**CSCI E-124 - Minimum Spanning Trees in Random Graphs**

**By: Tofik Mussa**

I was tasked with collecting data points to determine a formula for the average weight of a minimum spanning tree for a graph with randomly generated weighted edges in dimensions 0 and 2 through 4.

I used java for the implementation. I initially drew a complete graph and because of having to store the edges in memory, I quickly ran out of heap space. The maximum graph I could generate within a reasonable amount of time was only 2^13 vertices. I tried to come up with a regression line to predict the maximum weight for a given number of vertices in the two-dimensional case. I incremented the number of vertices by 100 to get a better approximation for finding out about the maximum weight. I run 20 trials for each number of vertices and computed the average for the trials.

I got this best fitting line running the experiment once for 27 data points for the zero-dimensional case starting from 2 up to 2702 incremented by 100.

Shape

Description automatically generated

I got this best fitting line running the experiment once again for 27 data points for the zero-dimensional case starting from 2 up to 2702 incremented by 100.

Shape

Description automatically generated

I got this best fitting line running the experiment once for 26 data points for the two-dimensional case starting from 2 up to 2602 incremented by 100.

Chart, scatter chart

Description automatically generated

I got this best fitting line running the experiment once again for 26 data points for the two dimensional case starting from 2 up to 2602 incremented by 100.

Chart, scatter chart

Description automatically generated

I got this best fitting line running the experiment once for 23 data points for the three-dimensional case starting from 2 up to 2302 incremented by 100.

Chart, scatter chart

Description automatically generated

I got this best fitting line running the experiment once again for 23 data points for the three-dimensional case starting from 2 up to 2302 incremented by 100.

Chart, scatter chart

Description automatically generated

I got this best fitting line running the experiment once for 23 data points for the four-dimensional case starting from 2 up to 2302 incremented by 100.

Chart, scatter chart

Description automatically generated

I got this best fitting line running the experiment once again for 23 data points for the four-dimensional case starting from 2 up to 2302 incremented by 100.

Chart, scatter chart

Description automatically generated

I then compared if throwing edges have any effect. I compared the data points I had before throwing out the edges and after to check if there is any significant difference between the weight of the spanning trees.

Seeding

How to run it

I created modular classes for object orientation.