

# Minimum Spanning Trees and $d$ -Heaps

Jonathan Turner

January 9, 2013

This note is adapted from *Data Structures and Network Algorithms* by Tarjan.

A *spanning tree* in a connected, undirected graph is a subtree of the graph that includes all its vertices. In an edge-weighted graph, the objective of the *minimum spanning tree problem* is to find a spanning tree for which the sum of the edge weights is as small as possible.

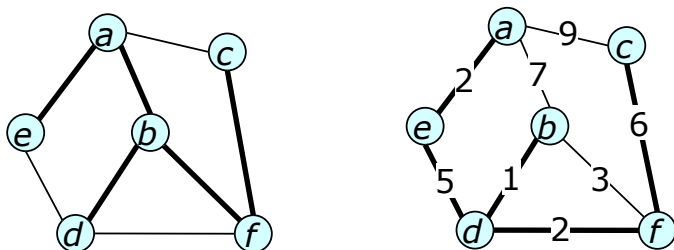


Figure 1: Example of a minimum spanning tree

This problem arises frequently in applications and often appears as a sub-problem within other optimization problems.

The *greedy method* is a general approach to finding a minimum spanning tree, and is the basis for a variety of different specific algorithms. Before describing the greedy method, we need a definition. A *cut* in a graph is a division of the vertices into two subsets  $X$  and  $X'$ . We say an edge *crosses the cut* if one of its endpoints is in  $X$  and the other in  $X'$ .

The greedy method assigns colors to the edges of a graph (*blue* or *red*) by repeatedly applying one of the two following rules.

- *Blue rule.* Select a cut with no blue edges, but at least one uncolored edge. Select a minimum cost uncolored edge crossing the cut and color it blue.

- *Red rule.* Select a simple cycle with no red edges and at least one uncolored edge. Select a maximum cost uncolored edge on the cycle and color it red.

The method terminates when neither of the rules can be applied. At this point, all the edges will be colored, and the blue edges will define a minimum spanning tree.

Notice that the greedy method leaves a number of things unspecified. For example, it doesn't say when to apply the blue rule, nor what cut to select when using the blue rule. We'll see shortly that no matter what choices we make for these things, the greedy method yields a minimum spanning tree. This establishes the correctness of any specific algorithm that implements the general greedy method, meaning that we don't need to prove correctness separately for any such algorithm. Defining algorithms in terms of a general method like this is extremely useful. It allows us to see more clearly what is essential and what is incidental, and it allows us to understand the fundamental similarities and differences among different algorithms.

We establish the correctness of the greedy method by showing that it maintains the following invariant.

*Color invariant.* There is a minimum spanning tree containing all of the blue edges and none of the red edges.

Suppose that the invariant is true before a step that uses the blue rule. Let  $e = x, y$  be the selected edge, and let  $T = (V, F)$  be an MST containing all the blue edges (and no red ones) before the step. If  $e \in F$ , then after the step,  $T$  still contains all the blue edges and no red edges. If  $e \notin F$ , then after the step, there is some other edge  $e'$  on the simple path from  $x$  to  $y$  in  $T$  that is also in the cut selected by the blue rule (see Figure 2). Since the cut

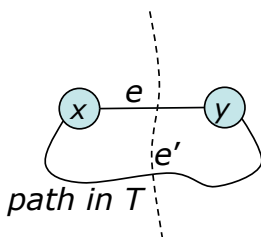


Figure 2: Blue rule maintains color invariant

contains no blue edges  $e$  is not blue. Consequently,  $T' = (V, F \cup \{e\} - \{e'\})$  is a spanning tree. Since  $cost(e) \leq cost(e')$ ,  $T'$  is a minimum spanning tree that contains all the blue edges and none of the red ones.

Suppose that the invariant is true before a step that uses the red rule. Let  $e = x, y$  be the selected edge and let  $T = (V, F)$  be a minimum spanning tree that contains none of the red edges and all the blue edges before the step. If  $e \notin F$  then  $T$  contains no red edges (and all the blue edges) after the step. If  $e \in F$ , then removing  $e$  from  $F$  splits  $T$  into subtrees  $T_1$  and  $T_2$ . There is some edge  $e'$  that is not in  $T$ , on the cycle selected by the red rule that joins a vertex in  $T_1$  to a vertex in  $T_2$  (see Figure 3). Since  $e'$  is on the

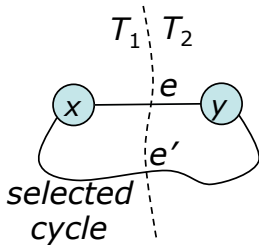


Figure 3: Red rule maintains color invariant

cycle, it is not red. Hence,  $T' = (V, F \cup \{e'\} - \{e\})$  is a spanning tree. Since  $cost(e) \geq cost(e')$ ,  $T'$  is a minimum spanning tree.  $T'$  contains no red edges and all the blue edges, so this step maintains the color invariant. To see that all edges are colored, suppose that at some point  $e = u, v$  remains uncolored. If  $u$  and  $v$  are connected by a blue path then that path plus  $e$  forms a cycle that the red rule can be applied to. If  $u$  and  $v$  are not connected by a blue path, then there is a cut crossed by  $e$  that the blue rule can be applied to. This gives us the following theorem.

**Theorem 1** *The greedy method colors all edges of a connected graph and maintains color invariant.*