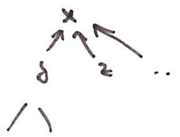
UNION-FIND: + úvod Kruskal

- uložení rozkladu množiny na třídy
- MAKE-SET(x) přidám {x} do systému
- UNION(x,y) sjednotím množiny obsahující x a y
- FIND(x) ... najdu množinu obsahující x

množina := strom



struktura pro vrcholové položky

parent: // pointer na rodiče, kořen je sám vrcholem
rank: // odhad hloubky stromu

prvky v universu identifikujeme s vrcholy ve stromech (například při uložení v poli)

FIND(x) ⇒ vrátí reprezentanta množiny (kořen stromu)

```
if x.parent ≠ x
    x.parent = FIND(x.parent)
    return x.parent
else return x
```

UNION(x,y)

```
x ← FIND(x)
y ← FIND(y)
if x==y return
if ~~rank~~ x.rank < y.rank      ⎫
    swap x,y                     ⎪
y.parent ← x                     ⎬ operace LINK
if x.rank == y.rank              ⎪
    x.rank ← x.rank + 1          ⎭
```

Amortizovaná složitost ~ skoro lineární vzhledem k počtu operací FIND, UNION,

tj. m ... počet operací celkem
n ... počet prvků

složitost $\Theta(m \cdot \alpha(n))$, kde α n je inverzní ackermanova fce.

Function iteration of a Function f:

$F: \mathbb{N} \to \mathbb{N}$

$$f^{(0)}(x) = x$$

$$f^{(i)}(x) = f(f^{(i-1)}(x)) \quad \text{for } i \geq 1 \quad \Rightarrow \quad F^{(1)}(x) = f(x)$$

Function $A_k(j)$ $\quad k \geq 0, \; j \geq 1$

$A_k(j)$ is strictly increasing

$$A_k(j) = \begin{cases} j+1 & \text{pro } k = 0 \\ A_{k-1}^{(j+1)}(j) & \text{pro } k \geq 1 \end{cases}$$

Argument k : level Funkce

$$A_0^{(i)}(j) = j + i$$

indukce: base-case: $A_0^{(0)}(j) = j$

hypotéza $A_0^{(i-1)}(j) = j + (i-1)$

$$A_0^{(i)}(j) = A_0(j + (i-1)) = j + i$$

$$\boxed{A_1(j) = A_0^{(j+1)}(j) = j + (j+1) = 2j + 1}$$

$$A_1^{(i)}(j) = 2^i (j+1) - 1$$

indukce:

$$A_1^{(0)}(j) = 2^0 (j+1) - 1 = j$$

definice $f^{(0)}$

hypotéza: $A_1^{(i-1)}(j) = 2^{i-1}(j+1) - 1$

$$A_1^{(i)}(j) = A_1(2^{i-1}(j+1) - 1) = 2^i(j+1) - 2 + 1 = 2^i(j+1) - 1$$

$$\boxed{A_2(j) = A_1^{(j+1)}(j) = 2^{j+1}(j+1) - 1}$$

AKERMANOVA FCE

~~k=1~~ j = 1 $A_0(1) = 2$ $A_1(1) = 3$ $A_2(1) = 7$

$$
\begin{aligned}
A_3(1) &= A_2^{(2)}(1) \\
&= A_2(A_2(1)) \\
&= A_2(7) \\
&= 2^8 \cdot 8 - 1 = 2047
\end{aligned}
$$

$$
\begin{aligned}
A_4(1) &= A_3^{(2)}(1) \\
&= A_3(A_3(1)) \\
&= A_3(2047) \\
&= A_2^{(2048)}(2047) \gg A_2(2047) = 2^{2048} \cdot 2048 - 1 > 2^{2048} \gg \boxed{10^{80}}
\end{aligned}
$$

počet atomů ve vesmíru

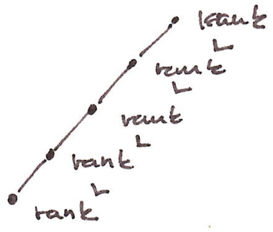
$$\alpha(n) = \min\{k \mid A_k(1) \geq n\} \Rightarrow \boxed{A_{\alpha(n)}(1) \geq n}$$ to vidíme z definice.

$$
\alpha(n) = \begin{cases}
0 & 0 \leq n \leq 2 \\
1 & n = 3 \\
2 & 4 \leq n \leq 7 \\
3 & 8 \leq n \leq 2047 \\
4 & 2048 \leq n \leq A_4(1)
\end{cases}
$$

$A_{k-1}(1) \leq \quad \leq A_k(1)$ pro obecné k

TOTO ÷ už uděláme na příště ?

