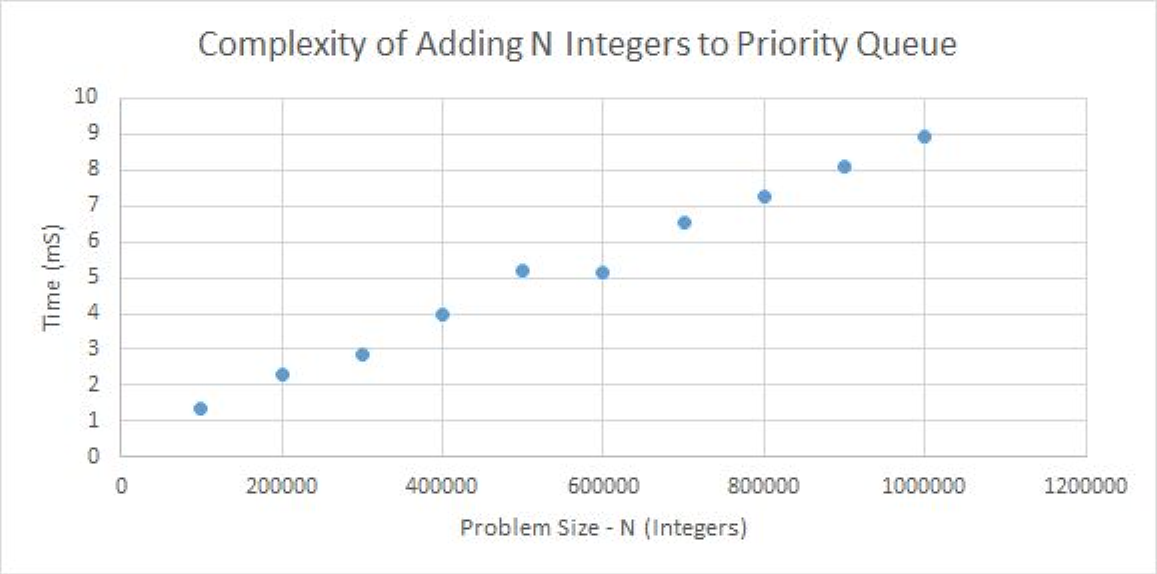
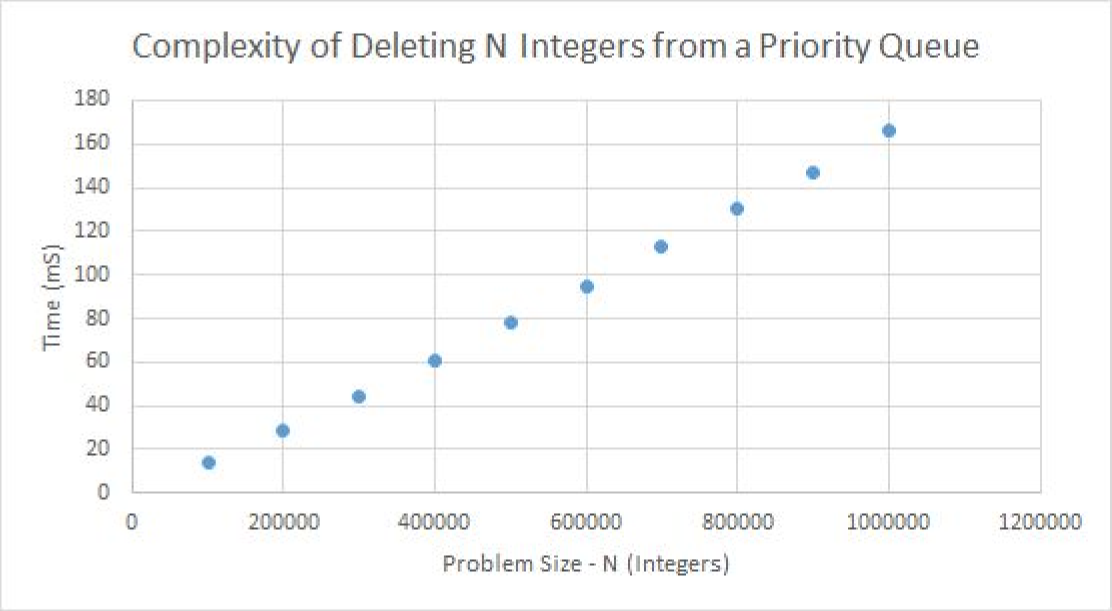
Dennis Kuduzovic
U1021366

Assignment 11 Analysis

1. Design and conduct an experiment to assess the running-time efficiency of your priority queue. Carefully describe your experiment, so that anyone reading this document could replicate your results. Plot the results of your experiment. Since the organization of your plot(s) is not specified here, the labels and titles of your plots(s), as well as, your interpretation of the plots is critical.



This graph shows that the complexity of the priority queue is a very slight logarithmic curve. In this way, the add() function of the priority queue is O(logN). I believe it is slight because finding a place to add an item takes 1.6 permutations.



This graph appears to be extremely slight logarithmic curve. I think this is due to the array not changing sizes very quickly. However, I calculated a slight decrease over the increase of the problem sizes and can conclude that the deleteMin() function is also O(logN)

2. What is the cost of each priority queue operation (in Big-O notation)? Does your implementation perform as you expected? (Be sure to explain how you made these determinations.)

Testing the findMin function is unnecessary because the function returns the first item in the storage array. This is in O(C) running time, so findMin is also O(C).

The first graph shows that the complexity of the priority queue is a very slight logarithmic curve. In this way, the add() function of the priority queue is O(logN). I believe it is slight because finding a place to add an item takes 1.6 permutations.

The second graph appears to be extremely slight logarithmic curve. I think this is due to the array not changing sizes very quickly. However, I calculated a slight decrease over the increase of the problem sizes and can conclude that the deleteMin() function is also O(logN)

3. Briefly describe at least one important application for a priority queue. (You may consider a priority queue implemented using any of the three versions of a binary heap that we have studied: min, max, and min-max)

Priority queues are mostly used in situations where items that are of equal weight are served fifo, but if a more important, or more weighted, item is added, then it is prioritized. Priority queues are useful with Dijkstra's algorithm. Priority queues can also be used in Huffman compression where you would need to obtain the two lowest frequency trees.

4. How many hours did you spend on this assignment?

6