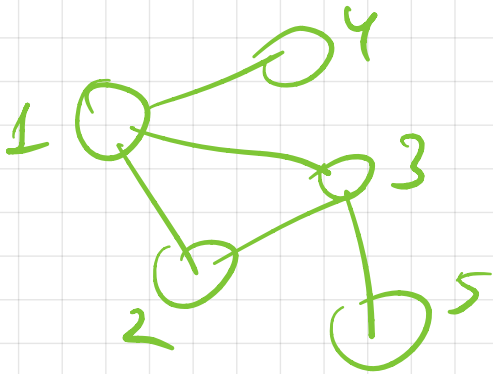
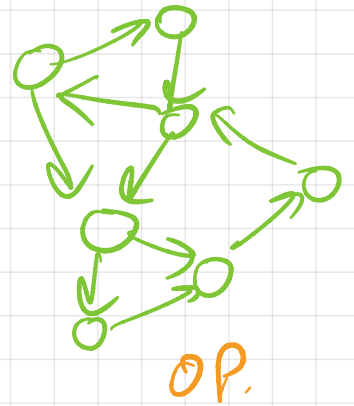


Графы и Поиск в ширину

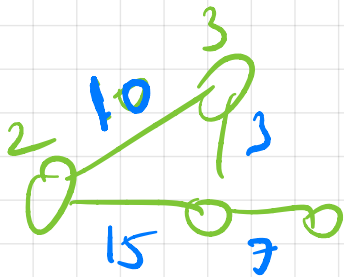


Класс.

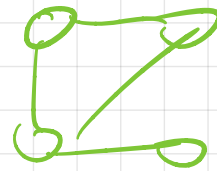


OP.

$$G = (V, E)$$

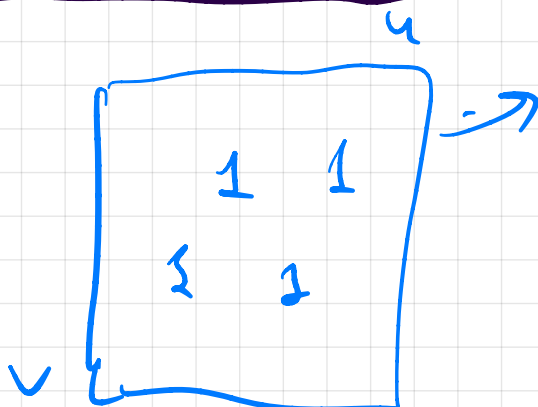


взв.



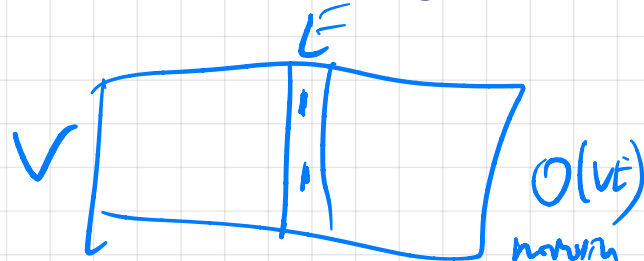
не взв.

МАТРИЦА СМЕЖНОСТИ



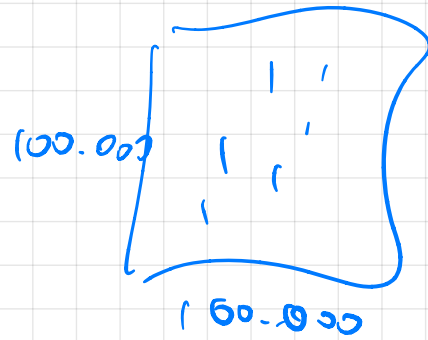
1 == есть ребро?

