Union-find : compression de chemin — union à niveaux — 2 optimisations ↗
(Union-link)

Propriété

1) sur les rangs

a) $\forall x : x \neq \pi(x) \Rightarrow r(x) < r(\pi(x))$ (invariant)

b) $r(x) = 0$ si la création de $\{x\}$, puis $r(x)$ ↗ tant que x est une racine (i.e $x = \pi(x)$), puis ne change plus.

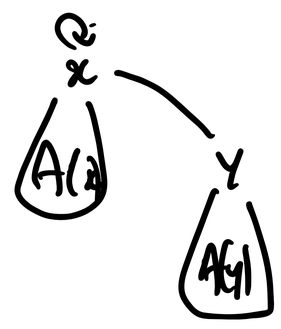
c) $r(\pi(x))$ est croissant au cours du temps

2) Pour toute racine x d'un arbre de la forêt, l'arbre $A(x)$ contient au moins $2^{r(x)}$ noeuds..

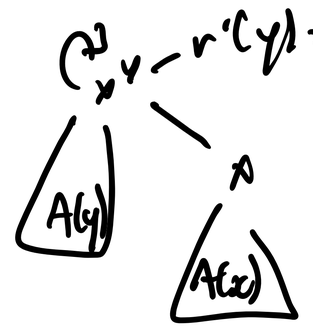
@@ invariant

Link (x, y) 1) $|A(x)| \geq 2^{r(x)}$
$|A(y)| \geq 2^{r(y)}$

$r(x) > r(y)$



2) $r(x) = r(y)$
$|A(x)| \geq 2^{r(x)}$
$|A(y)| \geq 2^{r(y)} = 2^{r(x)}$
donc $|A(x)| + |A(y)| \geq 2^{r(x)} + 2^{r(y)}$



3) Pendant toute l'exécution d'un algorithme utilisant m opérations (MakeSet, Find, Link) dont n MakeSet

pour tout $k \geq 0$, il y a au plus $\frac{n}{2^k}$ noeuds qui auront le rang k

Preuve fixé

- supposons qu'à chaque fois que le rang d'un noeud x passe à la valeur k on colore tous les noeuds de $A(x)$ avec une couleur "rang k"
- Un tel coloriage concerne au moins 2^k noeuds
- Un noeud ne sera jamais colorié 2 fois en couleur "rang k" (i.e d'une racine

le rang stricht plus grand de h
le rang des fils est fixé

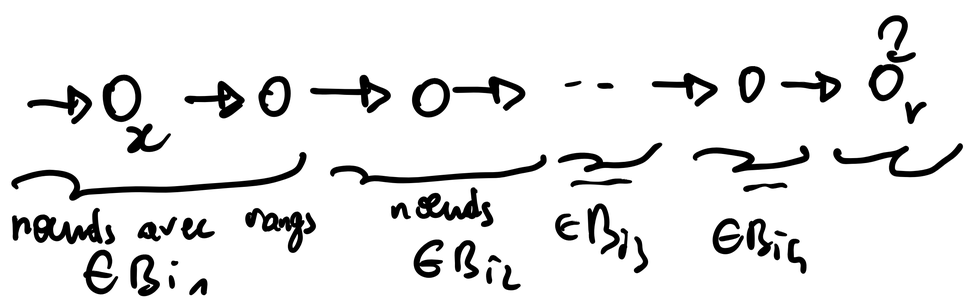
Donc plus $\frac{n}{2^h}$ noeuds recevant le rang h.

Cout Make-set → O(n)

Link → O(1)

Find ?

Il faut estimer Find :



On va compter les changements de blocs + la dernière transition vers r → $(m \log^* n)$ (nbr find, nbr de blocs)

+

les transitions internes à un bloc → $(n \log^* n)$

Définir un bloc, tous les rangs r de même $\log^* r$

rang	1	2	3 4	5 .. 16	17 .. 65536	65537 .. 2^{65536}	..
	0	1	2	3	4	5	6

bloc i : $r \in [\underbrace{2^{2^{i-1}}+1}_{b_{i-1}+1}; \underbrace{2^{2^{i-2}}}_{b_i}]$ jusqu'à $b_i = 2^{b_{i-1}}$

→ • le nbr total de blocs : $\underbrace{\log^*(r_{max})}_{\log n} + 1 = \log^*(n)$

• dans le bloc i, il y a moins de $\frac{n}{b_i}$ noeuds

nb noeud bloc i $\leq \frac{n}{2^{b_{i-1}+1}} + \frac{n}{2^{b_{i-1}+2}} + \dots + \frac{n}{2^{b_i}} \leq \frac{n}{2^{b_{i-1}}} \underbrace{\left(\frac{1}{2} + \frac{1}{4} \dots \frac{1}{2^{b_i}}\right)}_{\leq 1} \leq \frac{n}{b_i}$

• les transitions interne à un bloc

on a $\frac{n}{b'}$ noeuds dans B_i => Coût total de ces transitions internes aux bloc $\leq \frac{n}{b'}$ x taille

Et comme il y a $\log^* n$ blocs, on a en total : $O(n \log^* n)$

Théorème : m opérations MakeSet/Find/Link dont n MakeSet ont un complexité total en $m \log^* n$ dans le pire des cas.

// Méthode agrégative - ~~n~~ complexité

// de gauche

correction exam → suivre le cours (pas de piège suivre les exercices).

+ analyse amortie.

{ Comment résoudre ces prob dutch ?

-> changer de prob

-> algo d'approximation (ch sac à dos)

-> algo probabiliste // pas vu en cours // algo pic nique lancer pièce // problème philosophe