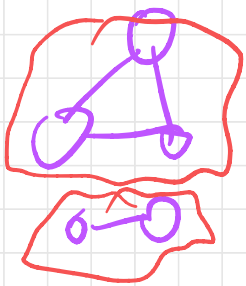
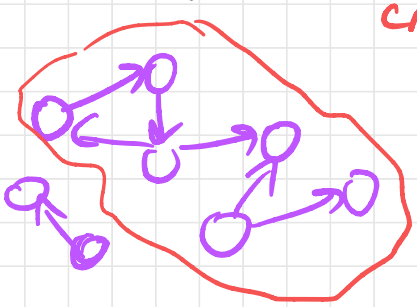
SCC

↑ Strongly connected components





слабая св.

$v \sim u$, если

$\exists\ v \xrightarrow{\text{путь}} u$

и

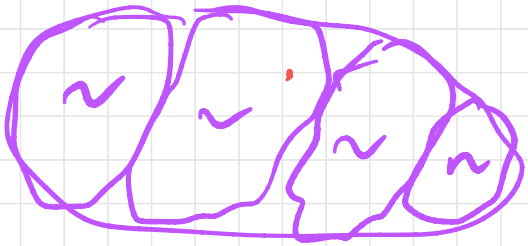
$\exists\ u \xrightarrow{\text{путь}} v$

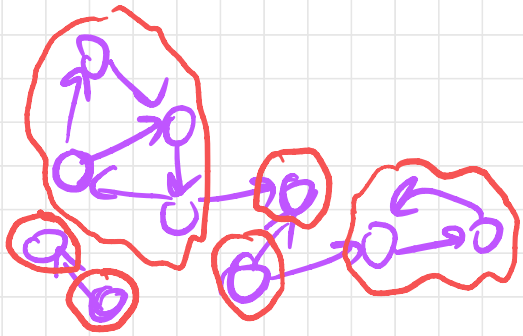
) сильная связность

Отношение экв.

- $\forall v \quad v \sim v$
- $\forall v,u \quad v \sim u \Rightarrow u \sim v$
- $\forall v,u,w \quad v \sim u,\ u \sim w \Rightarrow v \sim w$

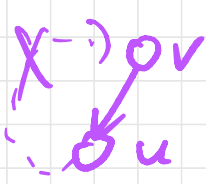
$\Rightarrow \exists$ разбиение на классы экв



