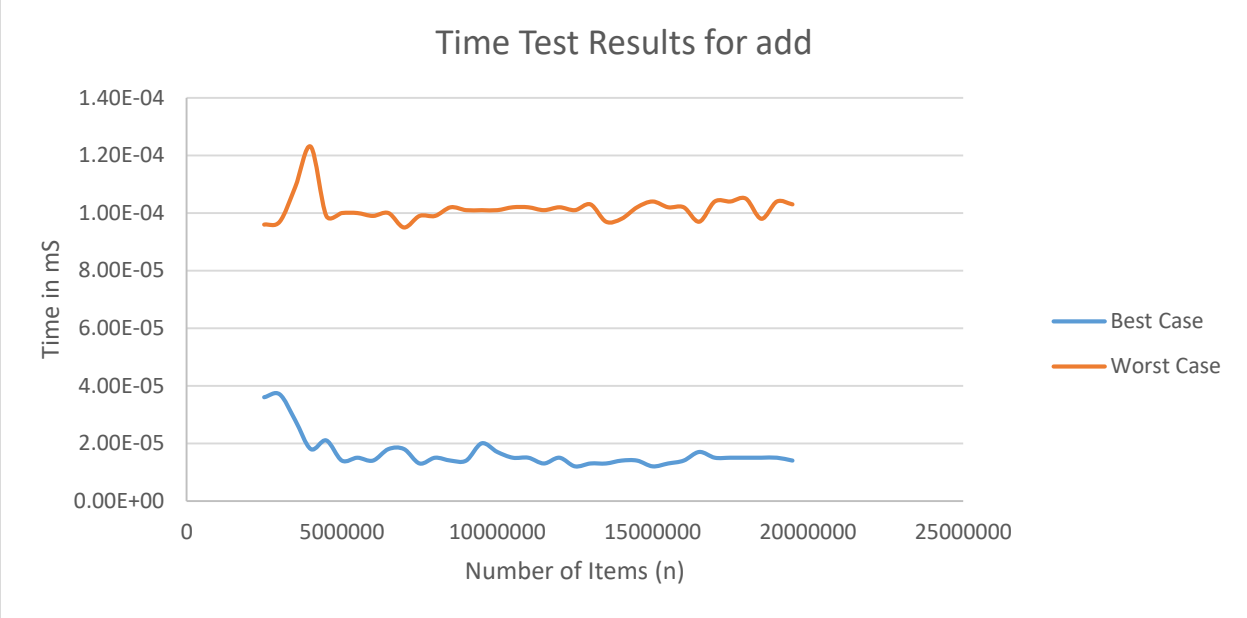
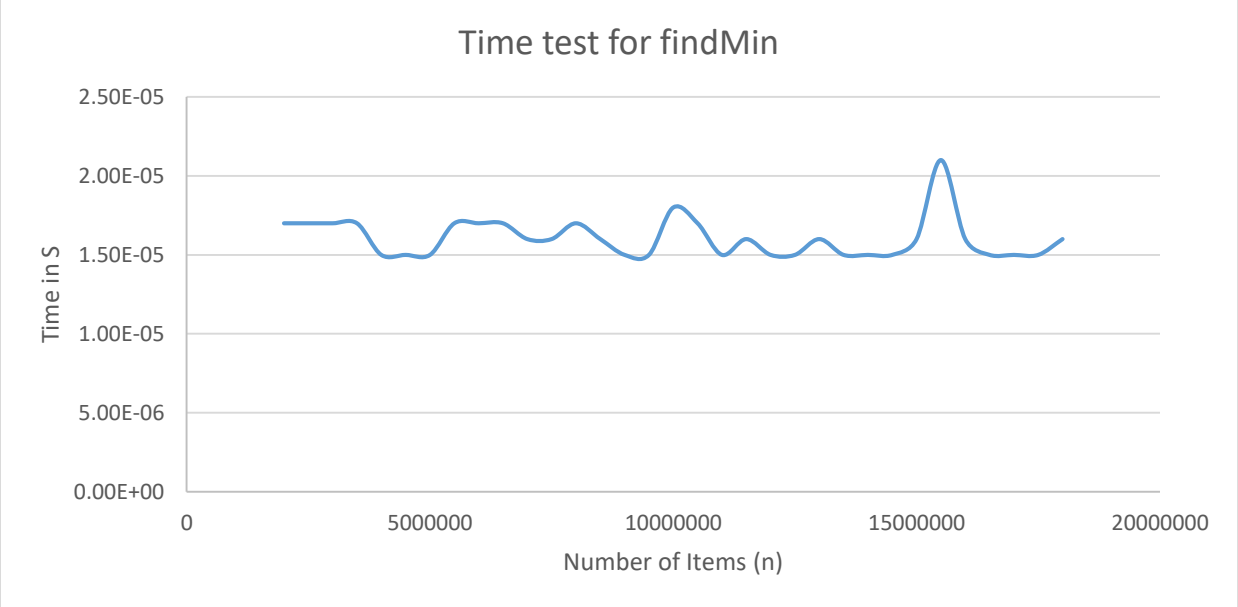
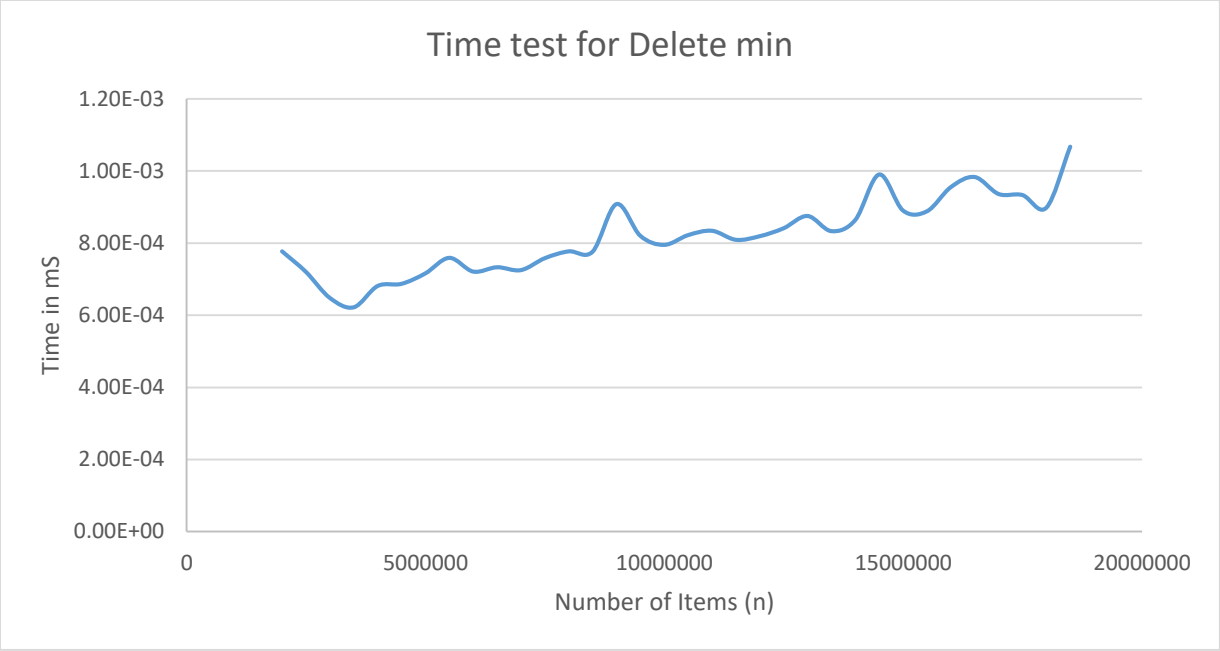
Chase Stephens

