Система непер. мн-тв. (DSU)

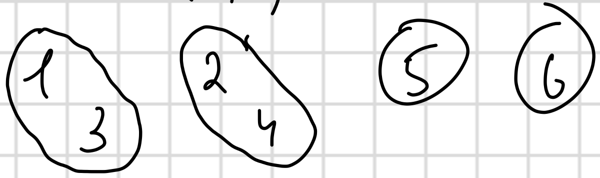
1 2 3 4 ... n ← мн-во

- unite(a, b) объединить

Пр.: 1 2 3 4 5 6

unite(1,3)

unite(2,4)



unite(4,6)

∃ это представители мн-ва

- get(a) получить представителя мн-ва

• unite(a,b)	$O(\log n)$
• get(a)	$O(1)$

∃ 1: {1}

2: {2}

3: {3}

4: {4}

5: {5}

6: {6}

+ для каждого эл-та храним его номер мн-ва

перекладываем/сливаем меньшее в большее →

1: {1, 3}

2: {~~2~~}

3: {~~3~~}

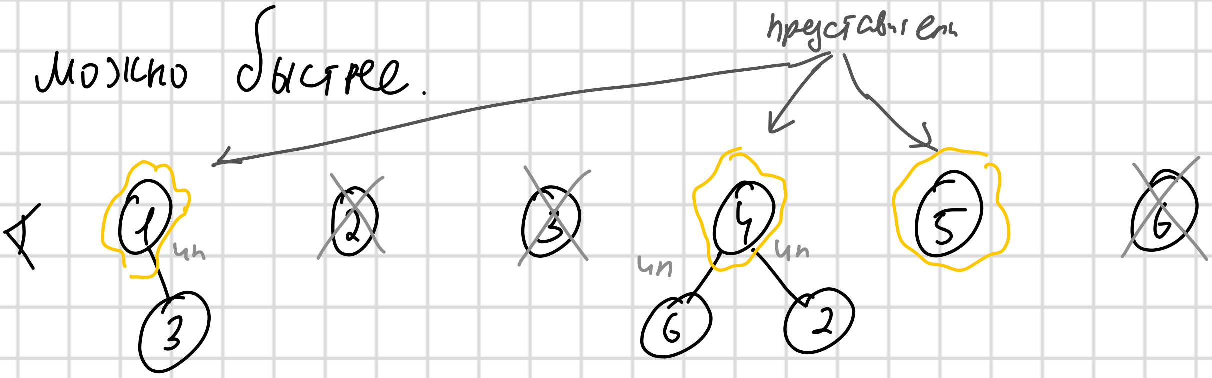
4: {4, 2, 6}

5: {5}

6: {~~6~~}

$O(\log n)$ т.к. когда перекладываем, то всегда размер мн-ва увеличивается, аммортиз. оценка.

можно быстрее.



$p[V]$ – родитель вершины V

(если V - корень, то $p[V] = V$)

- get(V):
```
    if v == p[v]: return v
    return get(p[v])
```
- unite(a,b)
```
    a = get(a)
    b = get(b)
    if a == b: return
    p[a] = b
```
неэффективная зависимость т.к. get зависит от высоты дерева

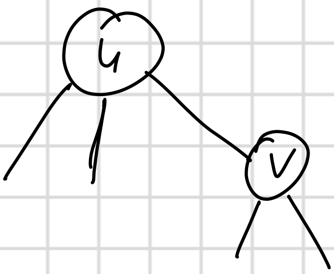
Это дает

unite(a,b)	$O(n)$
get(a)	$O(n)$

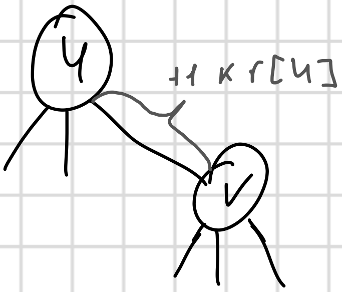
p[v] - родитель v

r[v] - высота вершины v

1) r[v] < r[u], тогда



2) r[v] == r[u], тогда



• unite(a,b)

```
a = get(a)
b = get(b)
if a == b: return
if r[a] > r[b]: swap(a,b)
p[a] = b
if r[a] == r[b]: r[b] += 1
```

