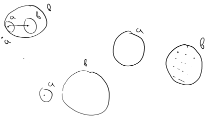
Система непересекающихся мн-в (СНМ) (Disjoint set union) DSU

n элементов, ∀ в своём мн.-ве
1). лежат ли два эл-та в одном мн-ве $get(v) \overset{?}{==} get(u)$
2). объед. два мн-ва $join(v,u)$

Реализация на векторах

```
def init(n):
    comp=[[i] for i in range(n)]      # ∀i  comp[i]=[i]
    color=[i for i in range(n)]       # ∀i  color[i]=i

def get(v):          O(1)
    return color[v]

def join(v,u):       O(|comp(b)|)
    a=get(v); b=get(u)
    if a==b:
        return

    for x in comp[b]:
        color[x]=a
    comp[a]+=comp[b]
    comp[b]=[]
```



$O(n^2)$ – за n запросов

Оптимизация „меньшее к большему"

```
def join(v,u):  O(min(|comp[a]|,|comp[b]|))
    a=get(v); b=get(u)
    if a==b:
        return
    if len(comp[b])>len(comp(a)): swap(a,b)
    for x in comp[b]:
        color[x]=a
    comp[a]+=comp[b]
    comp[b]=[]
```

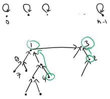
За m запросов:
Следим за жизнью эл-та v. Рожд. в мн-ве размера 1.
Если перекрашиваем v, то размер его мн-ва увеличив. хотя бы в 2 раза.
Знасит, v перекрашено $\le \log_2 n$ раз.

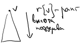
Всё время работы – сумма времён перекраш. по всем вершинам.
$Time = O(n \log n)$

Следствие Алг. Краскала с такой реализ. имеет $Time = O(\underbrace{E \log E}_{sort} + V \log V) = O(E \log E)$ ($E \ge V-1$, т.к. G связен)

Реализация с помощью леса

```
def init(n):
    p=[i for i in range(n)]


def join(v,u):      O(get+1)
    a=get(v); b=get(u)
    if a==b: return

    p[b]=a

def get(v):         O(расстояние от v до корня)
    if v==p[v]:                 while v!=p[v]:
        return v                    v=p[v]
    else:                       return v
        return get(p[v])
```





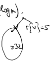
r[v] – ранг
высота поддерева

Эвристики

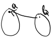
① Ранговая эвристика

```
def init(n):
    p=[i for i in range(n)]
    r=[0]·n

def join(v,u):      O(get+1)
    a=get(v); b=get(u)
    if a==b: return
    if r[b]>r[a]: swap(a,b)
    p[b]=a
    if r[a]==r[b]:

        r[a]+=1
```

Утв. get работает за $O(\log n)$.
Д-во Заметим, что $\forall v: v \ne p[v]$ $r[p[v]] > r[v]$.
get на ∀ шаге переходит к вершине с бо́льшим рангом.
Осталось показать, что все $r[v] \le O(\log n)$. Тогда get работает за $O(\log n)$.



Lm. ∀ v размер поддерева v $\ge 2^{r[v]}$.
Д-во. По индукции.
База: ∀ v $r[v]=0$. Размер поддерева = 1. $2^0 = 1$.
Переход. 1). $r[a] > r[b]$. $r'[a] = r[a]$
Новый размер поддерева a – сумма размеров поддеревьев a и b ≥
$\ge 2^{r[a]} + 2^{r[b]} = 2^{r'[a]} + 2^{r[b]} \ge 2^{r'[a]}$ ⊕



2). $r[a] = r[b]$. $r'[a] = r[a]+1$
Новый размер подд. a $\ge 2^{r[a]} + 2^{r[b]} = 2^{r[a]} \cdot 2 = 2^{r[a]+1} = 2^{r'[a]}$.

■

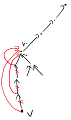
Знасит, у любой вершины $r[v] \le \log_2 n$.

■

Следствие Алг. Краскала работает с такой реал. за $O(\underbrace{E \log E}_{Sort} + E \log V) = O(E \log E)$

② Эвристика сжатия путей

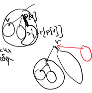
```
def init(n):
    p=[i for i in range(n)]


def join(v,u):      O(get+1)
    a=get(v); b=get(u)
    if a==b: return

    p[b]=a

def get(v):         O(расст. от v до корня)
    if v==p[v]:
        return v
    else
        p[v]=get(p[v])
        return p[v]
```



Утв. m операций get отработают за $O((m+n) \log n)$. (join тоже могут вызываться)
Д-во
Типы рёбер из v в p[v]:
1). p[v] – корень дерева. В ∀ get ≤ 1 такого ребра ⇒ O(m) на обр. таких рёбер.

2). лёгкие рёбра: $2 \cdot \underbrace{sz[v]}_{\text{размер поддерева } v} \le sz[p[v]]$

при прохождении по такому ребру sz[тек. в.] увелич. в ≥ 2 раза
⇒ в ∀ get $\le \log_2 n$ таких рёбер ⇒ $O(m \log n)$ на обр. таких рёбер



3). тяжёлые рёбра: $2 \cdot sz[v] > sz[p[v]]$
$sz[v] > sz[p[v]]/2$
сейчас v переподвесится к корню, кот. не совпадает с $p[v]=u$.
новое $sz'[u] < \frac{1}{2} sz[u]$.
При этом sz[u] уже никогда не увеличивается, т.к. u – не корень.

Знасит, поднимемся в u по тяж. ребру $\le \log_2 n$ раз.
Суммарно во всех get поднимемся по тяж. рёбрам $\le O(n \log n)$ раз.

$Time = O((m+n) \log n)$.

■

③ ①+② эвристика

$Time = O((m+n) \log^* n)$

$2^{2^{2^2}} = 2^{16}$ $\log^*(2^{16}) = 4$
$2^{2^{16}} = 2^{65536}$ $\log^*(2^{2^{16}}) = 5$
$2^{2^{65536}}$ $\log^*(2^{2^{2^{16}}}) = 6$

$Time = O((m+n) \alpha(n))$ – обр. ф-ия Аккермана $\alpha(n) \le 4$ для всех разумных n

Следствие Алг. Краскала работает за $O(\underbrace{Sort}_{E \log E} + (V+E)\alpha(V))$.