarrow KCC[\neg z] > KCC[z]$

$y \rightarrow z \Rightarrow KCC[z] \geq KCC[y]$

$\neg z \rightarrow \neg y \Rightarrow KCC[\neg y] \geq KCC[\neg z]$

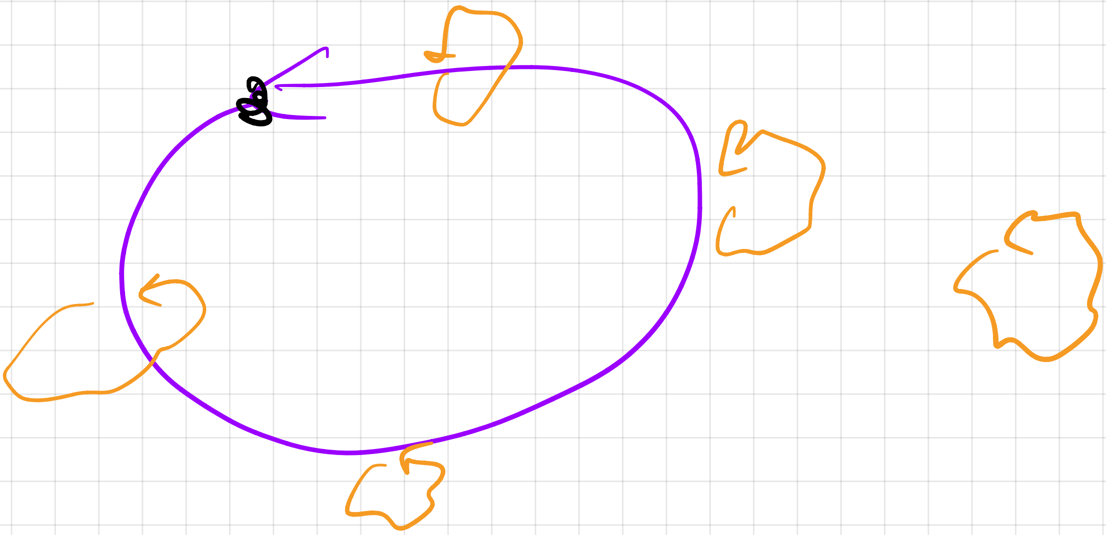
$$\begin{array}{ccc} a & > & b \\ \wedge & & \vee \\ 0 & < & c \end{array}$$

$\square$

Эйлеров путь (цикл)

$\deg^-(v) = \deg^+(v)$, иначе ответа не сущ.

идея алгоритма.



go(v) {

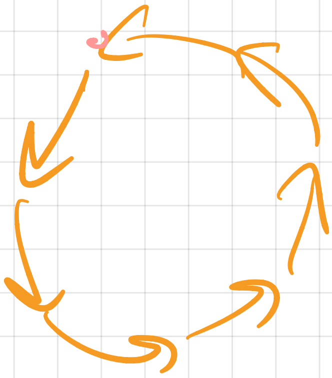
while $\exists\, v \to u$ — не посещ. ребро

(посетить ребро)

go(u)

ans.push_back($v \to u$)

