Cameron Pickle

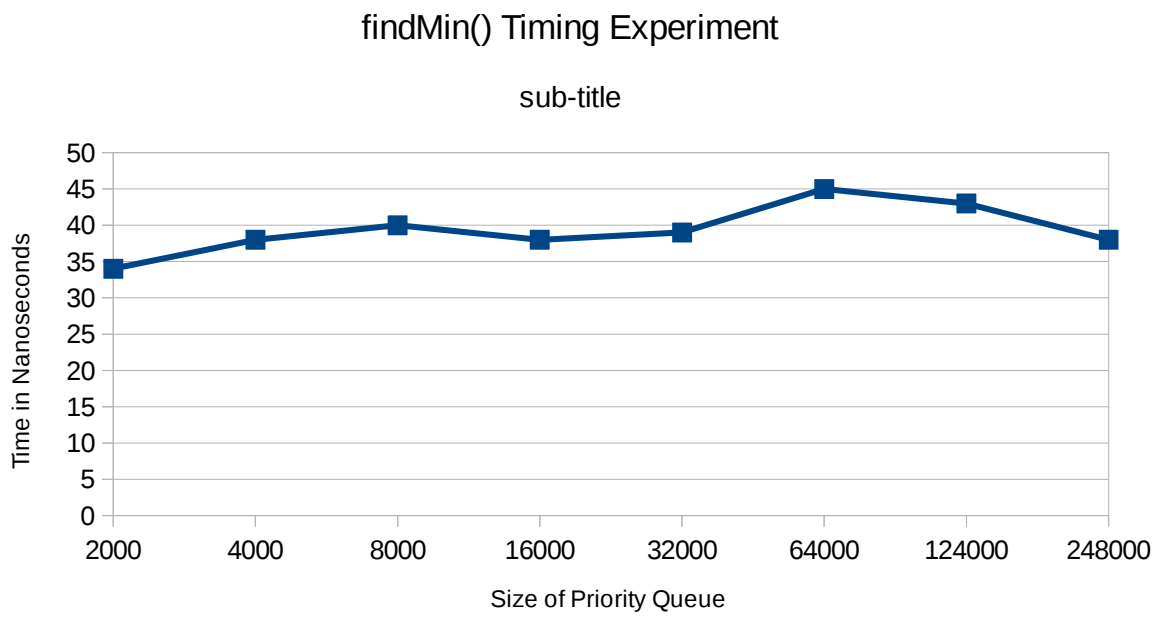
1. Who is your programming partner? Which of you submitted the source code of your program?

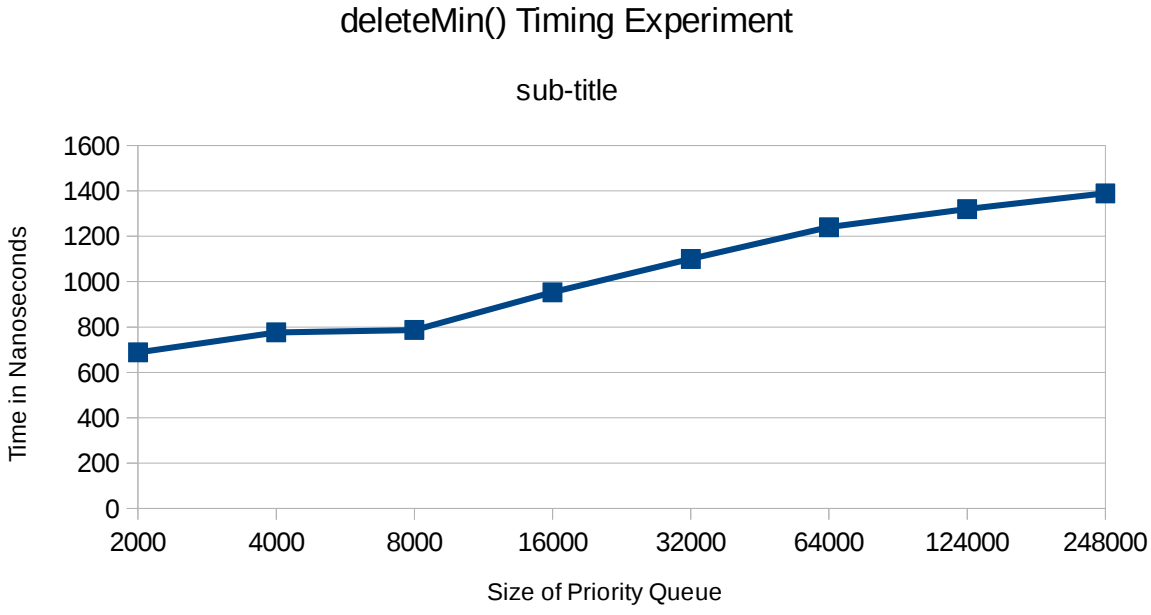
Daniel Avery. He will turn in the code.

