Analysis: Union By Rank

- Find(u): takes time proportional to u's depth in its tree.
- Show that if u's depth is h, then its tree has at least 2^h nodes.

- When union(u,v) performed, the depth of u only increases if its root becomes the child of v.
  - That only happens if v's tree is larger than u's tree.
- If u's depth grows by 1, its (new) treeSize is > 2 * oldTreeSize
  - Since v has size more than u's old tree size.
  - Each increment in depth, doubles the size of u's tree.
  - After n union operations, size is at most n**, so depth is at most log n.**

- Theorem: With Union-By-Rank, we can do find in O(log n) time and union in O(1) time (assuming roots of u, v known).
- *N-1* Unions, *O(N)* Finds: *O(N log N)* total time