

CS140 - Assignment 8

Due: Sunday, Nov. 6th at 11:59pm

1. **[8 points]** State whether the following statements about a graph G that is undirected and connected are true or false and justify your answer.
 - (a) Prim's algorithm works correctly if G has negative edge weights.
 - (b) The shortest path between two nodes is always part of some MST.

The first algorithm for computing minimum spanning trees was published by the Czech mathematician Otakar Borůvka in 1926 and was used for laying out electrical networks. It goes as follows:

```
A = {}; \\ Comment: A is a subset of a minimum spanning tree

Consider the n vertices in the graph as n connected components;
while A contains fewer than n-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;
    Indicate that edge (u,v) is "chosen" but do not add it yet to A;
  }
  Add all "chosen" edges to A;
  Compute the new connected components;
}
return A \\ Comment: This is intended to be a MST!
```

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.

2. **[8 points]** Give a counterexample that shows that Borůvka's Algorithm doesn't work!! (You might find it useful to use the fact that some edges in the graph may have the same weight.)

Show your counterexample graph and explain carefully why Borůvka's Algorithm would not compute a minimum spanning tree in this case.

3. **[15 points]** Now assume that no two edges in the graph have the same weight. Such a graph has exactly one minimum spanning tree (you may just use this fact here, although you can think about how to prove it!) Under this assumption, prove that Borůvka's Algorithm is correct. (Make sure you refer to the min-cut theorem!)
4. **[6 points]** Why doesn't your proof from part (b) work if some edges in the graph have the same weights?
5. **[6 points]** How could Borůvka's Algorithm be modified slightly to work in the most general case that edge weights are not necessarily distinct? Explain briefly why this modification preserves the correctness of the algorithm.
6. **[10 points]** Describe an implementation of Borůvka's Algorithm. The description should be in clear English, indicate the data structures that would be used to support the algorithm and how they would be used, and give a careful derivation of the asymptotic worst-case running time using these data structures. Points will be awarded both on correctness as well as efficiency.