

COP 3503 9/8/26

⑥ Next Time: Final Project Intro

① TODAY: Disjoint Sets

Section of Course - Advanced Data Structures

* I'm doing: 2-4 Trees, R-B Trees, ⁺ Binary Index Tree

Not doing: Splay Trees, Skip List Intro to

Look @ Comen Leiserson Rivest Stein Algorithms
→ quite a few more.

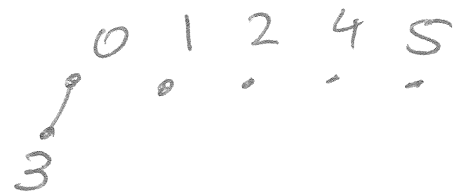
Start w/ separate sets

$\{0\}, \{1\}, \{2\}, \dots, \{n-1\}$

Union(a, b) - find set w/a
find set w/b
merge into one set

Union(0, 3)

$\{0, 3\}, \{1\}, \{2\}, \{4\}, \{5\}$

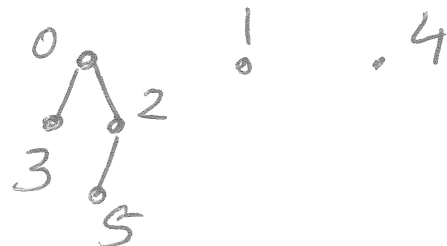


~~Find~~ find(3) - return root of tree that 3 is in.

Union(2, 5) $\{0, 3\}, \{1\}, \{2, 5\}, \{4\}$



Union(3, 5) $\{0, 2, 3, 5\}$
 $\{1\}$
 $\{4\}$

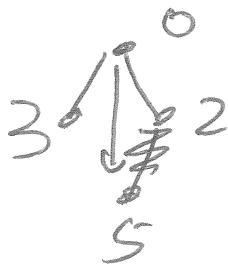


	0	1	2	3	4	5
parent	0	1	3	0	4	2

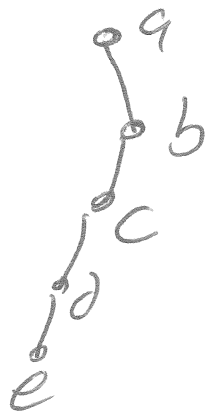
Goal: to do fast

Run-times: based on height of tallest tree.

Path Compression - Run times are based on the inverse Ackerman function



find(5) → find(2) →
find(0) → 0



find(e) → it tells me
a is root why not
reassign parent[e] = a?

return parent[e] = find(parent[e])