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 Ackerman function

Acknowledgement: 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 rank(p')$\geq$rank(p)$+1$

New idea: Stronger version of building block #2. In most cases, rank of new parent <u>much</u> 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 rank[parent[x]$\geq A_k$(rank[x])

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

Examples: Always have $\delta(x) \geq 0$
$\delta(x) \geq 1 \iff$ rank[parent[x]]$\geq$ 2 rank[x]
$\delta(x) \geq 2 \iff$ rank[parent[x]]$\geq$ rank[x] $2^{\mathrm{rank}[x]}$

Note: For all objects x with 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) $\Theta(\log^* n)$

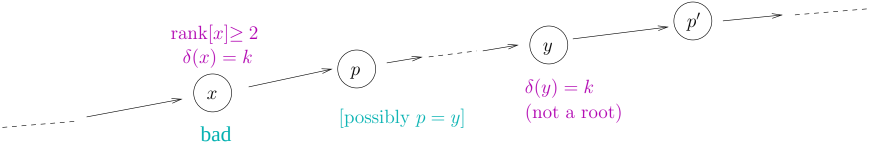
D) $\Theta(\log n)$

$\leq$ 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 $= O(m\alpha(n))$ (visits to good objects)$+$ total # of visits to bad objects (will show is $O(n\alpha(n))$)

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



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

$\Rightarrow$ rank[x's new parent] $\geq$ rank[p'] $\geq$ A_k(rank[y]) $\geq$ A_k(rank[p])

ranks only go up   since $\delta(y) = k$ ranks only go up

Proof of Tarjan's Bound II

Point: Path compression (at least) applies the A_k function to rank[x's parent]

Consequence: If r=rank[x] (≥ 2), then after r such pointer updates we have

$$\text{rank}[x\text{'s parent}] \geq (A_k \circ \ldots \text{ r times } \ldots \circ A_k)(r) = A_{k+1}(r)$$

Definition of Ackermann function

Thus: While x is bad, every r visits increases $\delta(x)$
$\Rightarrow \leq r\alpha(n)$ visits to x while it's bad

Proof of Tarjan's Bound III

Recall: Total work of m operations is $O(m\alpha(n))$ (visits to good objects) $+$ total $\#$ of visits to bad objects.

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

$$= \alpha(n) \sum_{r \geq 0} r \ \boxed{(\# \text{ of objects with rank } r)}$$

$\leq n/2^r$ for each r by the Rank Lemma

$$= n\alpha(n) \boxed{\sum_{r \geq 0} r/2^r} \longrightarrow = O(1)$$

$$= O(n\alpha(n)). \qquad \text{QED!}$$

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 Non Linear Set Union Algorithm", Journal of the ACM, 1975.

Conjecture proved by [Fredman/Saks 89]!