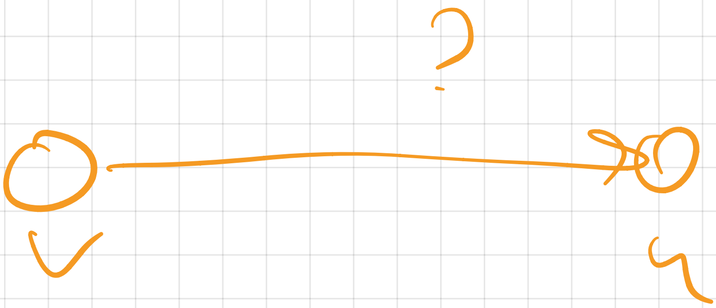
}

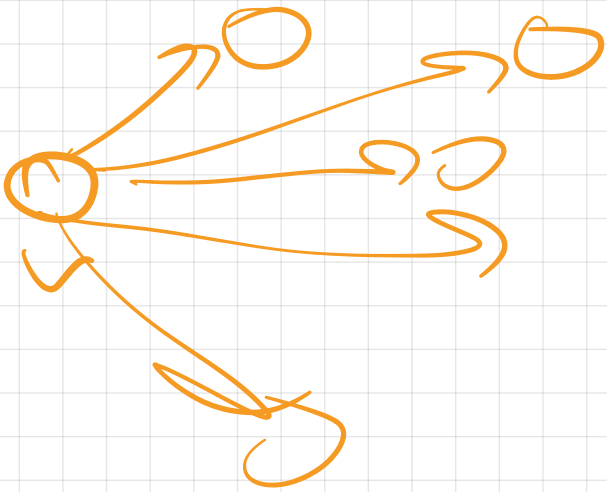


$O(V+E)$

BFS. (поиск в ширину)

Типы задач поиска кр. пути.

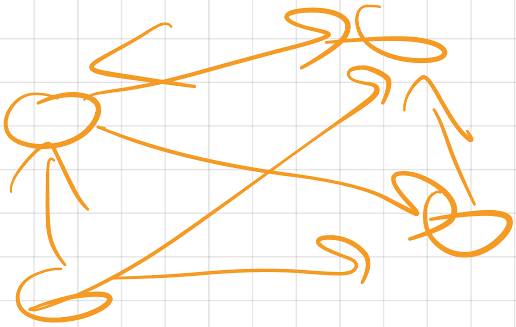
1) 

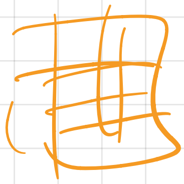
2) 

SSSP

single source short. paths

3)

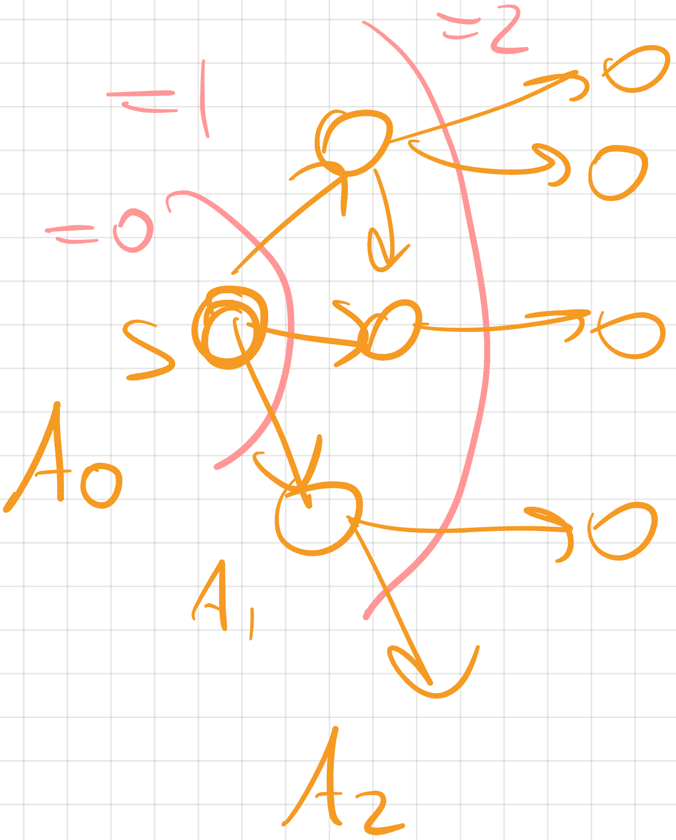




APSP

All pairs shortest paths

BFS (SSSP)



$N(S) =$

$= \bigcup_{v \in S} \delta(v)$

$A_d = N(A_{d-1}) \setminus (A_0 \cup \dots \cup A_{d-1})$

↑ used

$A = [[\,] - - - -]$

$A[0] = S$

$used[S] = 1$

for $d = 0 .. n-1$

for $v \in A[d]$:

for $u \in adj[v]$

if $!used[u]$

$A[d+1] \leftarrow u$

$used[u] = 1$.