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

```
get(x):
    if p[x]==x:
        return x
    else:
        return get(p[x])
```

$\xrightarrow{\text{эвр. сжатия путей}}$

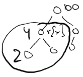
```
get(x):
    if p[x]==x:
        return x
    else:
        p[x]=get(p[x])
        return p[x]
```

```
link(c1,c2): //c1≠c2
    p[c2]=c1
```

$\xrightarrow{\text{эвр. рангов}}$

```
link(c1,c2):
    if r[c2]>r[c1]
        c1,c2=c2,c1
    p[c2]=c1
    if r[c1]==r[c2]:
        r[c1]+=1
```

① Только эвр. рангов



$S(x)$ – поддерево верш. x

Утв. $r[x]$ – высота поддерева x

Утв. x – не корень $\Rightarrow r[p[x]]>r[x]$

$get(x)$ работает $\le O(\max\limits_v r[v])$, т.к. $\geq +1$ на $\forall$ подъём

Утв. Все ранги $\le \log_2 n$.

Lm. $\forall k$ найдётся $\le \frac{n}{2^k}$ вершин ранга k.

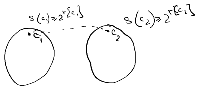
Зам.: $r[x]=r[y] \overset{x\neq y}{\Rightarrow} S(x)\cap S(y)=\varnothing$ (т.к. иначе одно из них в поддереве другого $\Rightarrow$ ранги не равны)

Lm. Заметим, что $|S(x)|\geq 2^{r[x]}$.

Д-во. По инд. по шагам алгоритма.

get – не меняют лес

link(c_1,c_2)



1). $r[c_1]\neq r[c_2]$

Н.У.О., $r[c_1]>r[c_2]$

Ранги не поменялись

$\underbrace{S(c_1)}_{\text{стала больше}}\geq \underbrace{2^{r[c_1]}}_{\text{не поменялось}}$

2). $r[c_1]==r[c_2]$

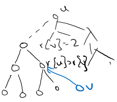
$$\underset{\text{новый размер}}{S'(c_1)} = S(c_1)+S(c_2) \geq 2^{r[c_1]}+2^{r[c_2]} = 2\cdot 2^{r[c_1]} = 2^{r[c_1]+1} = 2^{r'[c_1]}$$

новый ранг

Следствие Лес только с эвр. рангов работает за get: $O(\log n)$ (без аморт.) link: $O(1)$

② Обе эвристики аморт. время работы get $O(\log^* n)$

Жизненный путь произвольной вершины v:



1). v – корень $[0,f_v]$

$p[v]=v$

$r[v]\nearrow$

$r[p[v]]=r[v]\nearrow$

$|SR(v)|=|S(v)|\geq 2^{r[v]}$

(т.к. get пока не влияет на $|S(v)|$)

2). v – не корень $[f_v+1,+\infty)$

$p[v]\neq v$

$r[v]$ больше не меняется

$r[p[v]]>r[v]$

$r[p[v]]\nearrow$

$|\underset{\text{не меняется с момента } f_v}{SR(v)}|\geq 2^{r[v]}$ – не меняется с момента f_v

f_v – последний момент времени, когда v ещё корень

$$SR(v)=\begin{cases} S(v), \text{ если } v \text{ – корень} \\ S(v) \text{ в момент времени } f_v, \text{ если } v \text{ не корень}\end{cases}$$

Lm. $\forall k$ найдётся $\le \frac{n}{2^k}$ вершин ранга k.

Д-во. $v\neq u$.

$r[v]=r[u]=k$. Покажем, что $SR(v)\cap SR(u)=\varnothing$.

a). v,u – корни. $S(v)\cap S(u)=\varnothing$

в). v – корень, u – не корень $\left[\begin{array}{l} u \text{ в другом дереве} \Rightarrow SR(v)\cap SR(u)=\varnothing \\ u \text{ – потомок } v \Rightarrow r[u]<r[v] \text{ (?!)}\end{array}\right.$

c). v – не корень, u – не корень

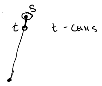
Н.У.О., $f_v>f_u$

Момент времени f_v: v – корень, u – не корень

С этого момента $SR(v)$, $SR(u)$ не менялись

Но в этот момент по в). $SR(v)\cap SR(u)=\varnothing$.

Оценка времени работы: a get-ов:



$$\sum_{i=1}^{a} Time(get_i) = \underbrace{a}_{\text{вызовы от корня}} + \underbrace{a}_{\text{вызовы от сыновей корня}} + \text{обр. остальных рёбер}$$

Зам. Если вызвали $\underset{\substack{x \text{ – не корень} \\ \text{и не сын корня}}}{get(x)} \Rightarrow r[p[x]]\uparrow$.

$r[s]>r[p[x]]$

Остальные рёбра:

— $r[p[v]]\gg r[v]$ крутое ребро

$r[p[v]]$ лежит в другой корзине, чем $r[v]$

Таких рёбер на пути get-а $\le \log^* n$.

Все ранги $\le n \Rightarrow$ все они лежат в первых $\log^* n$ корзинах

z	Корзины:
1	$\{1\}$
2	$\{2\}$
4	$\{3,4\}$
16	$\{5,6,\dots,16\}$
2^{16}	$\{17,18,\dots,2^{16}=65536\}$ $\# \le \frac{n}{2^{17}}+\frac{n}{2^{18}}+\dots+\frac{n}{2^{2^{16}}}$
$2^{2^{16}}$	$\{2^{16}+1,\dots,2^{2^{16}}\}$

2^k $\{k+1,\dots,2^k\}$

# вершин в корзине

$\frac{n}{2^{k+1}}+\frac{n}{2^{k+2}}+\dots+\frac{n}{2^{2^k}}\le\frac{n}{2^k}$

$z=2^k$

— пологие рёбра

$r[p[x]]\uparrow$ на ≥ 1 при каждом проходе по ребру.

Таких проходов будет $\le z$, определяемое корзиной $r[x]$

$$\sum_{i=1}^{a} \#\text{проходов по пологим рёбрам} = \sum_{\text{корзинам}} (\#\text{вершин в этой корзине})\cdot(z \text{ этой корзины})$$

$$= \sum_{\text{корз.}} \frac{n}{2^k}\cdot 2^k = \#\text{корзин}\cdot n \le \log^* n\cdot n$$

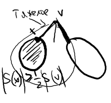
$$\sum_{i=1}^{a} Time(get_i) = \underbrace{a}_{\text{корень}} + \underbrace{a}_{\text{сыны корня}} + \underbrace{a\cdot\log^* n}_{\text{крутые}} + \underbrace{n\log^* n}_{\text{пологие}} = O((a+n)\log^* n)$$

$a\geq n$ среднее время get: $O\left(\frac{a\log^* n}{a}\right)=O(\log^* n)$.

Rem. Можно оценить обр. ф-цей Аккермана $\alpha^{-1}(n)$

Краскал: $O(Sort+n+m\log^* n)=O(Sort+m\log^* n)$

$O(Sort+m\alpha^{-1}(n))$



s
#легких ребер ?
v