Advanced Union-Find

Tarjan's Analysis

Algorithms: Design and Analysis, Part II

Tarjan's Bound

Theorem: [Tarjan '75] with Union by Rank and path compression, m Union + Find operations take $O(m\,\alpha(n))$ time, where $\alpha(n)$ is the inverse Ackermann function.

Acknowledgment: Kozen, "Design and Analysis of Algorithms".

Building Blocks of Hopcroft-Ullman Analysis

Block #1: Rank Lemma (at most $n/2^r$ objects of rank r).

Block #2: path compression $\Rightarrow$ if x's parent pointer updated from p to p', then $\text{rank}(p') \geq \text{rank}(p) + 1$

New idea: stronger version of building block #2. In most cases, rank of new parent much bigger than rank of old parent (not just by 1).

Quantifying Rank Gaps

Definition: Consider a non-root object x (so rank[x] fixed forevermore).

Define $\delta(x) =$ max value of k such that $\text{rank}[\text{parent}(x)] \geq A_k(\text{rank}[x])$.

(note $\delta(x)$ only goes up over time)

Examples: always have $\delta(x) \geq 0$.

$\delta(x) \geq 1 \iff \text{rank}[\text{parent}(x)] \geq 2 \cdot \text{rank}[x]$

$\delta(x) \geq 2 \iff \text{rank}[\text{parent}(x)] \geq \text{rank}[x] \cdot 2^{\text{rank}[x]}$

Note: for all objects x with $\text{rank}[x] \geq 2$, then $\delta(x) \leq \alpha(n)$.

[since $A_{\alpha(n)}(2) \geq n$]

Good and Bad Objects

Definition: An object x is bad if all of the following hold:

(1) x is not a root

(2) parent(x) is not a root

(3) rank(x) ≥ 2

(4) x has an ancestor y with $\delta(y) = \delta(x)$

x is good otherwise.

Quiz

Question: What is the maximum number of good objects on an object-root path?

A) $\Theta(1)$

B) $\Theta(\alpha(n))$

C) $\Theta(\log^* n)$

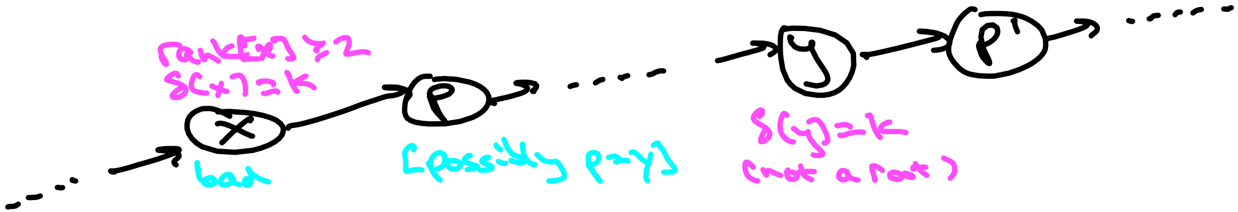
D) $\Theta(\log n)$

$\leq$ 1 root + 1 child of root
+ 1 object with rank 0
+ 1 object with rank 1
+ 1 object with $\delta(x) = k$
for each $k = 0, 1, 2, \ldots, \alpha(n)$

Proof of Tarjan's Bound

Upshot: Total work of m operations = $\underbrace{O(m\alpha(n))}_{\text{visits to good objects}}$ + $\underbrace{\text{total \# of visits to bad objects}}_{\text{will show } = O(n\alpha(n))}$

Main argument: Suppose a FIND operation visits a bad object x:



Path compression: x's new parent will be p' or even higher.

$\Rightarrow \text{rank}(x\text{'s new parent}) \underset{\text{ranks only go up}}{\geq} \text{rank}(p') \underset{\text{since } s(y)=k}{\geq} A_k(\text{rank}(y)) \underset{\text{ranks only go up}}{\geq} A_k(\text{rank}[p])$

Tim Roughgarden

Proof of Tarjan's Bound II

Point: path compression (at least) applies the A_k function to rank(x's parent).

Consequence: if $r = \text{rank}[x]$ (≥ 2), then after r such pointer updates we have

defn of Ackermann function

$$\text{rank}(x\text{'s parent}) \geq \underbrace{(A_k \circ \cdots \circ A_k)}_{r \text{ times}}(r) = A_{k+1}(r)$$

Thus: while x is bad, every r visits increases $\delta(x)$

$\Rightarrow \leq r \cdot \alpha(n)$ visits to x while it's bad

Proof of Tarjan's Bound III

Recall: Total work of m operations is

$\underbrace{O(m\alpha(n))}_{\text{visits to good objects}}$ + total # of visits to bad objects

$$\leq \sum_{\text{objects } x} \text{rank}[x] \cdot \alpha(n)$$

$$= \alpha(n) \sum_{r \geq 0} r \cdot \left(\begin{array}{l}\text{\# of objects}\\ \text{with rank } r\end{array}\right)$$

$\leq \frac{n}{2^r}$ for each r, by the Rank Lemma

$$= n \cdot \alpha(n) \sum_{r \geq 0} \frac{r}{2^r}$$

$= O(1)$

$$= O(n\alpha(n)).$$

QED!

Epilogue

"This is probably the first and maybe the only existing example of a simple algorithm with a very complicated running time....I conjecture that there is *no* linear-time method, and that the algorithm considered here is optimal to within a constant factor."

-Tarjan, "Efficiency of a Good But Not Linear Set Union Algorithm", Journal of the ACM, 1975.

Conjecture proved by [Fredman/Saks '89]!