2.3

Definition

A **weighted graph** G is a graph alongside a function $w : E(G) \to \mathbb{R}^+$.

If G is a weighted graph and $H \subseteq G$, then, $w(H) := \sum_{e \in E(H)} w(e)$.

A **minimum weight spanning tree** (or MWST) is a spanning tree T such that $w(T)$ is minimized among all possible spanning trees. In other words, $w(T) \le w(T')\ \forall T' \subseteq G$ where T' is a spanning tree.

Kruskal's Algorithm

Input Weighted graph G with n vertices

Output A MWST, T^* if G is connected, otherwise a message "G is not connected"

Step 1 Create a list of edges, L_E, in order from smallest weight to largest weight. Start T^* with no edges but all vertices of G.

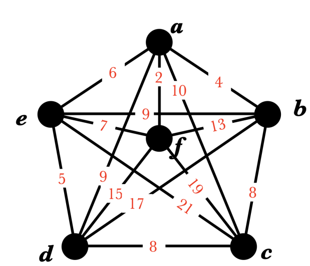
Step 2 If the number of edges in T^* is strictly less than $n-1$ and if there are still edges in L_E, examine the first edge in L_E (i.e., the edge with smallest weight), $e = \{a, b\}$

- Substep 2.1 If a and b are in different components of T^*, add e to T^* and remove e from L_E. Return to Step 2.
- Substep 2.2 If a and b are in the same component, remove e from L_E and do not add to T^*. Return to Step 2.

Step 3 If the number of edges in T^* is $n-1$, then output T^*, which is the MWST for G. Otherwise, the number of edges in T^* is strictly less than $n-1$ and G was not connected.

Example

We will find a MWST for the following graph:



First, we create the following table of all the edges in G.

Edge	Cost
af	2
ab	4
de	5
ae	6
ef	7
bc	8
cd	8
ad	9
be	9
ac	10
bf	13
df	15
bd	17
cf	19
ce	21

We can read the following table describing the steps in Kruskal's algorithm from left to right (i.e., we check the edge of lowest weight, then we check the components, then we select our substep).

Edge	Components Of T'	Substep to be used
af	$\{a\},\{b\},\{c\},\{d\},\{e\},\{f\}$	2.1
ab	$\{a,f\},\{b\},\{c\},\{d\},\{e\}$	2.1
de	$\{a,b,f\},\{c\},\{d\},\{e\}$	2.1
ae	$\{a,b,f\},\{c\},\{d,e\}$	2.1
ef	$\{a,b,d,e,f\},\{c\}$	2.2
bc	$\{a,b,d,e,f\},\{c\}$	2.1
—	$\{a,b,c,d,e,f\}$	—

Proof of Kruskal's Algorithm

If G is connected, then Kruskal's Algorithm produces a minimum weight spanning tree.

Let G be a connected graph, and let T_K be the output from Kruskal's algorithm on G. It is easy to check that T_K is a spanning tree, since it is acyclic and has $n(G)-1$ edges.

Suppose T^* is a MWST with largest edge intersection with T_K — in other words, $|E(T_K)\cap E(T^*)| \geq |E(T_K)\cap E(T')|$ for any other MWST T'.

If $T^* = T_K$, then we are done, since T_K is assumed to be a minimum weight spanning tree. Otherwise, assume toward contradiction that $T^* \neq T_K$. Let e be the first edge chosen by Kruskal's algorithm that is not in T^*. Then, by a previous result, $\exists e' \in E(T^*) - E(T_K)$ such that $T' = T^* + e - e'$ is a spanning tree.

We are assuming, however, that Kruskal's algorithm would choose e over e'. Let $e_1, \dots, e_k$ be edges in $E(T_K)$ before e. Since e_i was selected before e, we know that $e_i \in E(T^*)$ for each $i \in [k]$.

Let $G_k = (V, \{e_1, \dots, e_k\})$. we are assuming that Kruskal's algorithm would not choose e' for two reasons:

- If e' shows up before e, then e' would connect two vertices in $G_j = (V, \{e_1, \dots, e_j\})$. Since $G_j + e' \subseteq T^*$, then T^* contains a cycle and isn't a spanning tree.
- If e' shows up after e in L_E, then $w(e') \geq w(e)$, meaning $w(T') \leq w(T^*)$, meaning that $w(T_K) \leq w(T^*)$, implying that T_K is of a lower weight than T^*.