

Algorismes i Estructures de Dades II

Conjunts Disjunts

Francesc J. Ferri

Dept. d'Informàtica. Universitat de València

Febrer 2008

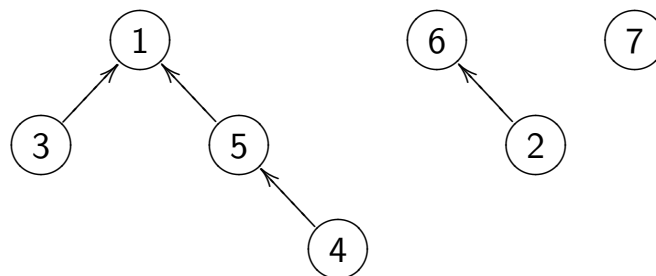


El TAD Conjunt Disjunt

Serveix per representar conjunts de conjunts de manera que cada element del tipus base només pot estar en un d'ells. Cas particular: particions.

Les operacions bàsiques són: **unir** i **trobar**

Representació arborescent: $\{1, 3, 4, 5\}$, $\{2, 6\}$, $\{7\}$



Els elements 1, 6 i 7 són els representants dels conjunts de què formen part.



Operacions

trobar(x)

donat un element x torna el (representant del) conjunt a què pertany

Cost millor: $O(1)$

Cost pitjor: $O(n)$

unir(i,j)

donats 2 representants dona com a resultat el representant de la unió (un dels dos arbres passa a ser subarbre de l'altre).

Cost: $O(1)$ en qualsevol cas



Unió per rang

Rang de x , $r(x)$

altura del subarbre que penja de x .

($r(x)$ és exactament el cost de trobar(x))

Unió per rang

el arbre amb menys rang es converteix en subarbre del que té més rang

Conseqüència:

El rang total no creix, a no ser que els dos rangs siguin iguals i aleshores creix en 1.

Es pot demostrar que el rang màxim és $O(\lg n)$ si hi ha n elements en total. Exercici!



Conjunts disjunts. Resum

Sense unió per rang:

	millor	pitjor
unir	1	1
trobar	1	n

Amb unió per rang:

	millor	pitjor
unir	1	1
trobar	1	$\lg n$

Exercici:

demostrar que en cap dels dos casos pot existir un cost amortitzat menor.



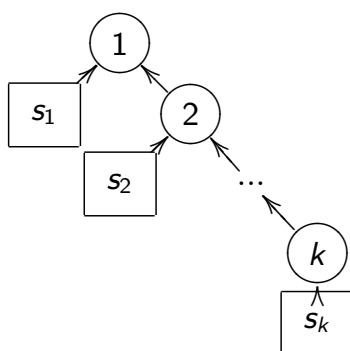
Compressió de camins

Observació

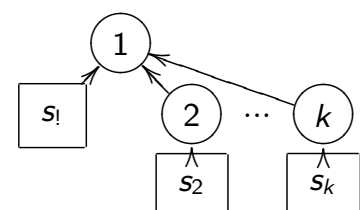
la operació trobar no modifica l'“estat”.

Idea

trobar implica recórrer un camí des d'un node fins l'arrel. Es pot fer que tots els subarbres recorreguts passen a ser subarbres de la mateixa arrel



trobar(k)



Compressió de camins. Anàlisi amortitzada

Totes les operacions unir (per rang) tenen cost “real” constant (encara que el rang ja no siga l'altura del subarbre!)

