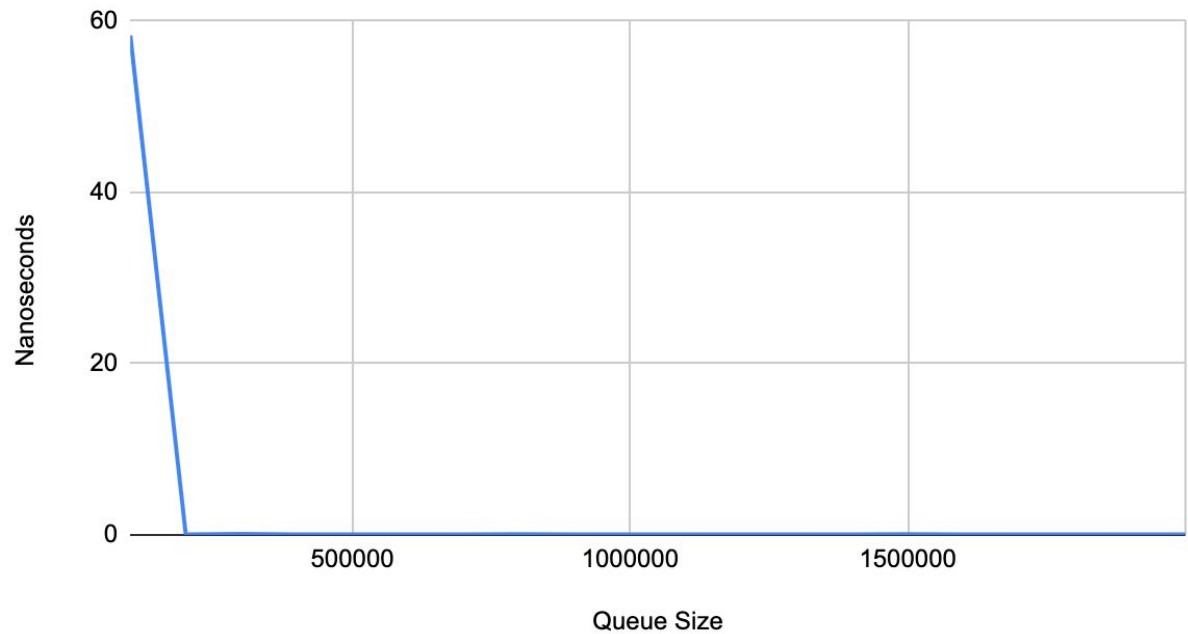


Analysis Document — Assignment 3

1. I am **not** invoking one of my three exemptions for pair programming due to an extenuating circumstance. My programming partner is Braden Morfin, and I submitted the program to Gradescope.
2. Pair programming experience:
 - My partner and I would switch roles every few methods completed or so. I think we both tried our best to keep the time to switch fair, so I don't think I would have preferred to switch less/more often.
 - I think my partner and I applied the techniques of pair programming to this assignment very well. I think we both learned from each other a lot.
 - Together, my partner and I developed the program for this assignment fairly efficiently. We got a little stuck on the binary search code, but we were able to resolve that after stepping away from the code for a bit.
 - My partner and I spent about two hours each day to complete the assignment and create tests for the assignment for a total of 6 hours.
 - Braden's very smart and is a nice person. I would plan to work with Braden again.
3. My partner and I did use a basic array to back the priority queue. I feel that using Java ArrayList would've been the same because while ArrayList has a fixed amount of array and copies its elements over to a newer and larger array, what we do with the basic array within the code is exactly what we would do with ArrayList when we reach the capacity of the array. I supposed it could be argued that with a basic array we can start out with a bigger array length to delay the need to create a larger array.

4. In order to collect the data, I plotted the running time of the findMin method for problem sizes 100,000 to 2,000,000 by steps of 100,000.

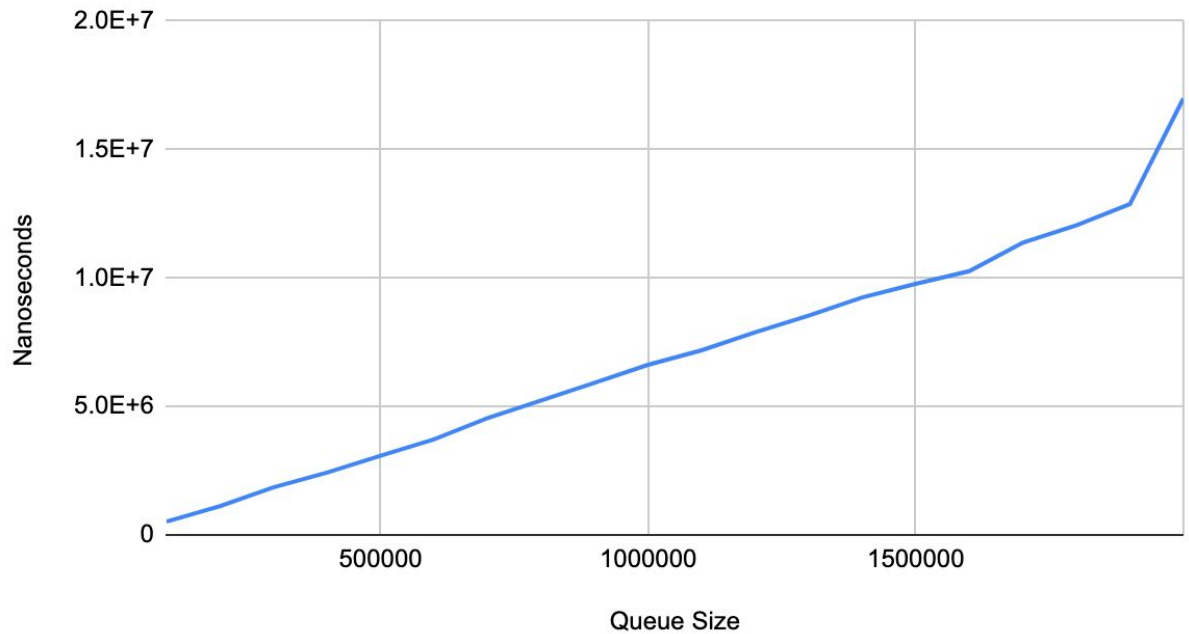
Run Times of findMin() Method



I think this makes sense to me because the findMin method only uses size-1 to determine the last address and returns the value within that address. The big O would be constant ($O(1)$) and so it would be really fast.

5. In order to collect the data, I plotted the running time of the insert method for problem sizes 100,000 to 2,000,000 by steps of 100,000.

Run Times of insert() Method



The Big-O behavior of the insert method is $O(N)$ because the method has two for loops that are sequential. For the majority of the graph, I would say that the linear pattern was expected.