VLASTNOSTI RANKU vrcholů



a) rank < n − 1

b) rank s počtem operací neklesá

~~m~~ n prvků (n-krát MAKE-SET)

m operací FIND-SET, LINK (UNION ÷ 2x FIND SET + LINK, ÷ to lineární vztah)

Potenciálová fce

$$\Phi_q = \sum_x \phi_q(x) \qquad // \, x \text{ ÷ vrchol}$$

$$\phi_q(x) = \begin{cases} \alpha(n) \cdot x.rank & (x \text{ ÷ kořen, nebo } x.rank = 0) \\ \text{viz dále} & (\text{tj. v def dále } x.rank \geq 1 \text{ a } x \text{ není kořen}) \end{cases}$$

Jak to musím ustanovit, abych nepoužil rodičovský rank

co popisuje A_K?

$x.parent$

vztah ranků

def: $level(x) = \max \{ k \mid x.parent.rank \geq A_k(x.rank) \}$

Claim: $0 \leq level(x) < \alpha(n)$,

Proof:

1) $x.parent.rank \geq x.rank + 1 = A_0(x.rank)$ $\quad (0 \leq level(x))$

2) $A_{\alpha(n)}(x.rank) \geq A_{\alpha(n)}(1) \geq n > x.parent.rank$ $\quad (level(x) < \alpha(n))$

rank < n − 1

($\alpha(n)$ už ÷ moc, protože $A_{\alpha(n)}(x.rank) > x.parent.rank$)

∎

def: $iter(x) = \max \{ i \mid x.parent.rank \geq A^{(i)}_{level(x)}(x.rank) \}$

Claim: $1 \leq iter(x) \leq x.rank$

Proof:

1) $x.parent.rank \geq A_{level(x)}(x.rank) = A^{(1)}_{level(x)}(x.rank)$ $\quad$ tj. $iter(x) \geq 1$

[definice level(x)]

2) $A^{(x.rank+1)}_{level(x)}(x.rank) = A_{level(x)+1}(x.rank) > x.parent.rank$ $\quad$ a $x.parent.rank > x.rank$

↳ definice level

∎

definujeme potenciál pro ~~vnitřní~~ nekořenové uzly s rankem ≥ 1:

$$\phi_q(x) = (\alpha(n) - level(x)) \cdot \underset{x.rank}{\cancel{rank[x]}} - iter(x)$$

$\phi_q(x) = \alpha(x) \cdot x.rank$ ji pro root a $x.rank = 0$.

Lemma: Pro každý uzel x a q platí

$$0 \leq \phi_q(x) < \alpha(n) \cdot rank[x]$$

Důkaz: pro x kořen nebo $x.rank = 0$ je to jasné

Do vzorečku pro $\phi_q(x)$ dosadíme z dolního a horního omezení pro level(x) a iter(x) [pozor ~~u iter~~ antitoně, protože je v definici $\phi_q(x)$ odečítaného]

$$\phi_q(x) \geq (\alpha(n) - (\alpha(n) - 1)) * x.rank - \cancel{rank}\, x.rank$$
$$= x.rank - x.rank = 0$$

[$0 \leq level(x) \leq \alpha(n)$
$1 \leq iter(x) \leq x.rank$]

zespodu, dosazujeme max

$$\phi_q(x) \leq (\alpha(n) - 0) \cdot x.rank - 1$$
$$= \alpha(n) * x.rank - 1$$
$$< \alpha(n) * x.rank$$

shora, dosazujeme min

∎

Lemma: [pohled] x ... not a root, q-tá operace: LINK, FIND-SET

Obecně: $\phi_q(x) \leq \phi_{q-1}(x)$

~~Pokud~~ Operace změní level(x) nebo iter(x) a $x.rank \geq 1$, ~~potom~~ pk.: $\phi_q(x) \leq \phi_{q-1}(x) - 1$.

Důkaz:

q-tá operace nemění ~~rank pro~~ $x.rank$, protože x není kořen, stejně tak předpokládejme $\alpha(n)$ se nezmění. (neděláme MAKESET)

Všechny MAKESET můžeme udělat na začátku

$$x.rank = 0 \Rightarrow \phi_q(x) = \phi_{q-1}(x) = 0$$

level(x) se monotónně zvětšují [viz podstatu operací FIND-SET a LINK = vzdáli, mezi x.rank a x.parent.rank pouze roste (z pohledu x).] s počtem operací.

A) level(x) se nezmění:

$\Rightarrow$ iter(x) se buď zvětší nebo zůstane stejný

$\hookrightarrow \phi_q(x) = \phi_{q-1}(x)$

iter(x) se zvětší aspoň o 1,

tj. $\phi_q(x) \le \phi_{q-1}(x) - 1$ (tj. odečítáme jedna)

B) level(x) se zvětší (nejméně o 1)

výraz $(\alpha(n) - level(x)) * x.rank$ se změní nejméně o x.rank.

~~to~~ iter(x) klesne nejvýše o x.rank − 1 (z hodnoty x.rank na hodnotu 1, viz omezení).

celkem tedy potenciál klesne nejméně o 1

$\boxtimes$

Lemma: Amortizovaná složitost MAKE-SET je $O(1)$

Důkaz: Hned vidíme, že rank přidaného uzlu je 0.
Nic jiného se nezmění. Změna pot. je 0, akt. slož. je $O(1)$.

