**FractionalKnapsack Growth Rate Investigation**

**CS4310 - Design & Analysis of Algorithms**

**Spring 2017**

**John Harvey**

**04/09/2017**

**Hypothesis:**

The FractionalKnapsack algorithm has a time complexity of O(n logn)

**Test Design:**

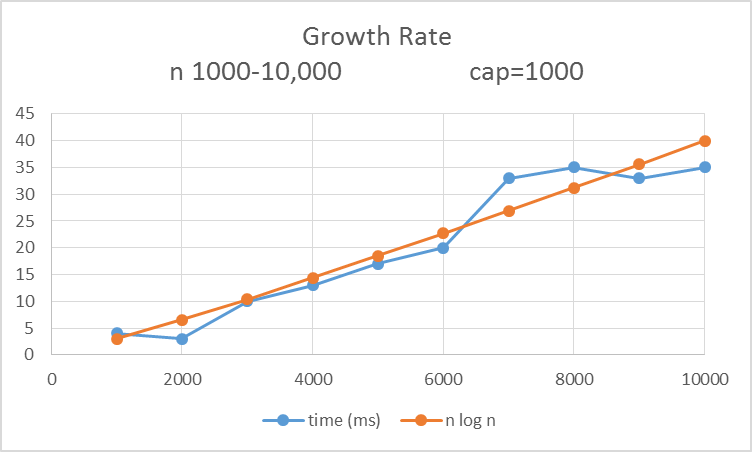
In order to test the hypothesis involved, I have taken the following steps:

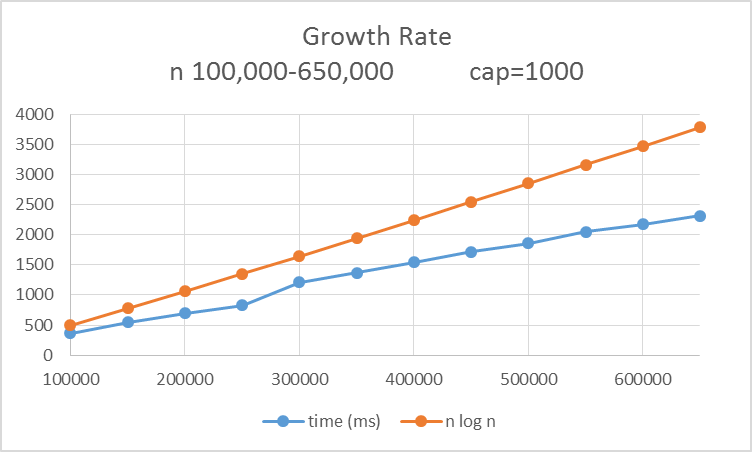
1. Implement the FractionalKnapsack algorithm in Ruby
2. Run the code for multiple amounts of n items
   1. Test with various maximum capacities
   2. Analyze all data on individual graphs
   3. Analyze all data on a single graph
3. Inspect the graphs and compare the growth to an n log n growth rate
4. Examine code and make changes if results do not seem right

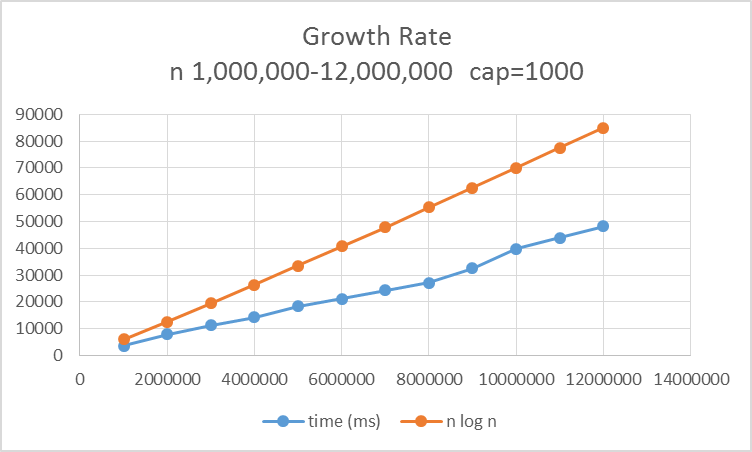
**Evaluation of Data:**

All files will be included in the submission .zip.

**Visual Inspection:**







The graphs above show each simulation given different size inputs of n. Each run is plotted against the n log n line. By looking at each one, we can easily reject the exponential possibility for all. We can see that each simulation comes fairly close to the n log n trend line. Actually, our simulation line comes up below the n log n trend line in each one except for the first. The time for each one is in milliseconds.

**Conclusion:**

The growth of the fractional knapsack algorithm is very close to n log n. As I pushed the boundaries, my computer would fail to allocate memory.