Union-Find Problems

- **Union-Find Problems**
  - Elements 1, ..., n.
  - Initially, each element i is in a set by itself.

- **Operations**:
  - \( \text{FIND}(i) \): return the name of set containing i.
  - \( \text{UNION}(A, B, C) \): union elements of set A and B and call the result C (delete A=B)
  - \( \text{MAKE}(x) \): make a set on element x

**Example Application to Minimum Spanning Tree (MST)**

**MST**: Let \( G(V, E) \) be a connected, undirected graph with a cost function mapped to each edge.

A **spanning tree** is a undirected tree that connects all vertices \( V \) in the graph. The cost of a spanning tree is just the sum of the cost of its edges. The goal is to find a spanning tree of min cost in \( G \).

**Algorithm**

1. **Initialize**
   - tree edge \( \emptyset \)
   - vertex set \( \emptyset \)

2. For each vertex \( v \) do
   - insert \( v \) into vertex set

3. While \( |\text{vertex set}| > 1 \) do
   - extract edge \( (u, v) \) of lowest cost in \( E \) (edge set)
   - if \( \text{FIND}(u) \neq \text{FIND}(v) \) then
     - insert \( (u, v) \) into tree edge
     - \( \text{UNION}(\text{FIND}(u), \text{FIND}(v)) \)

4. **While**
   - return
Disjoint-set forest

- Array of roots inorder to locate the root of set \( i \)

- UNION(A, B, C)
  - Make root of \( TA \) a child of root of \( T_B \) and change the root of \( T_A \) to \( C \)

- FIND(C)
  - Following the pointer to the root

- Heuristics
"Union By Rank"

\[
\text{TA} \rightarrow \text{TB} \rightarrow \text{TS}
\]

Make the root with smallest rank point to root of large rank.

\[
\text{LINK}(x, y)
\]

If \( \text{rank}[x] \geq \text{rank}[y] \)
\[
\text{parent}[y] \rightarrow x
\]

Else
\[
\text{parent}[x] \rightarrow y
\]

\[
\begin{align*}
\text{if } \text{rank}[x] &= \text{rank}[y] + 1 \\
\text{then } \text{rank}[y] &= \text{rank}[y] + 1 \\
\text{else } &\text{No tree in the forest height greater than } O(\log n)
\end{align*}
\]