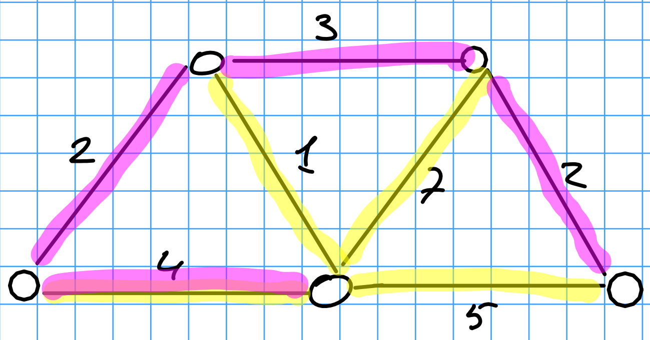
Минимальное остовное дерево (MST, Minimal Spanning Tree)

NB: "жадные" - "greedy"

Минимальное остовное дерево для $G=(V,E)$ это граф $T=(V,E')$, $E' \subseteq E$, T - дерево. и $\sum\limits_{e \in E'} w(e)$ - минимальна (по выбору E')



Стоимость = 17

Стоимость = 11

...

Свойство разреза

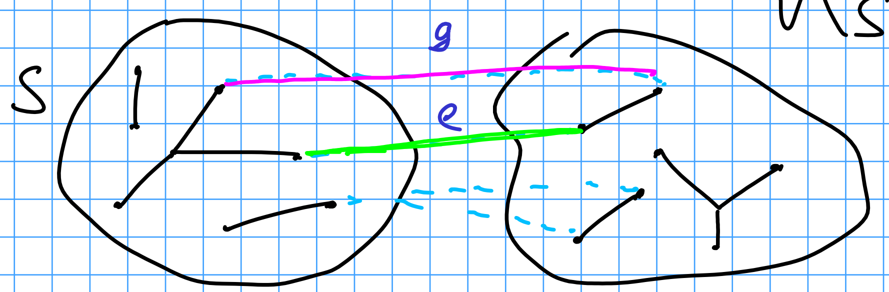
$\exists\ X \subset MST$

$\exists S \subset V$: в X нет ребра из S в $V \setminus S$

$\exists e$ - это минимальное ребро из S в $V \setminus S$

$\Downarrow$

$X \cup \{e\} \subset MST$



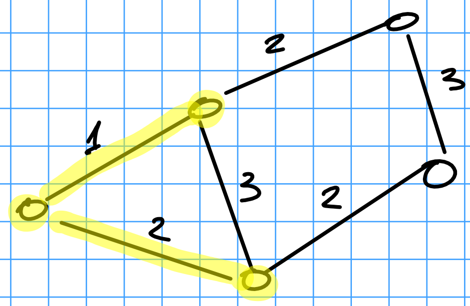
▷ $\exists\ e \notin MST \Rightarrow \exists\ g \in E$ из S в $V \setminus S$: $g \in MST$

Добавим e в MST $\Rightarrow$ образуется цикл.

$\exists$ цикл проходит по g.

Если мы выкинем g, то получим некоторое ST'.
Стоимость дерева не возросла, т.к. $w(e) \le w(g)$. $\Rightarrow$ получили MST.
(i.e. $w(e) = w(g)$). $\triangleleft$

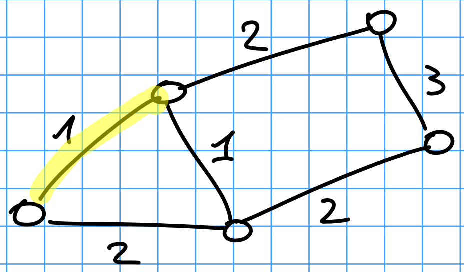
Алгоритм Прима



$O((V+E)\log V)$

```
Prim((V, E), v0)
  for v ∈ V:
      dist[v] = ∞
      prev[v] = -1
  dist[v0] = 0
  P = Make_Priority_Queue()
  P.enqueue(v0, 0)
  while P.size() ≠ 0
      v = P.extract_min()
      for (v, u) ∈ E:
          if dist[u] = ∞:
              P.enqueue(u, w(v,u))
              prev[u] = v, dist[u] = w(v,u)        было в Дейкстре
          if dist[u] > w(v, u):                    + dist[v]
              dist[u] = w(v, u)                    + dist[v]
              P.decrease_key(u, dist[u])
              prev[u] = v
```

