**Kruskal’s Algorithm Growth Rate Investigation**

**CS4310 - Design & Analysis of Algorithms**

**Spring 2017**

**John Harvey**

**04/25/2017**

**Hypothesis:**

Kruskal’s algorithm has a time complexity of O(m logn)

**Test Design:**

In order to test the hypothesis involved, I have taken the following steps:

1. Implement Kruskal’s algorithm in Ruby
2. Run the code for multiple amounts of n and m items, also with various random weights
   1. Test and gather data for when number of vertices = number of edges
      1. Test again with higher range of random edge weight
   2. Test and gather data for when there are more edges than vertices
      1. Test again with higher range of random edge weight
3. Inspect the graphs and compare the growth to an m log n growth rate
4. Examine code and make changes if results do not seem right

**Evaluation of Data:**

How the data was generated:

The user inputs a number for the amount of vertices, and the amount of edges. The edge weight is a randomly generated integer between 0 and any number you would like. I hard coded different edge weight ranges to reduce time. Multiple ranges of vertices, edges, and edge weights were done to achieve more accurate result. All data was logged in Excel.

**Visual Inspection:**

When there are more edges than vertices:

This shows that as the amount of vertices increase, the time increases, but when the edge amount increases more than the vertices, the time difference is minute.

**Conclusion:**

I discovered that Kruskals algorithm does in fact run in Big O(m log n). Every set of tested data ran in under m log n time. The time it takes grows as the vertices grow more than as the edges grow. Some tests even ran faster with more edges. I also discovered that increasing the range of the possible edge weight did not drastically change the time elapsed.