Intro to Algorithms, COMP-160, Homework #12
Benjamin Tanen, 04/29/2016

2. (a) **Given a connected graph G with distinct weights, explain how to construct a spanning tree T with maximum sum of weights.**

In order to make a maximum spanning tree T for our graph G, we could use a slightly altered version of Prim's algorithm. In Prim's original algorithm, as we build up our minimum spanning tree, we consider all of the edges that connect our MST to the nodes we haven't yet marked and pick the edge e with the smallest weight w.

We can make our alteration by instead picking the edge e with the *largest* weight w. By doing this, we essentially take the inverse of Prim's algorithm by processing the largest edges first.

As we run this algorithm, we will encounter some edge that span between two nodes already in our tree T. In Prim's original algorithm, if we encounter such an edge, we can ensure that there is already a path between these two nodes that minimizes the weight. With our alternate strategy, we can again ensure there is a path that exists, but this time, this path allows for the maximized weight in our tree T.

This algorithm takes the same time as Prim's original algorithm since the alteration takes the same time (find the largest edge as opposed to the smallest edge) so our new algorithm takes $O(E \log V)$

(b) **Show that for any two vertices x, y in G the following holds:**
Every path from x to y in G contains an edge with a weight that is no greater than the smallest weight in the path from x to y in T. In other words, T maximizes the *weakest connecting link*, for every pair of vertices.

Consider any path p_1 from x to y in G. Also consider the path p_2 from x to y in T, where e_{small} is the smallest edge in p_2. Ultimately, we are attempting to show there is an edge e in p_1 such that $w(e) \leq w(e_{small})$.

First, let us consider the scenario where p_1 is a path entirely of edges in T. We thus know that $p_1 = p_2$. Thus the smallest edge e_{small} in p_2 must be in p_1. Therefore, we know there exists some edge e (namely $e = e_{small}$) in p_1 such that $w(e) = w(e_{small}) \leq w(e_{small})$.

Alternatively, we should consider the scenario when p_1 is not entirely made up of edges in T. We can attempt to show this via a proof by contradiction.

Let's assume the smallest edge e in p_1 has a larger weight than that of e_{small} (i.e. $w(e) > w(e_{small})$). Let's also make a cut to our G into G_1 and G_2 where the cut splits both e and e_{small}. We should hold the heaviest ways to traverse from G_1 to G_2 in our MST T (by definition of our maximal spanning tree). However, since $e > e_{small}$, we can traverse from G_1 to G_2 using e, which is heavier than e_{small} and not yet in our MST. Thus, we could improve our MST by swapping out e_{small} for

e. However, since we know that we already have the MST T in the begining, it is a contradiction to be able to improve the MST by swapping out one of the edges. Thus, we can see that the smallest edge e in p_1 cannot have a greater weight than that of e_{small} ($w(e) \not> w(e_{small})$).

Thus, for the smallest edge e in p_1, $w(e) \leq w(e_{small})$ so there exists a valid e in p_1 where $w(e) \leq w(e_{small})$.