МАТР. смежности.



$$E \approx V$$

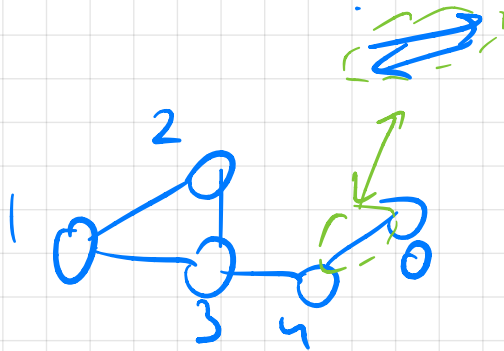
$$E = V = 100.000$$



$O(V^2)$
нормата

Список смежности

$$E \approx V^2$$



$$0: 4$$

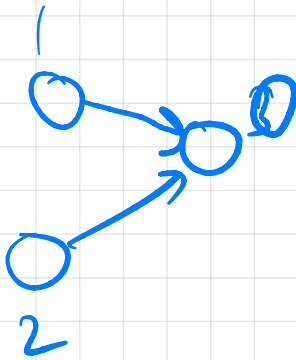
$$1: 2, 3$$

$$2: 1, 3$$

$$3: 1, 2, 4$$

$$4: 3, 0$$

$O(V+E)$
нормата



$$0:$$

$$1: 0$$

$$2: 0$$

Список ребер

$$1\ 2$$

$$1\ 3$$

$$2\ 3$$

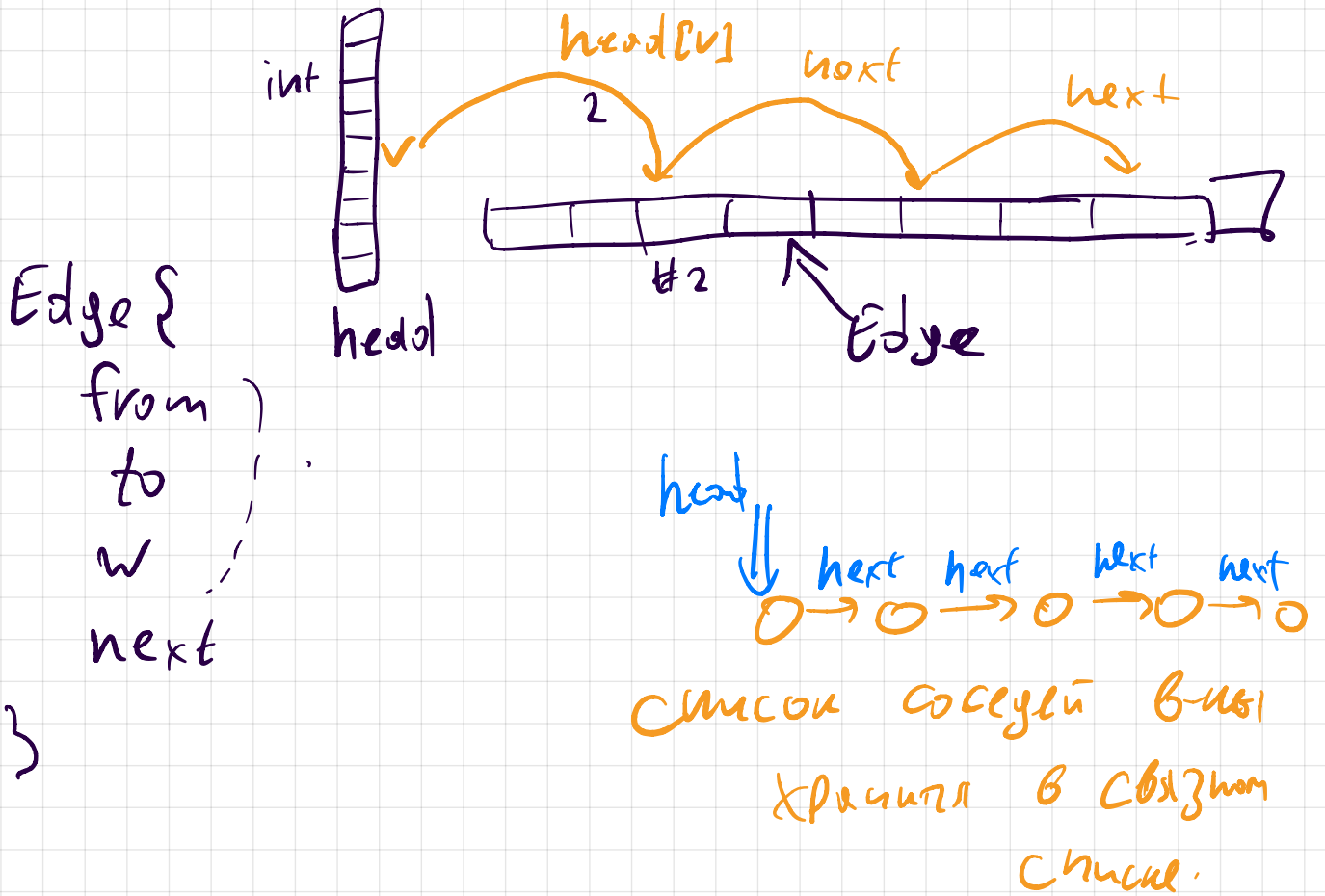
$$3\ 4$$

$$0\ 4$$

$O(V+E)$

нормата

Массив списков смежности.



```
for (int e = head[u]; e != -1;  
