Algorithms
Computer Science 140 & Mathematics 168
Instructor: B. Thom
Fall 2003
Homework 10b
Due on 11/11/03 (in class)

1. **[20 Points] Some Prim and Kruskal Games!** In these exercises you'll explore what happens when you know that all weights are integers in the range $[0, W]$.

   (a) **Kruskal Games!** CLRS, Exercise 23.2-4 (page 574). Explain your answers carefully.

   (b) **Prim Games!** CLRS, Exercise 23.2-5 (page 574). Explain your answers carefully. (This problem might require a bit of cleverness in the algorithm's data structures!)

2. **[25 Points] Barůvka's Algorithm!** In class we mentioned that the first algorithm for computing minimum spanning trees was published by the Czech mathematician Otakar Barůvka in 1926. The algorithm works like this:

```
E_f = {};  \* Comment:  E_f is a subset of a minimum spanning tree *\
Consider the V vertices in the graph as V connected components;
while E_f contains fewer than V-1 edges
{
   for each connected component C
   {
      Find the least weight edge (u,v) with one vertex in C and
        one vertex not in C;
      Add edge (u,v) to E_f;
   }
   Compute the new connected components;
}
return E_f \* Comment:  This is a minimum spanning tree! *\
```

   Notice that if C_1 and C_2 are two different connected components before we begin the for loop, then inside the for loop the algorithm will choose the least weight edge coming out of component C_1 and also the least weight edge coming out of C_2. The edge chosen by C_1 might join C_1 and C_2 into a new connected component, but this new connected component will not be discovered until the for loop has ended! In other words, both C_1 and C_2 will each get an opportunity to choose the least weight edges coming out of their components!

   (a) Give a counter-example that shows that Borůvka's Algorithm doesn't work!! Show your counter-example graph and explain carefully why Borůvka's Algorithm would not compute a minimum spanning tree in this case.

(b) Now assume that no two edges in the graph have the same weight. By the OPTIONAL BONUS PROBLEM below, such a graph has exactly one minimum spanning tree (you may just use this fact here, although you are encouraged to prove it in the bonus problem!). Under this assumption, prove that Borůvka's Algorithm is correct.

(c) Why doesn't your proof from part (b) work if some edges in the graph have the same weights?

(d) How could Borůvka's Algorithm be modified slightly to work in the most general case that edge weights are not necessarily distinct?

(e) Describe an implementation (how the algorithm would be implemented using appropriate data structures) for Borůvka's Algorithm and derive its running time. Try to make your implementation as fast as you can.

(f) **[25 Point OPTIONAL BONUS PROBLEM!]** Let G be a connected undirected graph in which each edge has a distinct edge weight (that is, no two edges have the same weight). Show that there is a *unique* minimum spanning tree in the graph.