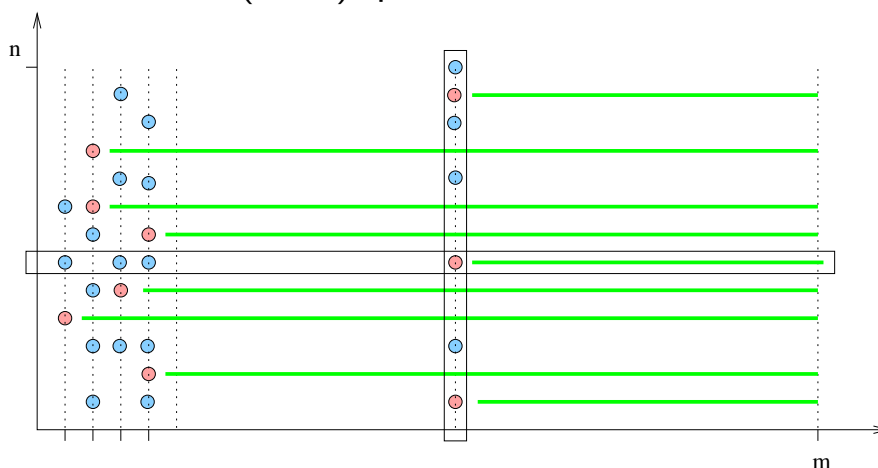
Les operacions trobar(x_0) que recorren els nodes $x_0, x_1, \dots, x_k$ tenen cost “real” $k \leq c \lg n$.

Quin és el cost de m operacions trobar sobre els mateixos n elements (encara que hi hagen operacions unir intercalades)?



Compressió de camins. Anàlisi amortitzada

Suposem m operacions **trobar** i representem el cost en passos de cada una assignant-lo al l'element (node) que es visita en cada moment.



Sabem que en cada columna hi ha com a molt $\lg n$. Es pot fitar la suma total per alguna cosa inferior a $m \lg n$?



Compressió de camins. Anàlisi amortitzada

Siga $x_0, x_1, \dots, x_k$ la seqüència de nodes visitats en una operació trobar.

Marcarem (amb roig/blau) cada un dels $k + 1$ passos en funció del valor dels seus rangs corresponents, $r_0, r_1, \dots, r_k$, de la següent manera:

Si $\lg \lg r_i < \lg \lg r_{i+1}$ (o $i = k$) pas roig
Si $\lg \lg r_i = \lg \lg r_{i+1}$ pas blau

Ara sumarem els passos rojos per columnes i els passos blaus per files.



Costs roig/blau

Cost roig

El nombre màxim de passos rojos en qualsevol columna ve donat per $1 + \lg \lg n$.

Tant el rang d'un node com la diferència entre aquest rang i el de son pare, són no decreixents en m

Per tant, els passos blaus sobre un mateix node només es poden donar abans de l'únic pas roig que es pot assignar a eixe node.

I com a molt hi haurà $2^{2^j} - 2^{2^{j-1}}$ (nodes pitjors) per a algun j



Cost blau

Es pot demostrar que el nombre de nodes que tenen un valor de rang r en un moment donat és com a molt $\frac{n}{2^r}$

Per tant, el màxim nombre de nodes pitjors per a un determinat j serà:

$$\sum_{2^{2^{j-1}}+1}^{2^{2^j}} \frac{n}{2^r} < \frac{n}{2^{(2^{2^j-1})}} \sum_{\ell=0}^{\infty} \frac{1}{2^\ell} < \frac{n}{2 \cdot 2^{2^j}} 2 = \frac{n}{2^{2^j}}$$

I sumant per a tot j

$$\sum_{j=0}^{\lg \lg n - 1} \frac{n}{2^{2^j}} (2^{2^j} - 2^{2^{j-1}}) < \sum_{j=0}^{\lg \lg n - 1} \frac{n}{2^{2^j}} (2^{2^j}) = n \lg \lg n$$



Cost total

El cost de les m operacions trobar

$$(m + n) \lg \lg n$$

El cost amortitzat (si $m \gg n$)

$$\lg \lg n$$

	millor	pitjor	amortitzat
unir	1	1	1
trobar	1	$\lg n$	$\lg \lg n$

Pregunta:

Què passaria si repetirem l'anàlisi utilitzant la funció $\lg \lg \lg n$?

