Disjoint Set Data Structures

- **n items**: $\{a, b, c\}$, $\{d, e\}$, $\{a, b\}$, $\{b, c\}$, $\{c, d\}$, $\{d, e\}$

- **MakeSet(x)**: do this n times with a, b, c, d, e

- **Union(x, y)**: x, y are members $\{a, b\}$, $\{c\}$, $\{d, e\}$

- **FindSet(x)**: returns representative
  
  return same thing if called twice.
Make-Set(x): See above

Union(x,y):

Find-Set(x): \[ \text{Assumes pointer to } x \]

n: Number of elements in all sets

m: Number of operations

\[ \# \text{ MakeSet ops: } n \]
\[ \# \text{ Union ops: } \leq n-1 \]
\[ \# \text{ FindSet ops: } \geq 0 \]
\[ m \geq n \]
n MakeSets

\{a\} \{b\} \{c\} \{d\} \{e\} \quad \text{Find Set}(a) = \text{Find Set}(c)

\{a, c\} \{b\} \{d\} \{e\}
\{a, c\} \{b\} \{d, e\}
\{a, b, c\} \{d, e\}
\{a, b, c, d, e\}

\checkmark \quad \text{Find Set}(b) = \text{Find Set}(c)

\{a, b, c, d, e\}
Linked List Representation

MakeSet(x) \quad A[i].head \rightarrow \boxed{x} \rightarrow \ll 0(1)\ll

Union(x, y) \quad \text{Combine the two} \quad 0(n)\quad

FindSet(x) \quad \text{Follow rep pointer} \quad 0(1)\quad

Union(\text{efg, e} \ldots e^3) \quad 1 + 2 + 3 + \ldots + n-1 = O(n^2) \quad \text{total}

\frac{O(n^2)}{n} = O(n) \quad \text{am.}
$\Theta(n)$ per operation? Can we do better?

Each obj has a rep ptr.

- 3

\[ \delta 4 : 3 3 \]

1st time it's in a set of size 1

- 2

new size at least 2

\[ \frac{1}{2} \] new size at least 4

new size at least 8

\[ \vdots \]

= n \\ \text{lg n depth}

Total changes = $O(n \text{lg } n)$

\[ \frac{o(n \text{lg } n)}{n} = O(\text{lg } n) \text{ am time} \]
Disjoint Set Forests

Make Set: Makes node point to itself
Find Set: Follows ptrs up
Union: Takes root of one tree makes it point to the root of the other
Union by Rank

\[ \text{rank}_A = 3 \quad \text{rank}_B = 1 \quad \text{rank} \leq \text{rank}_A + 1 \]

Make root \( y \) smaller rank point to other root.
Path Compression

Am Time = O(\(\alpha(n)\))

\(\alpha(n) = O(!\log^* n)\)

\(\alpha(n) \leq 4\)