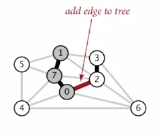
Kruskal’s Algorithm

1. Take edges, consider by weight (in order of ascending weight).
2. Add next edge to tree, so long as edge does not create a cycle.

Proposition: Kruskal’s algorithm computes the MST

Proof: Kruskal’s algorithm is a special case of the greedy MST algorithm

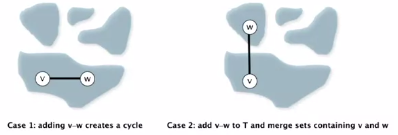


* Suppose Kruskal’s algorithm colors the edge e = v-w black
* Cut = set of vertices connected to v in tree T
* No crossing edge is black
* No crossing edge has a lower weight. Why?  
  *Haven’t considered yet, because we’re considering in increasing order of weight*

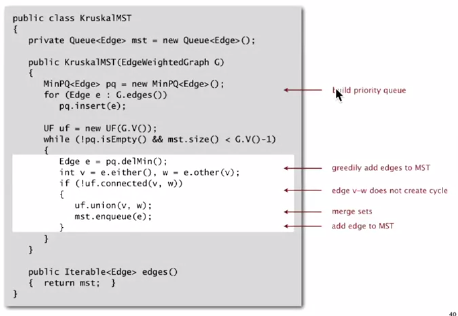
How would we know whether adding the edge would create a cycle?   
How difficult is the question to answer?

1. V (linear time) : run DFS from v. Check if w is reachable (T has at most V-1 edges)
2. Log\* V : Use union-find data structure

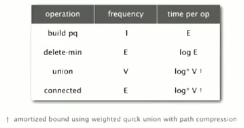
Using union-find to determine cycles:

* Maintain a set for each connected component in T
* If v and w are in same set, then adding v-w would create a cycle
* To add v-w to T, merge sets containing v and w

Kruskal’s algorithm implementation



Running time: Kruskal’s algorithm computes MST in time proportional to E log E (in the worst case).



Note: if edges are already sorted, order of growth is actually E log\* V!