     e = edges[e].next)  
    print(edges[e].to)
```

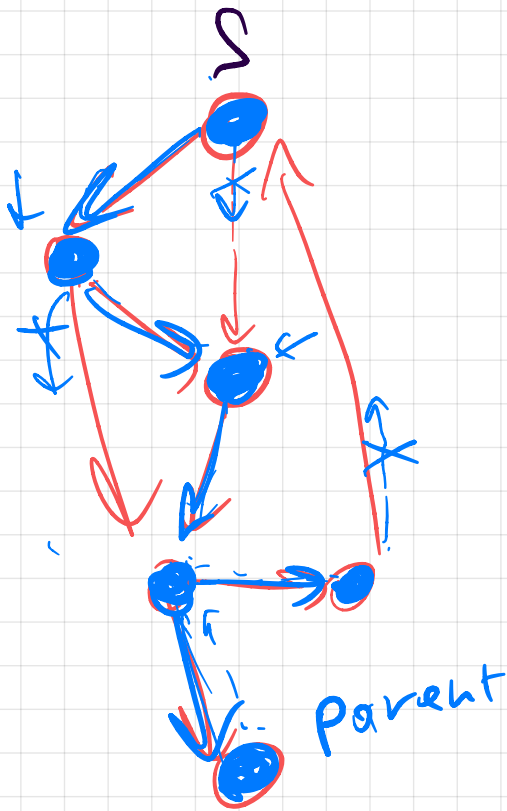
adj: List[List[int]]:

```
for u in adj[u]:  
    print(u)
```

Поиск в змубуны

Tree max

18 букв



Дерево dfs
depth first search

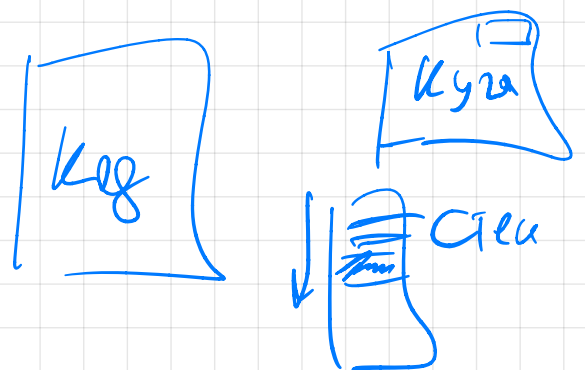
used = [false, ..., false]

dfs(v):

used[v] = True

for u in adj[v]:
if not used[u]:
dfs(u)

dfs(S)



Задача dfs

1) уз оглоби θ -на dfs(s)

2) уз мнозина θ -а:

for $v = 0 \dots n-1$

if not used[v]