Ранговая эвристика

Теперь

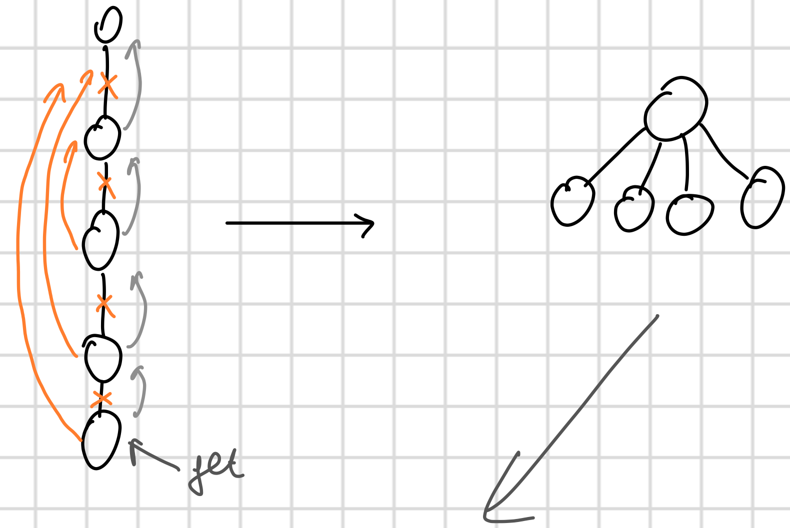
• unite(a,b)	$O(\log n)$
• get(a)	$O(\log n)$

Лемма 1

В дереве с $\overset{r[m]}{\text{рангом}}$ r хотя бы 2^r вершин

Факт: $\forall v$ (кроме корня) $r[p[v]] > r[v]$

$\triangleleft$



• get(v):

```
if v == p[v]: return v
p[v] = get(p[v])
return p[v]
```

и теперь ранг дерева бесполезен

Эвристика сжатия путей

Факт: если убрать ранги, но оставить сжатие путей, то работает за амортизированный $O(\log n)$

Теперь

• unite(a,b)	$O(\log^* n)$
• get(a)	$O(\log^* n)$

Опр.: $F(0) = 1, \quad F(n) = 2^{F(n-1)} \quad (\forall n > 0)$

Д

n	0	1	2	3	4	5
$F(n)$	1	2	4	16	65536	2^{65536}
		$\parallel$	$\parallel$	$\parallel$	$\parallel$	
		2^2	2^{2^2}	$2^{2^{2^2}}$	$2^{2^{\cdots}}$	

$\log_2^* n$ — Итерированный логарифм

$$\log_2^* n = \min K : F(K) \geq n$$

$\log_2^* 1000 = 4$; $\log_2^* 4 = 2$

Лемма 2

Кол-во вершин с рангом r не превосходит $\frac{n}{2^r}$

$\exists$ K вершин с рангом r из л.1. $\geq K \cdot 2^r$ вершин

$$K \cdot 2^r \leq n$$

$$\Downarrow$$

$$K \leq \frac{n}{2^r}$$

Опр.: $g(v)$ — группа вершины v

$$g(v) = \log_2^*(r[v])$$

Всего есть $\leq \log^* n$ групп

Лемма 3:

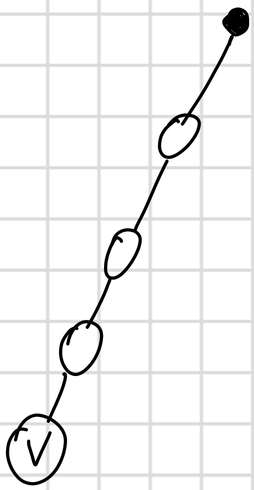
Кол-во вершин в группе $g \leq \dfrac{n}{F(g)}$

в группе g ранги бывают от $F(g-1)+1$ до $F(g)$

$$\sum_{r=F(g-1)+1}^{F(g)} \frac{n}{2^r} \leq \frac{n}{2^{F(g-1)+1}} \cdot \left(1 + \frac{1}{2} + \frac{1}{4} + \ldots\right) = \frac{2n}{2^{F(g-1)+1}} = \frac{n}{2^{F(g-1)}} = \frac{n}{F(g)}$$

если сделаем m вызовов get $\rightarrow$ работает за $O(m \log_2^* n)$ (при $m \geq n$)

▷



$\exists\ A = \{$ корень или сын корня $\}$ их $2m$

$B = \{v : g(v) \neq g(p[v])\}$ $m \log_2^*(n)$

$C = \{v : g(v) = g(p[v])\}$ $n \log^* n$

$g(v) = g \qquad r[v] \in [F(g-1)+1, F(g)]$

$F(g-1)+2 \ldots\ F(g) \rightarrow F(g) - (F(g-1)+2) + 1$
увеличений

$$\sum_{g=0}^{\log_2^* n} \frac{n}{F(g)} \cdot \left(F(g) - F(g-1) - 1\right) \leq \sum_{g=0}^{\log_2^* n} \frac{n}{\cancel{F(g)}} \cdot \cancel{F(g)}$$

$\max\{A, B, C\} = m \log_2^* n$ ◀