6. Boruvka's Algorithm

Minimum Spanning Tree

Distinct edge weights $\implies$ MST is unique.

Proof by contradiction, assume we have two minimum spanning trees T, T', we'll prove that a pair of edges in G have the same weight.

We know each spanning tree contains an edge the other doesn't.
Let e be the minimum-weight edge in $T \backslash T'$
Let e' be the minimum-weight edge in $T' \backslash T$
WLOG, we assume $w(e) \leq w(e')$.

The subgraph $T' \cup \{e\}$ contains exactly one cycle that was created by introducing e. We then find an edge e'' that's not in T (this exists, otherwise it would imply T contains a cycle). It is possible for $e' = e''$. We know $e'' \neq e$, as $e \in T$, therefore $e'' \in T' \backslash T$
We then get

$$w(e'') \geq w(e') \geq w(e)$$

Now, if make the spanning tree $T'' = T' + e - e''$, we have

$$wt(T'') = wt(T') + wt(e) - wt(e'') \implies wt(T'') \leq wt(T')$$

But T' is a minimum spanning tree, therefore $wt(T'') = wt(T') \implies wt(e) = wt(e'')$. Hence, there exists a pair of edges which have the same weight, contradicting the fact that edge weights are distinct.

Common MST algorithms

Kruskal's - Sort the edges by their weight, and grow your forest greedily.
Prim's - Start from an arbitrary node, and grow your tree greedily.

Boruvka's Algorithm

We assume that the edges are distinct
Uses concepts similar to supernodes & superedges.

- For every vertex, pick an edge that is the edge of least weight incident on it.
- Contract all these edges to form super nodes.
- Edge between supernodes $(u, v) =$ Edge of least weight between them.
- While $\#\text{supernodes} > 1$ repeat.

When $\#\text{supernodes} = 1$, output all contraccted edges. These edges form a MST.

Correctness of Boruvka's Algorithm

Taking least edge weight in super edge is optimal

Claim - Edge of least weight incident on a vertex $\in \text{MST}$. More generally, a least weight edge of a cut belongs belongs to MST.

Proof By Exchange Argument - Assume we don't take the least edge weight of a vertex, and we instead take a heavier edge. This would form a spanning tree, but we can just instead remove that edge and take the lighter edge (need to explain better).

We can instead say that the least weight edge in a super edge $\in \text{MST}$, stating the above claim in a more generalized manner.

No triangle of edges can be contracted

Claim - There is no such case that in an iteration, the edges chosen form a triangle (assuming edges are distinct).
Proof -
Assume vertices $1, 2, 3$, where the edge weights are

$$(1, 2) = a$$
$$(1, 3) = b$$
$$(2, 3) = c$$

Say WLOG that vertex 1 picks edge of weight a. That inherently implies that $a < b$. As we want to form a triangle, it would mean that vertex 2 doesn't choose the same edge, and instead chooses c, hence $c < a$

Therefore

$$c < a < b$$

But if vertex 3 chooses b to form the triangle, it would contradict the relation $c < b$, hence a triangle can't exist.

Decrease in number of super nodes per iteration

Claim - $\#\text{supernodes}$ decreases by at least fraction of $\frac{1}{2}$ in each iteration.
Proof - Contracting an edge decreases the number of super nodes by 1. If we have n vertices, there can be at max $(n-1)$ unique edges picked. In the worst case, each edge is picked twice, giving us a minimum possible count of $\frac{n}{2}$, hence reducing the number of super nodes by half in each iteration.

Rewriting the problem as a recursive problem

$\text{MST}(G) = \text{MST}(G_1) \cup \{\text{edges picked in 1}\}$
$\text{MST}(G_2) = \text{MST}(G_3) \cup E_2$

Heaviest edge of n cycle $\notin \text{MST}$

Claim - Span of Boruvka's $\leq O(\log n)$, giving time complexity of $\leq O(E \log V)$
And contrasting Prim's and Kruskal's, this algorithm that can be run in **parallel**.