CS 374 **Spring 2015**

Homework 8

Nikhil Unni (nunni2)

3. (a) Say that we run Kruskal's algorithm and we get the following added edges, in order (as in from least to greatest c(e)), and get the sequence $S = S_1, S_2, \ldots, S_k$. We know for sure that this is a MST.

   Suppose that this MST is not the spanning tree that minimizes the bottleneck weight. Because the edges are ordered in Kruskal's algorithm, we know for a fact that the bottleneck weight of the MST generated by Kruskal's will be $c(S_k)$.

   If $c(S_k)$ is not the minimum possible bottleneck weight, then there must be another edge with weight less than $c(S_k)$ that is the true minimum, and we must remove S_k from the sequence altogether. But if this was the case, Kruskal's algorithm would have included such an edge, as it came before S_k in the order of sorted edges. The fact that it passed such an edge would mean that it would have introduced a cycle, and so it cannot possibly be part of the MST.

   Therefore we have a contradiction, and so we know that the MST minimizes the bottleneck weight.

   (b) From:[1]

```
Median ← CalculateMedian;
Partition graph into two subgraphs, grouping all edges ≤ Median in one
group, and all edges > Median in the other;
if the smaller grouping is connected then
 | return MinimumBottleneck(smaller subgraph)
end
else
 | foreach component ∈ smaller grouping do
 |  | Combine all vertices of the component into a single node
 | end
 | return MinimumBottleneck(larger subgraph);
end
```

Algorithm 1: MinimumBottleneck(V,E)

Because, at each stage, we are reducing the problem size by $\frac{1}{2}$, the algorithm runs in O(n) + O($\frac{1}{2}$) + O($\frac{1}{4}$) + ... = O(n). This is assuming that our median algorithm is linear time in the input size, which is true in the case of median of medians.

[1] http://people.scs.carleton.ca/ maheshwa/courses/5703COMP/14Seminars/BST-Report.pdf