Minimum Spanning Tree Algorithms
MST Algorithms

- Kruskal
- Prim
- Borouvkha

In case of unweighted graphs or just (not minimal) spanning tree use DFS or BFS tree.
Union Find Structure (UFS)

**UFS**

- Two operations
  1. find - gives representative for a set
  2. union - unites two sets

- tree structure

- find -
  1. walk up in the tree till the root is found
  2. return root as representative

- unite
  1. find both representatives
  2. link representative of smaller height to representative of greater height.

- use path compression in find operations (link all elements on the path directly to the root)

Kruskal

Kruskal - sort and unite

// ... read in edges into one array
sort(edges.begin(), edges.end());
Ufs ufs (vertices);
for (unsigned i = 0; i < edges.size(); ++i) {
    int u = ufs.find (edges[i].from);
    int v = ufs.find (edges[i].to);
    if (u != v) {
        ufs.unite (u, v);
        mst.add(u, v);
    }
}
// select an arbitrary start vertex and insert into PQ
// with weight 0 all other vertices are inserted into
// the PQ with weight infinity

while (!pq.empty()) {
    pq_pair curr = pq.top();
    pq.pop();

    for_all_edges(curr) {
        if (pq.find(curr_edge.dest) > edge_weight)
            pq.decrease_key(curr_edge.dest, edge_weight);
    }
}
MST Properties

Properties to think about

- Shortest Path Tree vs. MST
- Uniqueness
- Sequence of edge weights in MST
- in/exclusion of certain light/heavy edges