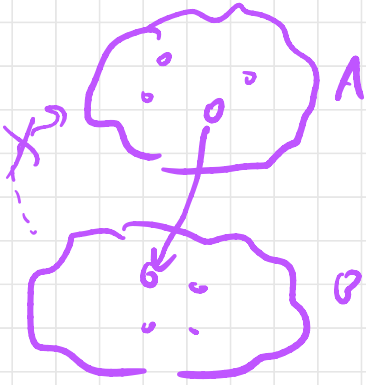


Алгоритм

Kosaraju 1978, Sharir 1981



$tout[v] > tout[u]$



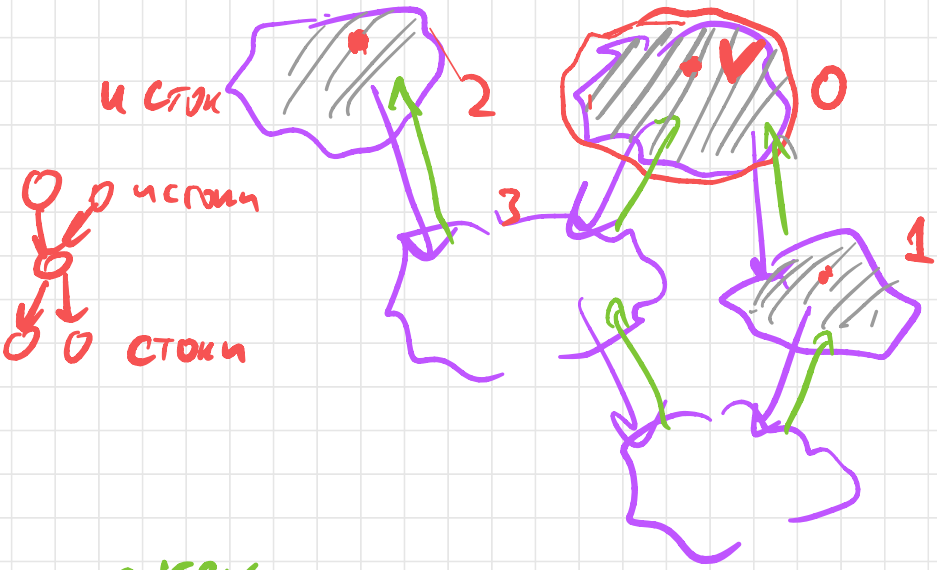
$tout[A] > tout[B]$

$=$

$\max_{v \in A} tout[v]$

G: tout, order

v: tout[v] = max



G^{rev}

dfs(G^{rev}, v)

Алгоритм:

1) Сортировка по топ

```
used = [false...]
order = []
for v = 0..n-1:
    if !used[v]
        dfs(G, v, order)
```

топ

2) `reverse(order)`

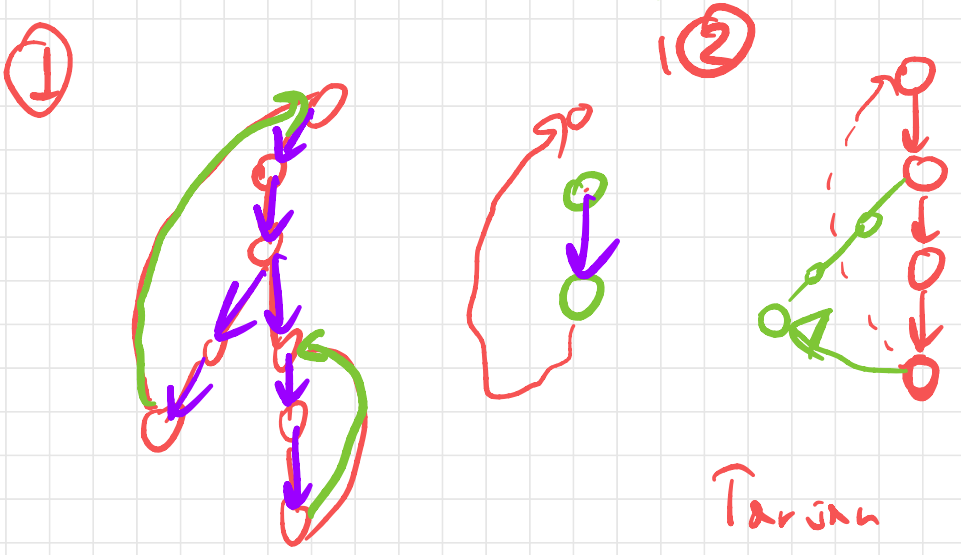
3) Выделение компонент

```
cid = [-1... -1]
no = 0
for v in order:
    if cid(v) == -1:
        dfs2(G^rev, v, no)
        no += 1
```

```
dfs2(G, v, no):
    cid[v] = no
    for u in G[v]:
        if cid[u] == -1:
            dfs2(G, u, no)
```

$O(V+E)$

SCC 의 종류

① ② 

Tarjan

2-SAT ← Satisfiability

$(x_1 \text{ или } x_2) \text{ и } (x_1 \vee \neg x_2) \text{ и } (x_2 \vee x_3)$

