PROBLEM 2 *Divide and Conquer with MSTs* [13 points]

Professors Horton and Hott have been discussing minimum spanning trees and attempting to create new algorithms to compute them. Professor Horton claims to have created a divide and conquer algorithm as follows:

> Given a graph $G = (V, E)$, partition the set of vertices V into two sets V_L and V_R such that $|V_L|$ and $|V_R|$ differ by at most 1. Let E_L be the set of edges that are incident only on vertices in V_L and E_R be the set of edges incident only on vertices in V_R. Recursively compute the minimum spanning tree on each of the two subgraphs $G_L = (V_L, E_L)$ and $G_R = (V_R, E_R)$, then select the minimum weight edge $e \in E$ that crosses the cut (V_L, V_R). Use e to combine the two minimum spanning trees into a single minimum spanning tree for G.

Help us to evaluate his algorithm. Either prove that it correctly computes a minimum spanning tree of graph G or provide a counterexample for which the algorithm fails.

Answer:

We don't always get an optimal solution with the divide and conquer method explained above.

Let's take in consideration the following graph G, where Edge(B, C) has cost 1 and (F, G) cost 2, and all others cost of 10.

```
A – B – C – D                                                            A – B    C – D
|     |     |     |         , after the first DAQ execution, we get the Graph like         |     |     |     |
E – F – G – H                                                            E – F    G – H
```

```
Then divide again, and get    A – B    C – D
                              |               |   , divide and conquer outputs this two as two MSTs. Let's
                              E – F    G – H    connect them at (B, C) and we get a MST with cost of 61.
```

```
But, the actual minimum spanning tree would be with cost 51  A – B – C – D
                                                             |
                                                             E – F – G – H
```

Hence, by the counterexample given, we might say that the Divide and Conquer algorithm was wrong.