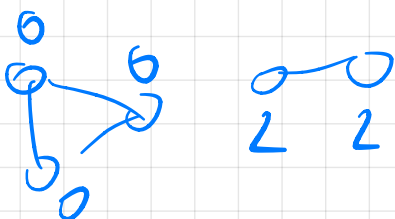
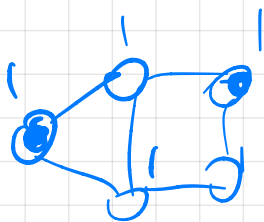
dfs(v)

Асимптотика: $O(V+E)$



Приложения

1) Посчитать число комн. связности.



cnt = 0

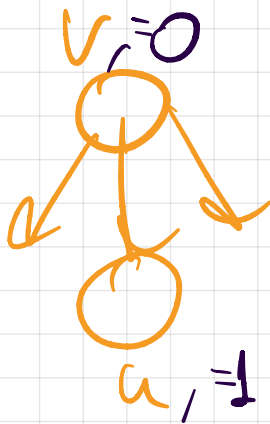
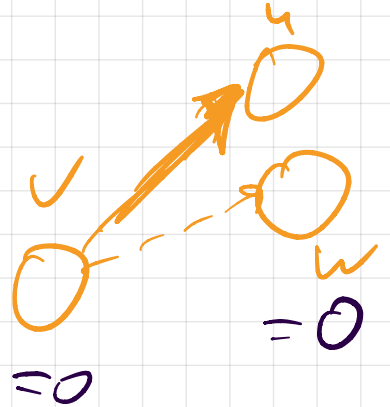
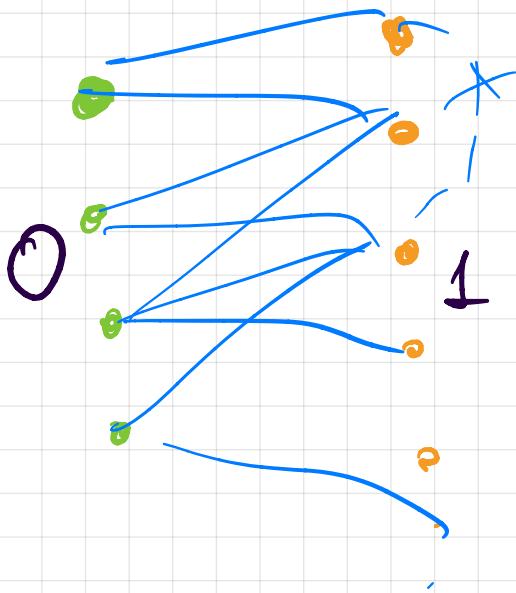
for $v = 0 \dots n-1$

if not used[v]

dfs(v, cnt)

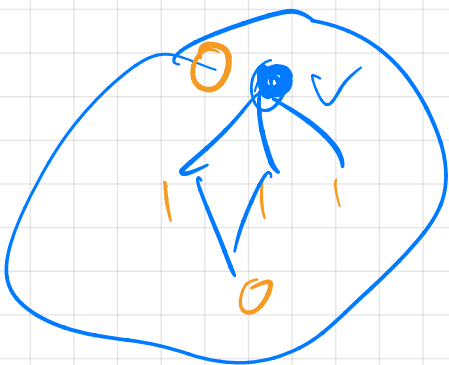
cnt += 1

2) Проверить двузначность графа.



dfs(v):
used[v] = True

for u in adj[v]:
if not used[u]:
types[u] = types[v] ^ 1
dfs(u)
else:
return types[v] != types[u]



