# Kruskals graphs algorithm

Algorithm for constructing trees based on an edge que.

$(V, E)$ connected graph, $c.E- > R$ cost function.

1. Start w/ $(V, o)$ subgraph consisting of all vertices of $(V, E)$ and no edges
2. List all edges in $E$ in a que so that the cost of the edges is non-decreasing in the que.
3. Then we consider edges successively from the front of the que, and determine whether or not the addition of that edge $e$ to a current subgraph will create a circuit. IF yes discard edge. Otherwise, add it to the subgraph. Continue until the que is empty

## Exmaple

See the rest here

Prove that Kruskals algorithm always yeilds a spanning tree that is minimal.

**Proposition:** let $(V, E)$ be a connected graph w/ associated cost function $c : E \to R$. Kruskcal's algorithm yeilds a spanning tree of $(V, E)$
**Proof:** Since an edge is added from the queue only if no circuit is created, we conclude the subgraph $(V, E\prime)$ produced by the algoithm bys be acylical (i.e contains no circurits). To prove $(V, E\prime)$ is a spanning tree of $(V, E)$, we must show $(V, E\prime)$ is connected. Assume not, Then $(V, E\prime)$ has components $(V, E\prime), (V_2, E_2\prime), .....(V_m, E_m\prime) m > z(V, E)$ is connected $\to$ $Eedgee_ijeE$ for $i < j, j < m, ifjs.tadding$ edge $$e_ij$ connects $(V_i, E_j\prime)$ and $(V_j, E_j\prime)$, but edge $e_ij$ could that have possibly created a circuit considered in the queue \\\$\to (V, E\prime)$
cannot have more than one connected component $\to (V, E\prime)$ is connected. QED$

**Proposition** let $(V, E)$ be a connected graph with associated cost function $c.E \to R$. A minimal spanning tree is yielded. **Proof** We alredy showed in the previous proposition that kruskals algorithm yields a spanning tree. Now we show the tree is minimal.
$w.s.tc.E \to R$, let $(v, E\prime)$ be the spaning tree generated. if $(V, E\prime) = (V, E)$ ie if the original connected graph is a tree, then there is nothing to prove. Assume $(V, E)$ contains some circuit. Let all the edges be $e_1, e_2, e_3 .... e_m$. That we used. s.t $c(e_i) < c(e_j) \exists i < j < m$. In other words $c(e_i) < c(e_m)$ The algorithm chooses the lowest cost edges from $e_1 \to e_m$ such that the subgrah is a spanning tree of (V,E). Thoug if $(V, E\prime\prime)$ is of the spanning tree then $c(E\prime) < c(E\prime\prime)$

**DEF** let $(V, E)$ be a connected grap with the associated cost function $c.E \to R$ let $(V, E\prime)$ be the minimal spanning tree produced by the algoithm is called a kruskal tree. **NB** If two or more edges have the same cost, then we can reshuffle them in the queue used to determine the tree, therefore, the kruskal tree might not be unique.

In the example we used to illistrate the algorithm we see this series at work.

NOTE

- Queue basically equates to $(V, E) \to E$ all be it with additional edges that could possibly equate to circuit
- Each edge should be added in order of the queue, and then discarded if need be based on whether or not it leads to the creation of a circuit.