University of North Carolina at Greensboro
CSC 330: Advanced Data Structures
Prof. Stephen R. Tate

Handout 11
November 15, 2016

Proof of Correctness for Prim's Algorithm

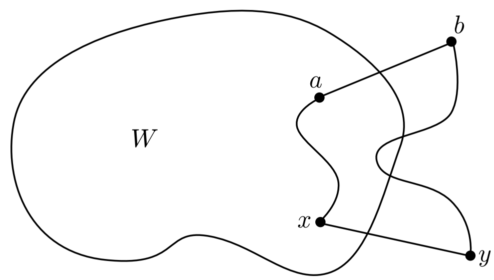
This handout refers to Prim's algorithm as given in the Hein *Discrete Structures* book.

Theorem 1 *If S is the spanning tree selected by Prim's algorithm for input graph $G = (V, E)$, then S is a minimum-weight spanning tree for G.*

PROOF: The proof is by contradiction, so assume that S is not minimum weight. Let $ES = (e_1, e_2, \cdots, e_{n-1})$ be the sequence of edges chosen (in this order) by Prim's algorithm, and let U be a minimum-weight spanning tree that contains edges from the longest possible prefix of sequence ES.

Let $e_i = \{x, y\}$ be the first edge added to S by Prim's algorithm that is not in U, and let W be the set of vertices immediately before $\{x, y\}$ is selected. Notice that it follows that U contains edges $e_1, e_2, \cdots, e_{i-1}$ but not edge e_i.

There must be a path $x \rightsquigarrow y$ in U, so let $\{a, b\}$ be the first edge on this path with one endpoint (a) inside W, and the other endpoint (b) outside W, as in the following picture:



Define the set of edges $T = U + \{\{x, y\}\} - \{\{a, b\}\}$, and notice that T is a spanning tree for graph G. Consider the three possible cases for the weights of edges $\{x, y\}$ and $\{a, b\}$:

T is better than U

Case 1, $w(\{a, b\}) > w(\{x, y\})$: In this case, in creating T we have added an edge that has smaller weight then the one we removed, and so $w(T) < w(U)$. However, this is impossible, since U is a minimum-weight spanning tree.

T & U could still be the same at e_i

Case 2, $w(\{a, b\}) = w(\{x, y\})$: In this case $w(T) = w(U)$, so T is also a minimum spanning tree. Furthermore, since Prim's algorithm hasn't selected edge $\{a, b\}$ yet, that edge cannot be one of $e_1, e_2, \cdots, e_{i-1}$. This implies that T contains edges $e_1, e_2, \cdots, e_i$, which is a longer prefix of ES than U contains. This contradicts the definition of tree U.

(x,y) is not the smallest across the cut.

Case 3, $w(\{a, b\}) < w(\{x, y\})$: In this case, since the weight of edge $\{a, b\}$ is smaller, Prim's algorithm will select $\{a, b\}$ at this step. This contradicts the definition of edge $\{x, y\}$.

Since all possible cases lead to contradictions, our original assumption (that S is not minimum-weight) must be invalid. This proves the theorem. □