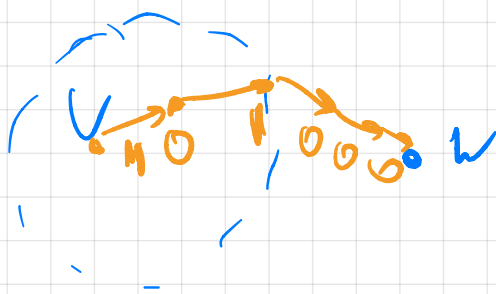
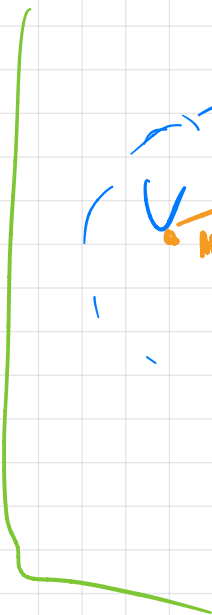
types[v] = 0
dfs(v)

3) Нам надо от V до u

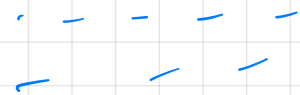
$dfs(v)$

$used[u] = true$

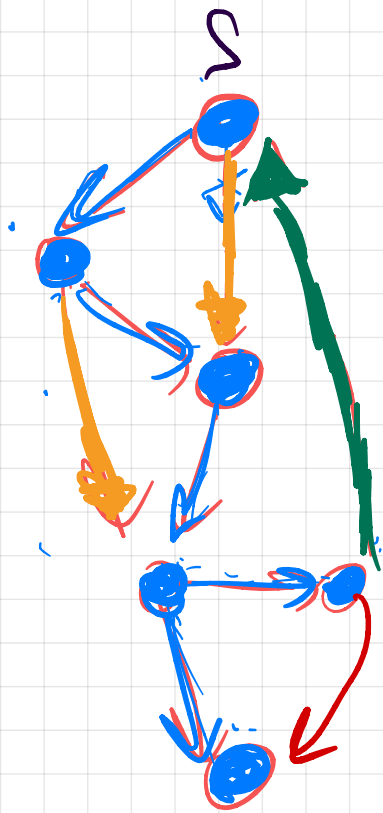
ГТБ: $dfs(v)$ посещает все вершины достижимые из V



$dfs(v) \rightarrow bool$



Классификация рёбер

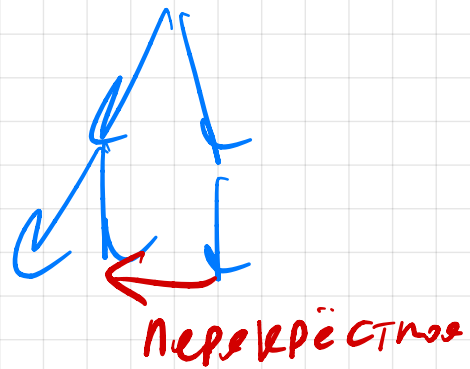
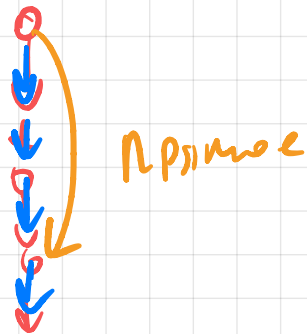
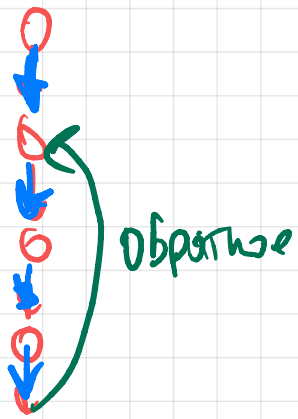


