5. I believe that my choices for the value of “n” are valid because I ran each test for every size of “n” several times, and the outputted time stamp was very close after each run. My choices for “n” were good ones because they increased by the same increment each time.

6. I chose to have the program choose any int values in the range from 0-10. I believe that this a good range because there is a guaranteed chance of having duplicate numbers, which will prove the validity of the algorithm and give a variance in the output.

7. The sanity checks are reasonable because they prove the validity of the code, and make sure that there is nothing that is being miscalculated.

8. My results support that statement that the maximum age of an element in a priority queue has O(n) behavior. This can be proven by looking at the graph that I made with all of the timestamps and corresponding sizes of the queue. As the size of the queue grows, the total time that it takes for the program to insert all of the elements, and then delete all of the elements in descending order takes longer. The larger the queue, the more time that it takes, not by much, but the rate increases at a steady rate of growth.