

Chapter 1

HW

Proof. We can start by proving the Bottleneck Cycle property. Let T^* be a minimum spanning tree. Suppose to the contrary that there exists an edge $e \in T^*$ where e is the smallest edge in the cycle. If we remove edge e , then T^* breaks into two disconnected trees. Consider $T^* - e + f$. This increases the bottleneck of T^* as $w_e < w_f$. $T^* - e + f$ is a spanning tree because a cycle exists. Therefore, the smallest edge in a cycle must not be in T^* .

Now suppose we have the optimal graph OPT that contains every optimal path in G for all $u, v \in G$. We can construct this by adding every vertex in G and every optimal path between every pair of vertices in G . By the Bottleneck Cycle property, we know that for every cycle the smallest edge can be removed in OPT , so there cannot exist any cycles in G . It must also be connected as there is a path from every vertex to another. Therefore, OPT must exist and must be a tree.

An algorithm for constructing this tree is by starting with all the edges and removing the smallest weight edge without disconnecting the graph. By the Bottleneck Cycle property, removing the smallest edge in the cycle is valid to create OPT , and if the edge is not part of a cycle, it will disconnect the graph. The runtime of this algorithm is $O(m \log n)$ because the implementation will use a priority queue. \square