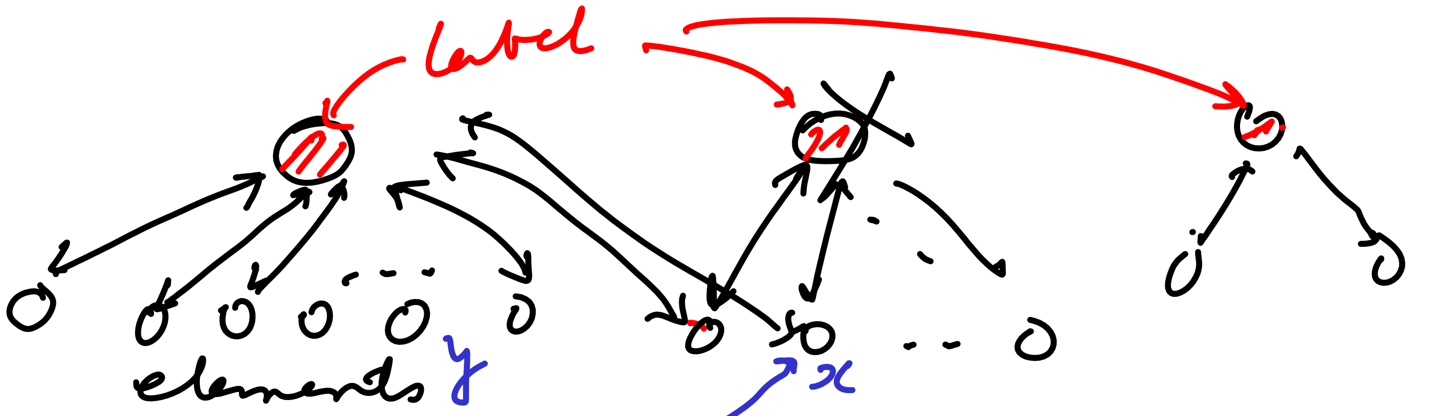
CSL 356 lect. 17, Sept 6

Data structure for Union-Find



Find(x) Read the label of x

Union$(C(x), C(y))$ when $C(x) \neq C(y)$

m Finds and n unions

$m = |E|$ $n = |V|$

What is the maxm no. of label changes for some vertex x? , say $n(x)$

Total cost of n unions $\sum_{x \in V} n(x)$

$n(x) \leq ? \log n$

$$\sum_{x \in V} n(x) \text{ is } O(n \log n)$$

So total cost of m Finds and n Unions is $O(m + n \log n)$

$O(|E| + |V|)$

The total cost of Basic greedy for MST is ordering edges by wt $O(|E| \log |E|)$

$$+ \; O(|E| + |V| \log |V|)$$

If $|E| \geq |V| \log |V|$, i.e. (slightly dense graph)

$O(|E|)$

<u>Goal</u>: Alternate Union-Find data structure with improved performance on unions

F F U F F U U . .

Union-Find-Sp~~Sk~~ will be more general not MST

How do we represent sets?

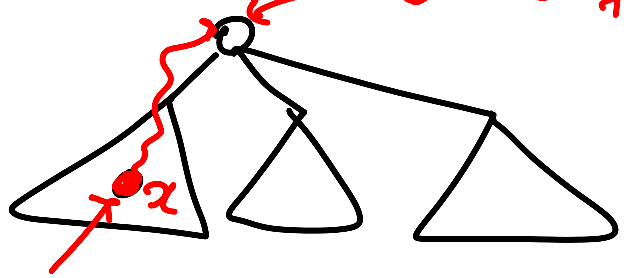
We will use trees to represent sets

base case: singleton vertices (elements)

the root has "rank" = 0

Find

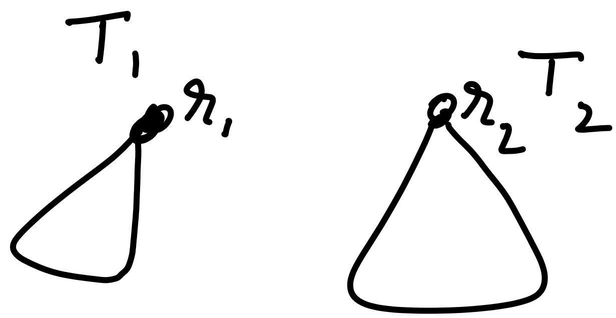
label of the set and rank $\in \mathbb{Z}^+$

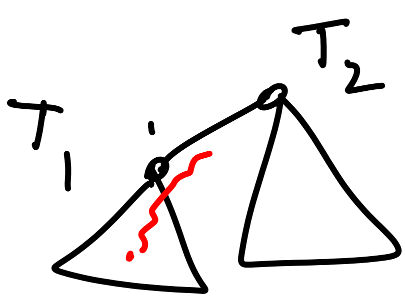


Find(x) move to the root using parent pointers and report the label

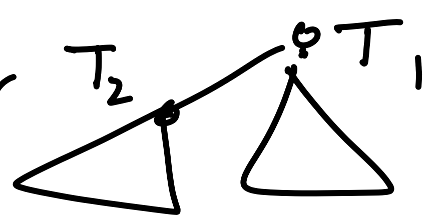
Cost: length of the path from x to root

Union (T_1, T_2)





or



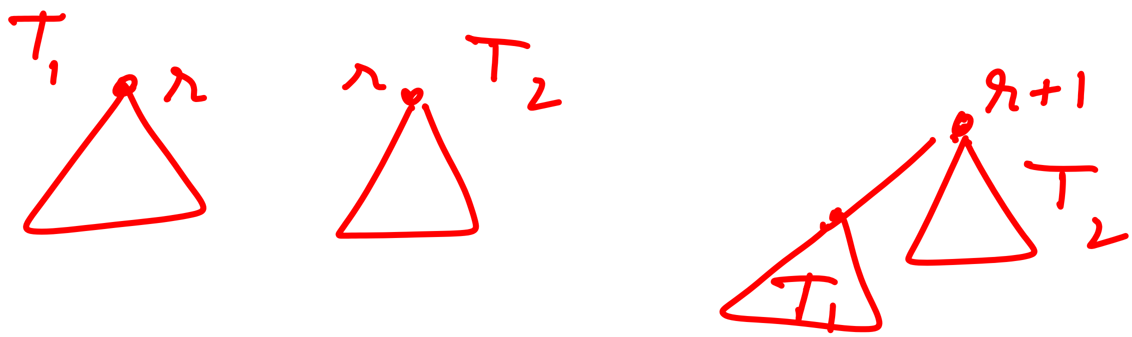
Cost $O(1)$

Union by rank heuristic

Make the root with smaller rank the child of the other root (no change in rank)

otherwise choose arbitrarily

increment the rank of the final root



Obs

1. A root node with rank r has at least 2^r descendents

(Rank is related to the maxm distance from any leaf node to the root)

Consequence is that Find takes at most $O(\log n)$ steps

Cost of m Finds and n Unions is bounded by $O(m \log n + n)$

2. The no. of nodes with rank r is bounded by

$$\frac{n}{2^r}$$

Note that once a node ceases to be a root during the course of Union Find, its rank is fixed and never changes in future (This node never becomes a root node)

3. The ranks increase monotonically in any path from leaf to root node.

Path compression heuristic

$$O((m+n) \log^* n)$$

$\log^* n$: min i, such that $\log(\log(\log \cdots (n))) \leq 2$