$(x_1 \vee x_2)$ — клозы

$x_1, \neg x_2, x_2, x_3$ — литерал

Проверить: $\exists$ ли выполняющий набор

2-SAT: $|\text{клоз}| \le 2$

линейный алгоритм

k-SAT: $|\text{клоз}| \in k$

$\forall k \ge 3$: предполагается

NP-трудная — это нельзя решить быстрее чем за c^n

SAT: нет ограничения

предполагается,

SETH: это нельзя решить

за $O(2^{cn})$ $\forall c < 1$

2-SAT

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

импликация

$x \vee y = 1$ ②

$\neg x \to y = 1$

a	b	$a \to b$
0	0	1
0	1	1
1	0	0
1	1	1

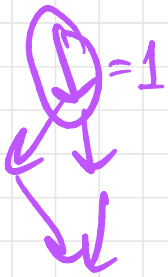
$a \to b =$

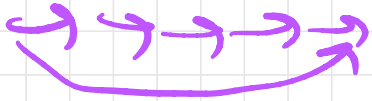
$= \neg a \vee b$

$V(G) = \{x_i, \neg x_i\}$

$E(G) = \{ \neg x \to y;\ \neg y \to x$ | $(x \vee y) \in$ formula

$(x \vee y)$ ← дизъюнкт





Лемма: $(a \to b) = 1 \;;\; (b \to c) = 1$

$\Downarrow$

$(a \to c) = 1$

Лемма: $x \longrightarrow \neg x$

$x = 0$

Лемма: $x \longrightarrow \neg x \;\Big\} \Rightarrow x = 0$

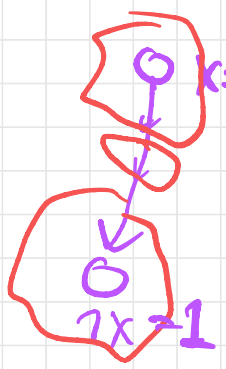
$\neg x \longrightarrow x \;\Big\} \Rightarrow x = 1$

нет решения Φ_{01}

$c_1 d[x] = c_2 0[\neg x]$.

Алгоритм

1) Создать граф
2) Вычислить К.С.С. $O(V+E)$
3) если $\exists x$: $cid[x] = cid[\neg x]$,
то решить нельзя
4) иначе:
$cid[x] < cid[\neg x]$,
то $x = 0$
$cid[x] > cid[\neg x]$
то $x = 1$



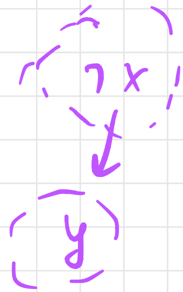
$cid[x]$ $cid[\neg x]$

Д-во корректности

$(x \vee y)$ - клоз

$(x \vee y) = 0 \Rightarrow x=0$ и $y=0$

$\neg x \to y$; $\neg y \to x$



① $cid[\neg x] \le cid[y]$

② $cid[\neg y] \le cid[x]$

$x=0 \to$ ③ $cid[x] < cid[\neg x]$

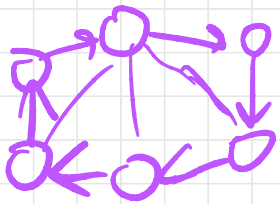
$y=0 \to$ ④ $cid[y] < cid[\neg y]$

$$cid[\neg y] \overset{2}{\le} cid[x] \overset{3}{<} cid[\neg x] \overset{1}{\le} cid[y] \overset{4}{<} cid[\neg y]$$

$\Rightarrow$ Такого клоза $(x \vee y)$ не сущ. $\square$.

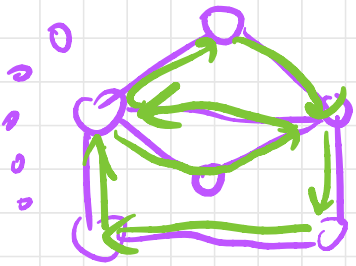
$$deg(v) = |\{u \mid \{v,u\} \in E\}|$$

Euler Cycle



Гамильтонов цикл (по всем вершинам)

NP-трудная



Эйлеров цикл (по всем рёбрам)

Линейное время

Гамильтонов путь; Эйлеров путь.

Лм. ∃ Эйлеров цикл ⇔

1) граф связен (исключая изол. вершины)

2) степень всех вершин чётна

Эйлеров путь.