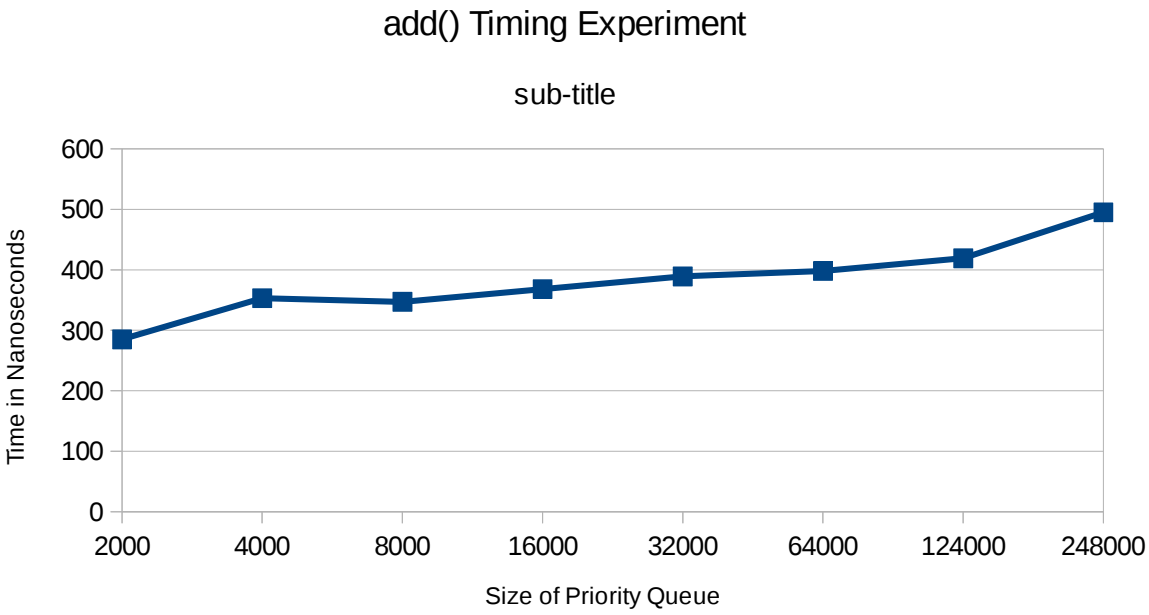
2. Evaluate your programming partner. Do you plan to work with this person again?

He does a good job and is very thorough in making sure that the code works how it should.

3. 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 our timing experiments we grew our size of N by doubling it every time. Therefore all the data on the charts is not how it appears to the eye.

For the findMin() experiment we got the results that we were expecting of O(C). All of the calls to it were in constant time as we grew the queue to very large sizes. We created a arrayList of priority queues and tested each one individually to ahieve the most accurate results.

The deleteMin() appears to be growing at a linear rate on the graph but when considering that the x axis doubles every time it becomes apparent that it is actually logarithmic growth as would be expected.

The add() timing experiment appears to grow at a logarithmic rate but when considered with the doubling x axis we can see that it is a very slow logarithmic growth that is almost constant like we expect.

4. 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.)

Priority Queue	Average Case	Worst Case
enqueue	O(C)	O(N)
dequeue	O(C)	O(C)

Our Priority Queue	Average Case	Worst Case
enqueue	O(C)	O(logN)
dequeue	O(C)	O(C)

Priority queues have two function enqueuing and dequeuing. Both of these functions on average should be of order O(C). The worst case however for enqueing could be O(N) because the queue must remain in sorted order of priority to dequeue in constant time.

5. 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)

One example of when a priority queue is needed is when implementing Dijkstra's sorting algorithm. A priority queue helps by returning the path that currently has the least distance first in order to find the path with the smallest distance in the end.

A priority queue could also be used to to make a program where you add patients at a hospital with a priority based on the severity of their medical problem. When a doctor is available the priority queue would then return the next patient with the greatest need.

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

We spent around six hours on the assignment.