↘ Прямое

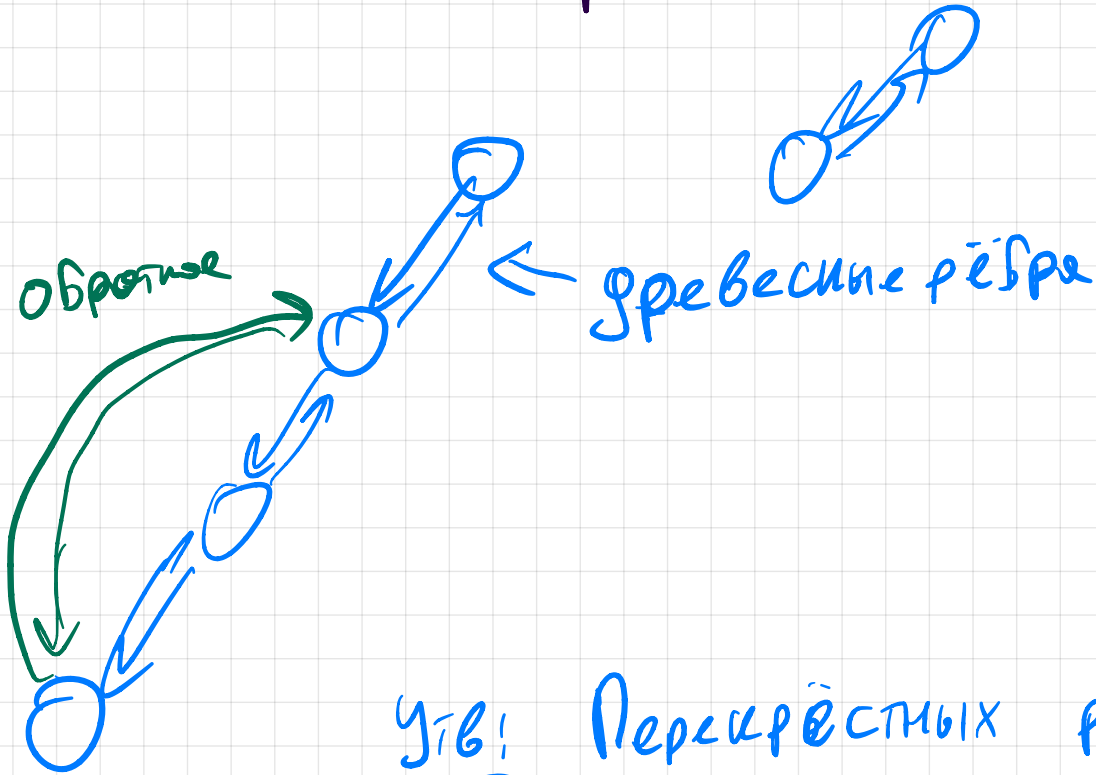
↑ Обратное

↓ Прямое

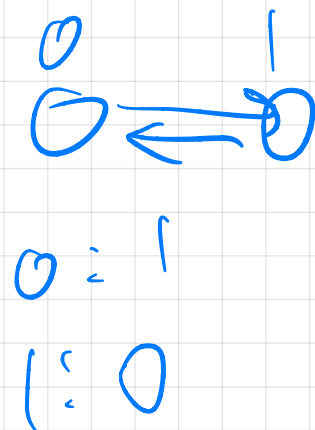
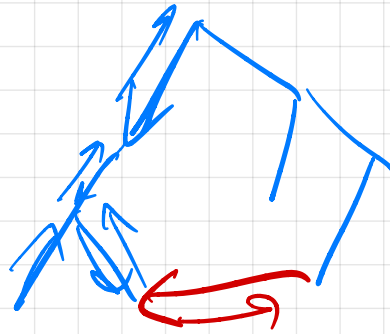
↪ Перекрестное



Упор. графы



УТВ! Перекрёстных рёбер
не бывает в
не упор. графах



Поиск цикла

Heap. space.

$$O(V+E)$$

$dfs(u, par)$

$used[u] = 1$

for $u \in adj[u]$:

