Algorithms
Computer Science 140 & Mathematics 168
Spring 2012
Homework 7a
Due Thursday, March 1

1. **[20 Points] A More General Cut Theorem for MSTs!** In class, we explored some greedy algorithms for the minimum spanning tree (MST) problem when all edge weights are distinct. In particular, Dr. Sheldon showed us how to greedily select an edge to add to our set T so that T is always a subset of a MST and, when T contains $n-1$ edges, T becomes a MST.

   Specifically, recall that a *cut* is defined as a partition of the vertices V of the graph into two disjoint sets, S and $V-S$. An edge is said to *cross* the cut if it has one endpoint in S and the other in $V-S$. Next, recall the "cut property" that we proved in class: if an edge e is the unique minimum weight edge crossing a cut, then it is part of every MST. (This relies on the assumption that all edge weights are distinct.)

   (a) To prove that Prim's and Kruskal's algorithms are correct *when edge weights are not necessarily distinct*, we need a slightly stronger theorem. We say that a cut *respects* the current set of edges T if no edge of T crosses the cut. Suppose that T is a subset of some MST and $S, V-S$ is a cut that respects T. Prove the following: if e is any edge of minimum weight crossing the respectful cut $S, V-S$, then e can be added to T and the resulting set T will still be a subset of *some* MST.

   (b) Explain how this new powerful theorem allows us to establish that Prim's algorithm finds a MST even when the edge weights are not distinct. (The same could be done for Kruskal's algorithm, and you might want to convince yourself of that, but we're not asking you to submit that as part of this assignment.)

2. **[15 Points] Prof. Boreal's MST Algorithm!** Professor R. Boreal of the Pasadena Institute of Technology has developed a new algorithm for the MST problem on completely connected weighted graphs (graphs in which there is an edge between every pair of vertices). It is based on the divide-and-conquer paradigm and it goes like this: Given a *completely connected graph* $G = (V, E)$ with weights on the edges (the weights are not necessarily distinct), partition the set of vertices V into two sets V_1 and V_2 whose sizes are equal (or differ by at most 1). Let E_1 be the edges incident only on vertices in V_1 and let E_2 be the edges incident only on vertices in V_2. Recursively solve the MST problem on the two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Finally, select the minimum-weight edge that crosses the cut (V_1, V_2) and use that edge to unite the two spanning trees found by the recursive calls. Professor Boreal believes that the algorithm can be proved correct by appealing to the theorem from Problem 1. Your job is either to use that theorem to prove the correctness of the algorithm *or* to explain what goes wrong when you try to apply that theorem, in which case you should also give a small counterexample where this algorithm does not find a MST.