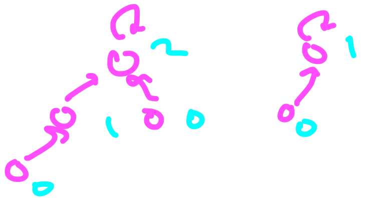
Algorithms: Design and Analysis, Part II

# Advanced Union-Find

# Union by Rank - Analysis

# Properties of Ranks



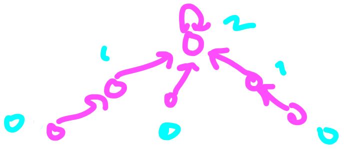
**Recall:** lazy Unions.

note $\max_x \text{rank}[x] \approx$ worst-case running time of FIND

**Invariant (for now):** rank[x] = max # of hops from a leaf to x.

**Union by Rank:** make old root with smaller rank child of the root with the larger rank.

[choose new root arbitrarily in case of tie, and add 1 to its rank]



**Immediate from Invariant / Rank Maintenance**

(1) For all objects x, rank[x] only goes up over time

(2) Only ranks of roots can go up [once x a non-root, rank[x] frozen forevermore]

(3) ranks strictly increase along a path to the root

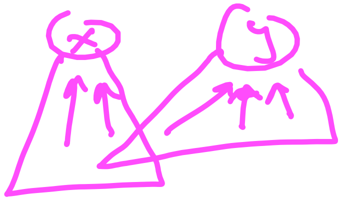
# Rank Lemma

**Rank Lemma:** Consider an arbitrary sequence of UNION (+FIND) operations. For every $r \in \{0,1,2,\dots,\}$, there are at most $n/2^r$ objects with rank $r$.

**Corollary:** max rank always $\leq \log_2 n$

**Corollary:** worst-case running time of FIND, UNION is $O(\log n)$. [with Union by Rank]

# Proof of Rank Lemma




Claim 1: if x, y have the same rank r, then their subtrees are disjoint.
↳ objects from which can reach x, y

Claim 2: the subtree of a rank-r object has size $\geq 2^r$.

[note Claim 1 + Claim 2 imply the Rank Lemma]

Proof of Claim 1: Will show contrapositive. Suppose subtrees of x, y have object z in common. $\Rightarrow \exists$ paths $z \rightsquigarrow x$, $z \rightsquigarrow y$
$\Rightarrow$ one of x, y is an ancestor of the other
$\Rightarrow$ the ancestor has strictly larger rank [by property (3)]
qed. (claim 1)

Tim Roughgarden

# Proof of Claim 2

rank $r$ $\Rightarrow$ subtree size $\geq 2^r$

By induction on the number of Union operations.

Base case: initially all ranks $= 0$, all subtree sizes $= 1$.

Inductive step: nothing to prove unless the rank of some object changes (subtree sizes only go up).

Interesting case: Union$(x,y)$, with $s_1 = \text{FIND}(x)$, $s_2 = \text{FIND}(y)$, and $\text{rank}[s_1] = \text{rank}[s_2] = r$. $\Rightarrow$ $s_2$'s new rank $= r+1$

$\Rightarrow$ $s_2$'s new subtree size $=$ $s_2$'s old subtree size $+$ $s_1$'s old subtree size $\geq 2^{(r+1)}$.

each at least $2^r$ by the inductive hypothesis

QED!