Assignment 11

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.







For each test, an integer priority queue was instantiated and added with integers one up to size. Multiple sizes were tested, starting at one million and incremented by five hundred thousand up to two million. For each size tested, 1,000 iterations of the test were performed and the average time was taken.

For each iteration in the worst case add test, the time required to add the number -1 was measured (smaller than any item in the tree). -1 was then removed using removeMin() in order to leave the queue in the same state for the next iteration. The worst case showed reduced performance from the best case as expected. The line looks slightly greater than linear, which would be expected for a O(log(n)) operation.

For each iteration in the best case add test, the time required to add an integer with the value size (greater than any item in the queue) was measured. The Min value was then removed and the time required to add size+1 was measured. The value added was incremented each time to guarantee that the largest number in the queue was added for each iteration. The worst case for adding shows a close to linear graph as well, which is what would be expected for an O(c) operation.

For each iteration in the findMin() methods, the time required to find the minimum value was measured. The graph looks roughly linear.

For each iteration of the deleteMin() test, the time required to delete the minimum value was measured. After removing the item, an element with value size was added back to the queue. For the next iteration, after removing the smallest element, an item with value size+1 was added. This was to avoid duplicates and make sure a bounded set of adjacent natural numbers was used for each test.

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.)

The implementation for each case generally performed as expected when ignoring some of the random fluctuations on the graphs.

For the worst case scenario for the add() method, we should expect a O(log(n)) complexity. This is because adding always initially adds the item to the bottom of the tree. A min value to the bottom of the tree requires log(n) steps for the item to percolate back to the top. For the best case, the item can remain in its initial position which then only requires O(c) complexity. In both cases, there will be points where the array has to grow. This may be responsible for some of the spikes in the graph. The nature of testing on an inconsistent machine may also create random fluctuations.

For the findMin() method a O(c) complexity is expected. This is because all that is required is returning the first item in the array. The graph clearly does follow a linear trend.

For the deleteMin() method an O(log(n)) n is expected. This is because the smallest item in the tree is replaced with a leaf node that must then percolate all the way to the bottom. The complexity should be O(log(n)) in any case.

3. Briefly describe at least one important application for a priority queue. (You Dijkstra's algorithm Dijkstra's algorithm **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)**

Dijkstra's algorithm uses a priority queue. This is because the item that currently has the smallest weighted path must be evaluated first. Another example could be a hospital. A patient with a more urgent situation might have a higher priority and would then need to be bumped up in the queue.

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

9