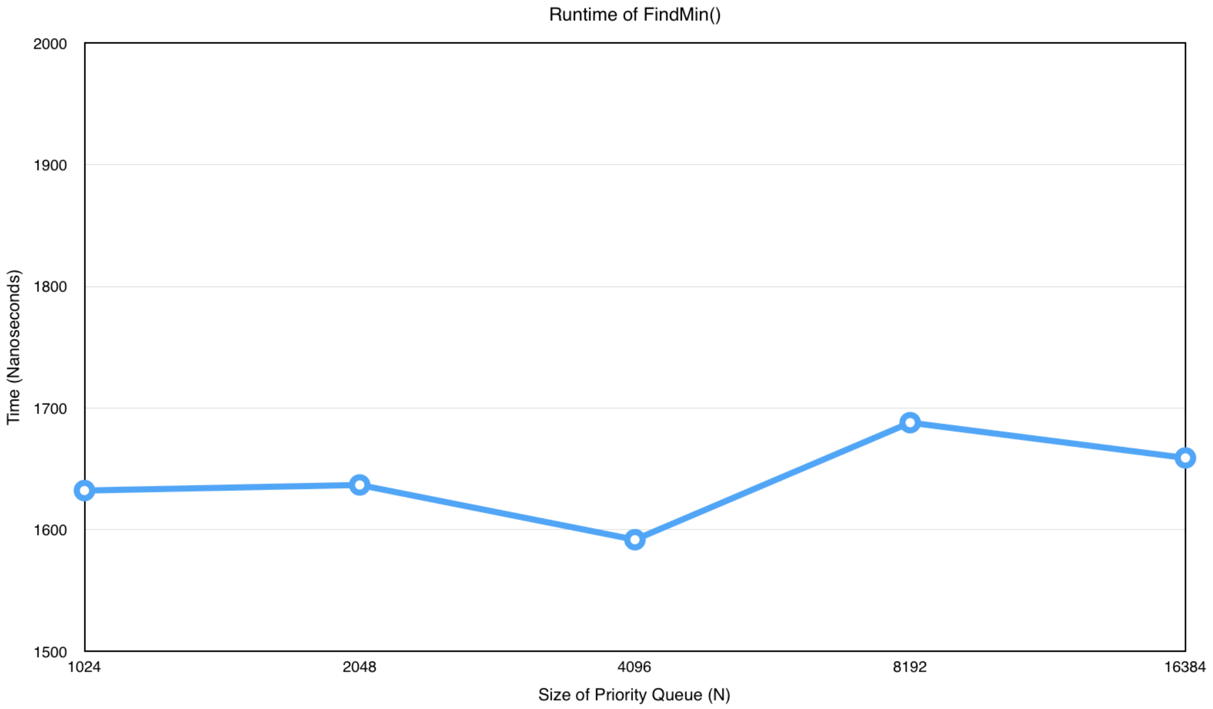
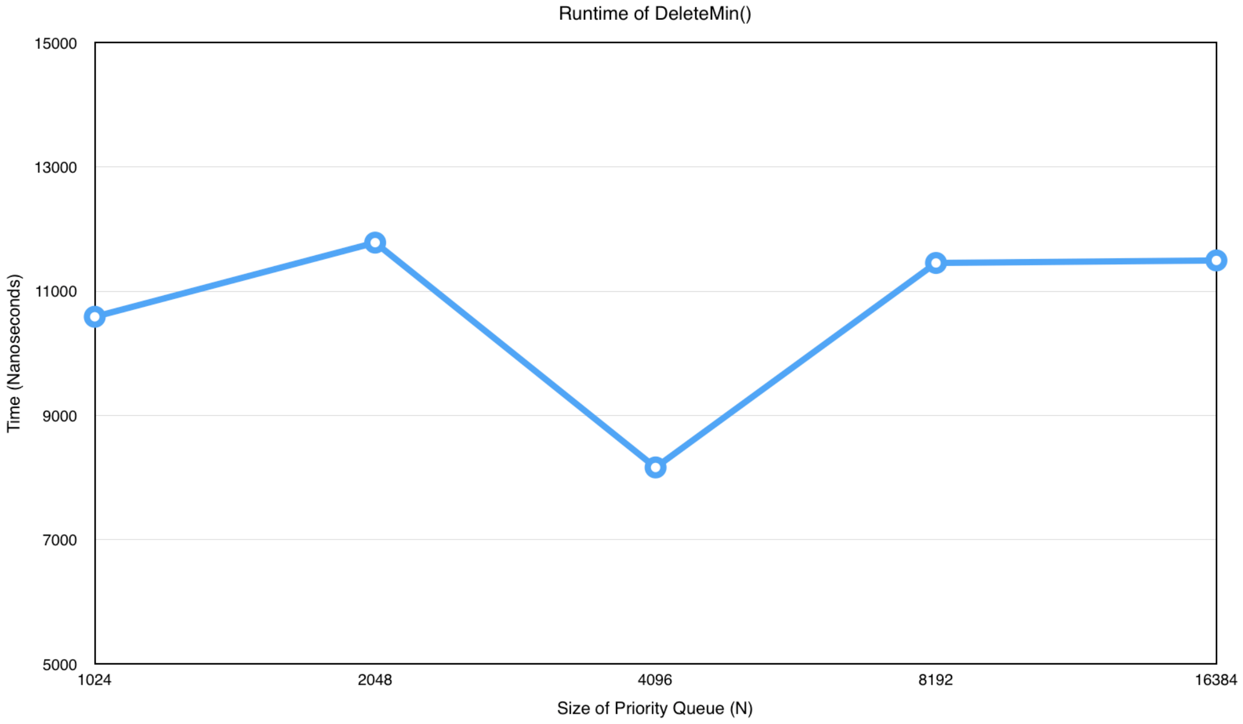
Yip Ho Lam u1025709

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.

I have plotted the graph for both FindMin() and DeleteMin(). The increments of N (which is the priority queue size) are done in powers of 2, starting from 2^10 to 2^14.





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.)
Add() costs O(logN). Whenever a new element is added, it will first be added to the end of the array. Then, it will be compared to its parent, if its parent is larger than itself, it will percolate upwards. The worst case of this operation is when the new element added is the smallest element in the entire minHeap, and it has to percolate from the bottom to the top.

FindMin() costs O(C). It should always return the first element of the array. As long as we are not adding or removing anything, simply accessing an element of an array is a constant operation.

DeleteMin() costs O(logN). The root (min item) is replaced with the last item in the array, and then percolated downwards. The worst case of this operation is when the last item in the array happen to go back to the lowest level of the minHeap, and it has to percolate from the top to the bottom.

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 very useful for implementing Djikstra’s Algorithm, in which we are always searching for the cheapest element (whatever cheapest is defined to be).

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