

Графы $G(V, E) \rightarrow$ ориент.

$$E \subseteq V^2$$

Обычно нет

- кратные ребра
- петли

$e = (v, u) \in E \Leftrightarrow$ v и u смежные
 v инцидентно e
 e инц. v

Неор. граф: v и u связаны $\Leftrightarrow \exists p_1, p_2, \dots, p_k \in V$
 $(p_1, p_2), (p_2, p_3), \dots, (p_{k-1}, p_k) \in E$
 $p_1 = v, p_k = u$

связн. - отн. эквив.
 1) v и v связн.
 2) v и u связн. $\Rightarrow u$ и v связн.
 3) v и w связн., w и x связн. $\Rightarrow v$ и x св.



Хранение графа

1) Матрица смежности

$$n = |V| \quad m = |E|$$



$$M_{ij} = \begin{cases} 1, & \text{если ребро } i \rightarrow j \\ 0, & \text{если нет} \end{cases}$$

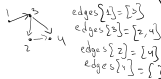
Память: $\Theta(N^2) = O(n^2)$

Время: 1) есть ли ребро $i \rightarrow j$ $O(1)$

2) $i \rightarrow$ перебрать всех соседей
 $O(|V|) = O(n)$

2) Списки смежности (списки инцидентности)

$v \rightarrow \text{edges}[v]$



Память: $O(V+E)$ (ор. граф $\rightarrow \forall$ ребро $\approx$ раз соп.)

Время: 1) $v \rightarrow u$: $O(\log V)$ неор. граф. $\rightarrow$ 2 раза

2) $v \rightarrow \text{edges}[v]$ $O(\deg(v))$

Поиск в глубину (depth-first search) dfs

нет деления и кратн. ребер.

$$\text{неор. граф: } |E| \leq \frac{|V|(|V|-1)}{2}$$

$$\text{ор. граф: } |E| \leq \frac{|V|(|V|-1)}{2}$$

def dfs(v):

used[v] = True

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

Списки смежности:

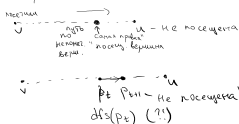
$$O(V+E)$$

Матрица см.:

$$O(V) + O(V^2) = O(V^2)$$

Утв. dfs(v) посетит те и только те вершины G, кот. достижима из v по непомет. верш. (used=False)

До-во. От противного.



$p_1, p_2, \dots, p_k$
" " " "
" " " "
 $t = \max_i: p_i \text{ посещ.$
 $i \leq k$

Компоненты связности: (граф неор.)

for $v \in V$:
if not used[v]:
dfs(v)

$$O(V+E)$$

Поиск цикла:

1) неор. dfs(v, p=None):
used[v] = True
for u in edges[v]:
if not used[u]:
dfs(u, v)
elif u != p:
Нашли цикл

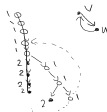


2) ориент.

три цвета вершины:
белый used[v] = 0
серый used[v] = 1 вошли, но не вышли
зеленый used[v] = 2



dfs(v):
used[v] = 1
for u in edges[v]:
if used[u] == 0:
dfs(u)
elif used[u] == 1:
Нашли цикл
used[v] = 2



Топологическая сортировка:



$$O(V+E)$$

очередь $Q \leftarrow$ все истоки $G: \{ \text{все } v, \text{deg-in}[v] = 0 \}$

while (!Q.empty())

$v = Q.pop()$

topsort.append(v)

for u in edges[v]

deg-in[u] -= 1

if deg-in[u] == 0

Q.append(u)

$$O(V+E)$$

Граф ориент.

порядок верш. такой, что все ребра из u в v

цикл $\Rightarrow$ тополог. нет $\Rightarrow$ нет

$\exists$ вершина с вх. степенью 0 (исток)

