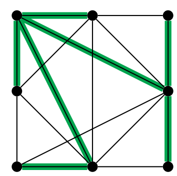
Notes on Spanning Trees

Scribe: Sophia Azaraev Lecturer/Editor: Chris Eppolito

22 April 2020

Definition. Given a graph G, a *spanning tree* in G is a subgraph $T \leq G$ such that T is a tree and $V(T) = V(G)$ vertices in T are vertices of G.

Example 1. We give a spanning tree T of the graph G below; the edges of the T are colored green.



Proposition. *Every connected graph has a spanning tree.*

Proof. Let G be a connected graph. We proceed by induction on $\#E(G)$.

Base Case: If $\#E(G) = 0$, then $G = K_1$ is its own spanning tree.

Inductive Step: Suppose the result holds for all graphs with k edges for some $k \in \mathbb{N}$, and assume G has $k+1$ edges. If G has an edge $e \in E(G)$ such that $G \setminus e$ is connected, then $G \setminus e$ has k edges, so it has a spanning tree $T \leq G \setminus e \leq G$ by the induction hypothesis. Otherwise, for every edge e of G, we have $G \setminus e$ is disconnected; thus G is minimally connected, so G is a tree (and thus its own spanning tree).

Hence the original statement holds by induction. □

Next we will look at a small example of spanning trees.

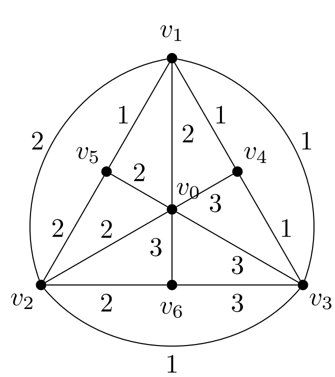
Example 2. We will find all subsets of $S = \{1, 5, 7, 17, 27, 42\}$ having sum 49 by building a spanning tree in the graph G with $V(G) = \left\{T \subseteq S : \sum_{t \in T} t \leq 49\right\}$. To build our spanning tree we start with vertex $\emptyset$ and add elements to the set, choosing "next largest" elements at each stage. We depict this process below.

$TREEPICTURE$!

Oftentimes in applications we want a spanning tree minimizing some objective function on connections.

Definition. An *(edge)-weighted graph* is a pair (G, w) where G is a graph and $w \colon E(G) \to \mathbb{R}$ is a function. The *weight* of a subgraph $H \leq G$ is $w(H) \coloneqq \sum_{e \in E(H)} w(e)$.

Example 3. The following is an example of a weighted graph; weights are given as labels on edges.



Problem. Given a weighted connected graph (G, w), compute a minimum weight spanning tree $T \leq G$.

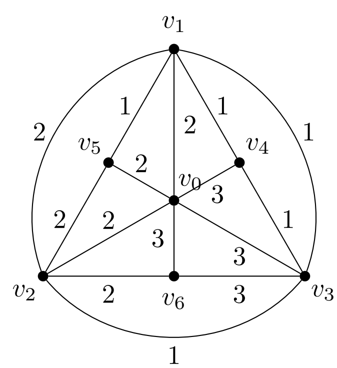
Algorithm (Prim's Algorithm)**.** Let (G, w) be a weighted connected graph.

1. Let $T_1 = (\{v_1\}, \emptyset)$ for any vertex $v_1 \in V(G)$.
2. While $V(T_k) \neq V(G)$:
   (a) Choose a minimum weight edge $e_k \in E(G) \setminus E(T_k)$ with exactly one end in $V(T_k)$.
   (b) Let $T_{k+1} \coloneqq T_k \cup \{e_k\}$ and continue.
3. Output T_n.

Note that there are choices involved in Prim's algorithm; the output might depend on these choices, but the amazing fact is that the weight of the result is always the same!

Proposition. *Prim's Algorithm always results in a minimum weight spanning tree.*

Example 4. We apply Prim's Algorithm to obtain a minimum weight spanning tree in our weighed graph.



The step-by-step results are outlined in green below.