1) всё ещё нужна связность

2) не более двух вершин нечётной степени

0 → найти цикл

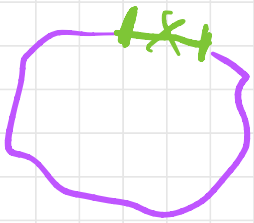
1 → не бывает

2 → 

$\sum deg(v) = 2|E|$

↓

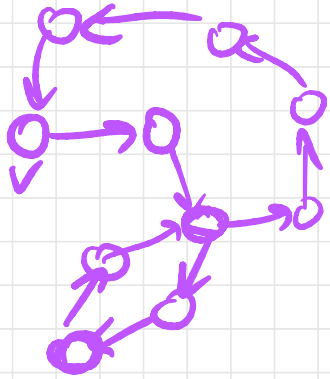
Кол-во нечет. вершин чётно.

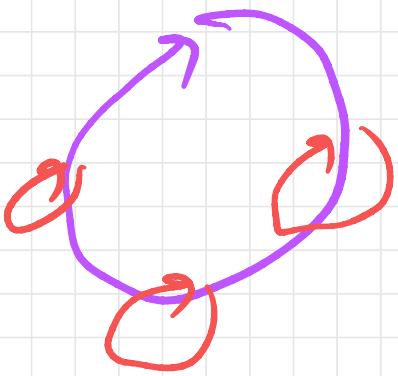


↓

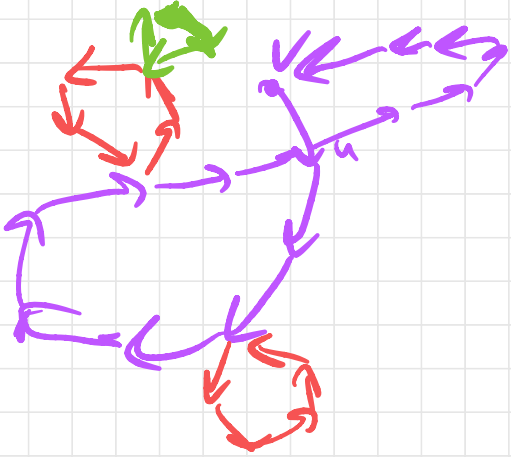
путь

Алгоритм для цикла.





```
euler(v):
    while adj[v] ≠ []:
        u = adj[v].pop()
        euler(u)
        ans.append({u, v})
```

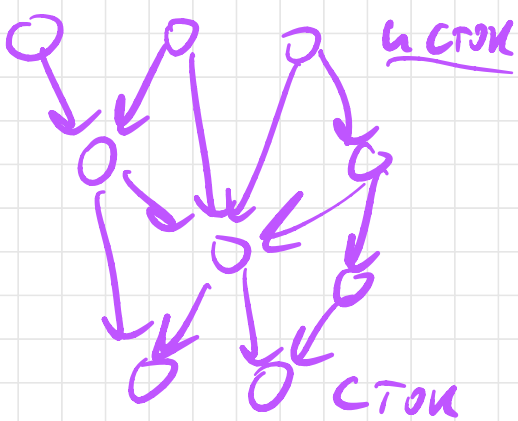


$O(V+E)$

Misc

Другой алгоритм поиска top. Sort. 1) топ

2)



Лемма: В любом Ац. графе $\exists$ 4 сток недостижим 4 сток

While True:

1) найти исток v ∈ for(v)

2) удалить v

3) ans.append(v)

indeg[v] = степень (вход.) вершины

1) вычисляем indeg

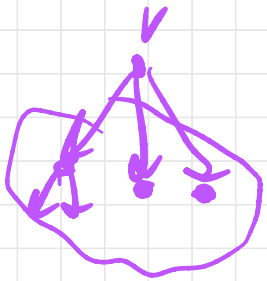
2) q = Queue()

добавляем туда все вершины с indeg[v] = 0

3) while len(q) > 0:

v = q.pop()

ans.append(v)



O(V+E)

```
for u in adj^out[v]:
    indeg[u] -= 1
    if indeg[u] == 0:
        q.push(u)
```