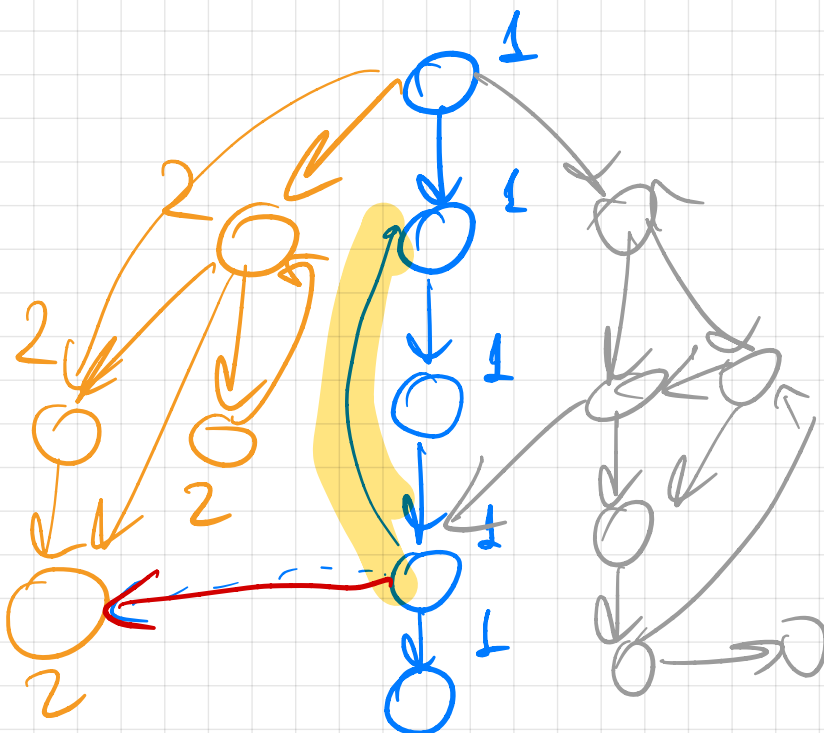
if not $used[u]$:

$dfs(u)$

elif $u \neq par$:

Обратное ребро
и значит цикл

Ориент. графы, 3 цвета



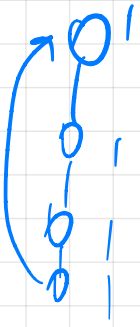
dfs(v)

used[v] = 1

for u in adj[v]:

посетить если used = 0

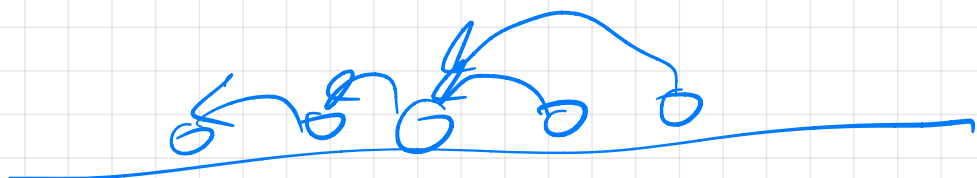
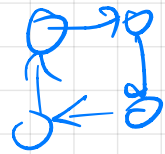
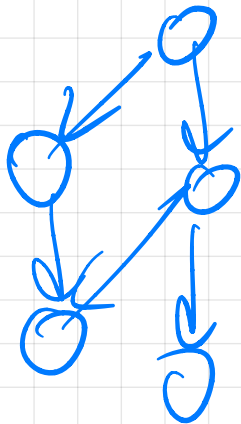
used[v] = 2



УТВ: обратное ребро это то, которое к моменту ~~возвращения~~ возврата в 1.

Топологически

Сортировка



$t=0$
 $dfs(v)$

$t_{in}, t_{out} =$
 $=$ время входа
и выхода

$used[v] = 1$

$t_{in}[v] = t++$

for $u \in \text{adj}[v]$:
if not $used[u]$
 $dfs(u)$

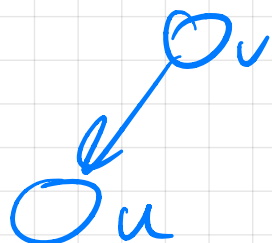
$t_{out}[v] = t++$
(order. push-back(v))

