Graphs III

Tuesday, March 27, 2018 12:36 PM

Kruskal's Algorithm

- MST by merging smaller trees into bigger ones
- Graph F = set of trees (each vertex = separate tree)
- Set S = set of edges
- While S not empty and F not spanning:
    - Remove minimum e from S
    - If edge connects two trees, add to F

DisjointSet ADT

- Very good for finding and union
- Keep each subtree as a set of vertices

Quick-Find

- Check if componentId (group) is the same
- find = O(1), union = O(n)

Quick-Union

- Record parent (root parent = itself)
- Find: compare root ids by traversing upwards
- Union: update parent
- Unbalanced = O(n) for both

Balancing Trees (Weighted Union)

- Instead of using the height of tree to balance, use size
- Max height = O(log n)
- Tree only increases in height when the merging trees have the same height
- Find and Union are now O(log n)

Path Compression

- After finding root, set parent of each traversed node to the root
- Alternative: make roots point to grandparent
- Weighted union + path compression = flat trees, almost linear operations

Kruskal's Algorithm - Improved

- Sort edges, iterate through edges in order = O(E log E)
    - O(E log E) = O(E log V^2) = O(E log V)
- For E edges, find/ union = O(Eα)