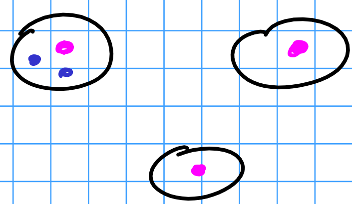
Алгоритъм Крускала



Kruskal((V, E)):
  for $v \in V$:
    Make_Set(v)
  Sort (E по w)
  $M = \emptyset$
  for $(u,v) \in E$
    if find(u) $\neq$ find(v):
      $M = M \cup \{(u,v)\}$
      union(u, v)
  return M

$O(V)$
$O(E \log E)$
$O(E)$
$O(E)$
$\to O(E \log V)$
$O(V^2)$
$O(V)$

Disjoint Sets
Make_Set(v)
Find(v)
Union(u, v)

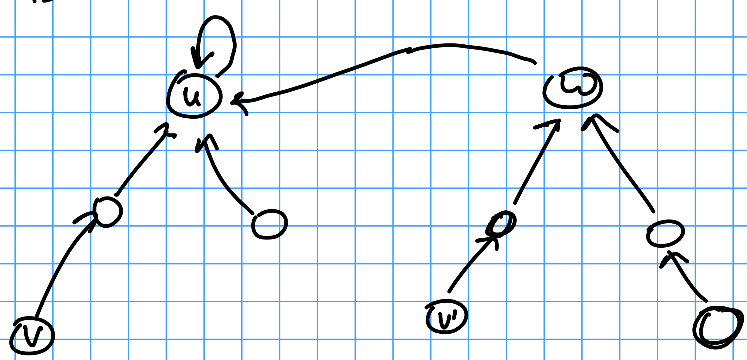


Реализация Disjoint Sets:

1. DFS на каждом шаге
   Make Set за $O(1)$
   Find за $O(1)$
   Union за $O(V)$
   $\Rightarrow O(V^2 + E \log E)$ $O(V^2)$

2. Disjoint Sets на DAG

   Make Set (u)
   Find (v)
   Union (u, w)



Наивная реализация

Make Set (v):

  parent[v] = v

Find (v):

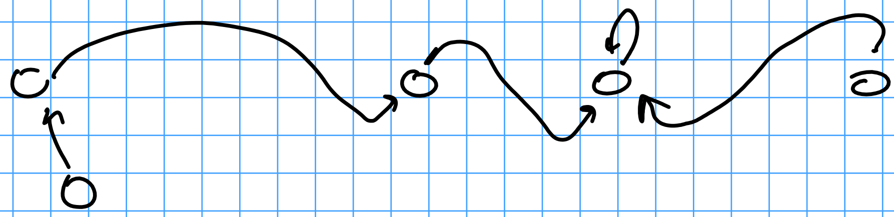
  while parent[v] ≠ v:

    v = parent[v]

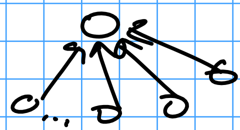
  return v

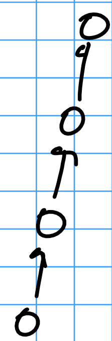
Union (u, w): // u и w - представители

  parent[w] = u



$\Rightarrow O(E \cdot V)$





Реализация с рангами:

Make_Set (v)

  parent[v] = v

  rank(v) = 0

Find (v):

  while parent[v] ≠ v:

    v = parent[v]

  return v

Union (v, w):

  if rank(v) > rank(w):

    parent[w] = v

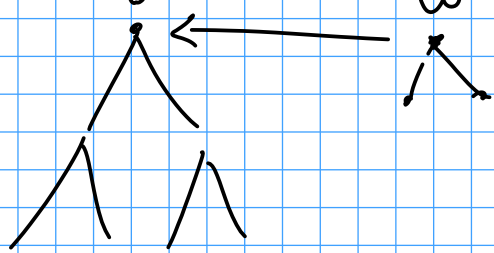
  if rank(v) < rank(w)

    parent[v] = w

  if rank[v] = rank[w]:

    parent[v] = w

    rank[w] += 1



Утв: rank = высоте.

Утв: У компоненты с rank = k $\geq 2^k - 1$ потомков

$\geq 2^k - 1 + 2^k - 1 + 1 = 2^{k+1} - 1$ (по индукции).

$\Downarrow$

$\max k \quad 2^k \leq |V| \Rightarrow k \leq \log |V| \Rightarrow$

Find работает за $O(\log |V|)$

$\Rightarrow$ Алгоритм Крускала за $O(E \log V)$