for $v = 0..n-1$
if ! $used[v]$:
 $dfs(v)$

уб (путь в графе и порядок вершин) и

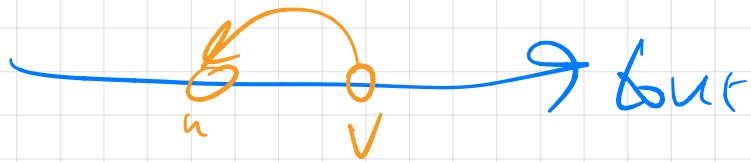
есть ребро $v \rightarrow u$,

тогда $t_{out}[v] > t_{out}[u]$

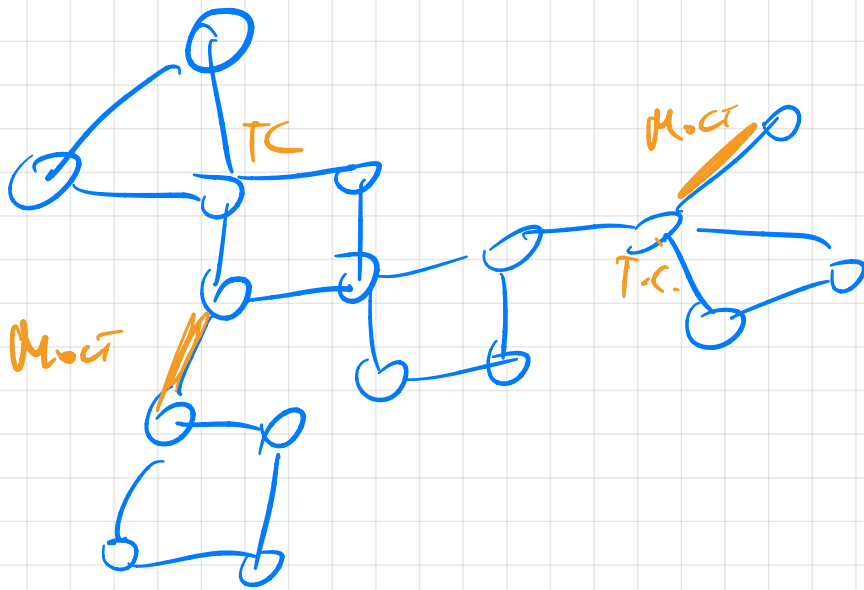


$t_{out}[v]$
 $>$
 $t_{out}[u]$

Применение к броду



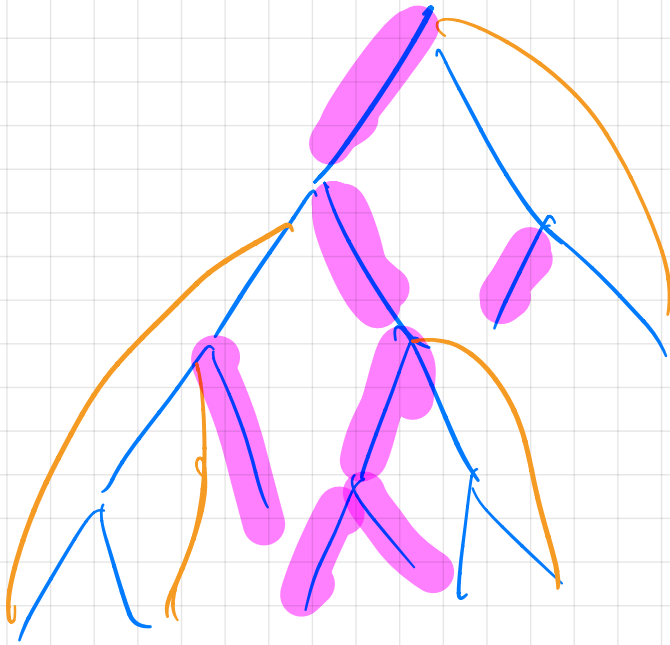
Мосты и Точки Соединения.



Def e -мост, если $c(G-e) > c(G)$

Def v -Т.с. если $c(G-v) < c(G)$

Мост.



Уте обратное ребро.
исключая мост

Утв предельное ребро
мост если
не найдено
обратным.