$\boxtimes$

Lemma: Amortizovaná složitost LINK je $O(\alpha(n))$.

Důkaz: LINK(x,y)

změny potenciálu:

$\rightarrow$ potomci y: podle lemma [pokles] jim potenciál nevzroste
(+ potomci x:) (i zbylé x a y)

$\rightarrow$ x: před linkem root: ~~def~~ $\phi_{q-1}(x) = \alpha(n) * x.rank$ [pokud byl x.rank = 0, pak je i po LINKu a potenciál je 0]

předp. x.rank ≥ 1, potom

$\phi_q(x) = (\alpha(n) - level(x)) * x.rank - \underbrace{iter(x)}_{iter \ge 1}$.

$0 \le level(x)$ ~~[illegible]~~

$\Downarrow$

$\le \alpha(n) \cdot x.rank - 1$

$< \alpha(n) * x.rank = \phi_{q-1}(x)$ [protože x byl před linkem root]

$\Rightarrow$ uzlu x ~~rank~~ ~~pokles~~. potenciál klesl.

→ y: $\phi_{q-1}(y) = \alpha(n) * y.rank$ | y zůstane kořenem.

a) LINK nechá stejný y.rank → potom je $\phi_q(y) = \phi_{q-1}(y)$

b) LINK zvedne rank o 1 → potom $\phi_q(y) = \phi_{q-1}(y) + \alpha(n)$

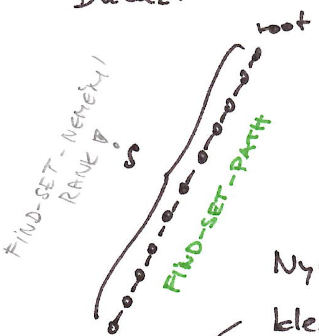
skutečná!

Potenciál se tedy operací LINK zvedne o max $\alpha(n)$. Amort. složit. je $O(1) + \alpha(n)$.
$= O(\alpha(n))$.

□

Lemma: Amortizovaná složitost FIND-SET je $O(\alpha(n))$

Důkaz:



skutečná složitost je $O(s)$

→ žádnému uzlu na cestě se nezvýší potenciál
kořen má stejný rank + Lemma [pokles] pro ostatní uzly

Nyní dokážeme, že $\max(0, s - (\alpha(n)+2))$ uzlům klesne potenciál.

Pak je změna potenciálu $- 0$ … $O(s) + 0$ (am. složitost je), ale to bude když $\boxed{s = O(\alpha(n))}$ / už dále

am. složitost: $O(s) - (s - (\alpha(n)+2)) = O(s) - s + O(\alpha(n)) = O(\alpha(n))$

scale the potential up.

vybereme:
x … uzel na FIND-SET-PATH, x není root, x.rank > 0 a
existuje uzel y, y != root, level(x) = level(y) [level() ≤ $\alpha(n)$ z definice (rank < n−1)]
a y je předchůdce x (⇒ je na FIND-SET-PATH).

takových uzlů existuje ≥ $s - (\alpha(n)+2)$

vynechat: ~~Krok~~ (první uzel: rank 0, poslední uzel: root, $\alpha(n)$ mezi uzly (last for which level = k for $k = 0, 1, 2, \alpha(n)-1$)
one = highest one.

$level(y) = level(x) = k$, $iter(x) = i$

Před FIND-SET máme:

$x.parent.rank \geq A_k^{(i)}(x.rank)$
$y.parent.rank \geq A_k(y.rank)$ } z definic level a iter

$y.rank \geq x.rank$. // vlastnost pořadí rank.

Máme tak

$$y.\text{parent}.\text{rank} \geq A_k(y.\text{rank})$$

$$\geq A_k(\cancel{y.\text{ra}}\ x.\text{parent}.\text{rank}) \longleftarrow (x.\text{parent}.\text{rank} \leq y.\text{rank})$$

$$\geq A_k(A_k^{(i)}(x.\text{rank}))$$

pretože $\boxed{x.\text{parent}.\text{rank} \geq A_k^{(i)}(x.\text{rank})}$

$$= A_k^{(i+1)}(\cancel{\text{ran}}\ x.\text{rank})$$

TJ: $y.\text{parent}.\text{rank} \geq A_k^{(i+1)}(x.\text{rank})$ (*)

Po FIND-SET majú x a y stejného rodiče, tj. $x.\text{parent}.\text{rank} = y.\text{parent}.\text{rank}$

Ale vďaka (*) vidíme, že se muselo zvednout iter(x) nebo level(x). [po vrchol x]

Použitím Lemma [pokles] dokážeme, že potenciál x klesol min o 1.

[Toto je kľúčové: posunutím vrcholu nahor ve FIND-SET zmenšujeme potenciál]

■