Algorithms: Design and Analysis, Part II

Advanced Union-Find

Path Compression: The Hopcroft-Ullman Analysis

Hopcroft-Ullman Theorem

Theorem: [Hopcroft-Ullman 73] with Union by Rank and path compression, m Union+Find operations take $O(m \log^* n)$ time, where

$\log^* n$ = the number of times you need to apply log to n before the result is ≤ 1

[will focus on interesting case where $m = \Omega(n)$]

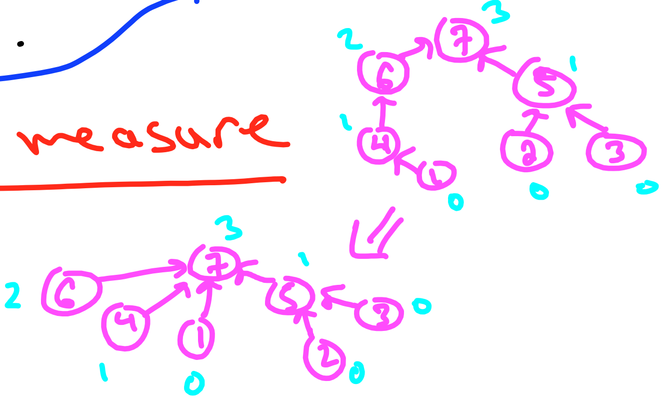
Measuring Progress

Intuition: installing shortcuts should significantly speed up subsequent FINDs & UNIONs.

Question: how to track this progress and quantify the benefit?

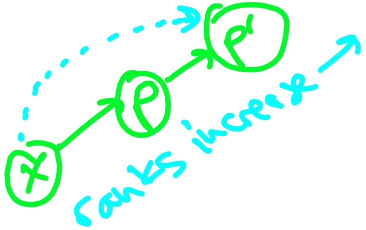
Idea: Consider a non-root object x. → recall: $rank[x]$ frozen forevermore

Progress measure: $rank[parent(x)] - rank[x]$.



Path compression increases this progress measure

If x has old parent p, new parent $p' \neq p$, then $rank[p'] > rank[p]$.



Proof Setup

Rank blocks: $\{0\}, \{1\}, \{2,3,4\}, \{5, \ldots, 2^4\}, \{17, 18, \ldots, 2^{16}\},$

$\{65537, \ldots, 2^{65536}\}, \ldots, \{\ldots, n\}.$

16 65536

Note: there are $O(\log^* n)$ different rank blocks.

Semantics: traversal $x \to$ parent(x) is "fast progress" $\Longleftrightarrow$ rank[parent(x)] in larger block than rank[x]

Definition: At a given point in time, call object x good if

(1) x or x's parent is a root OR

(2) rank[parent(x)] in larger block than rank(x)

x is bad otherwise

Proof of Hopcroft-Ullman

Point: every FIND visits only $O(\log^* n)$ good nodes
[2 + # of rank blocks = $O(\log^* n)$].

visits to good objects

Upshot: total work done during m operations = $O(m \log^* n)$ + total # of visits to bad nodes

need to bound globally by separate argument

Consider: a rank block $\{k+1, k+2, \ldots, 2^k\}$.

Note: when a bad node x is visited  , its parent is changed to one with strictly larger rank. $\Rightarrow$ can only happen 2^k times before x becomes good (forevermore)

Proof of Hopcroft-Ullman II

Total work: $O(m \log^* n) + O(\underbrace{\text{\# visits to bad nodes}}_{\le n \text{ for each of } O(\log^* n) \text{ rank blocks}})$

Consider: a rank block $\{k+1, k+2, \ldots, 2^k\}$.

Last slide: for each object x with final rank in this block, # visits to x while x is bad is $\le 2^k$.

Rank lemma: total number of objects x with final rank in this rank block is $\sum_{i=k+1}^{2^k} n/2^i \le n/2^k$

$\le n$ visits to bad objects in this rank block

Recall: only $O(\log^* n)$ rank blocks.

Total work: $O((m+n) \log^* n)$.

QED!

Tim Roughgarden