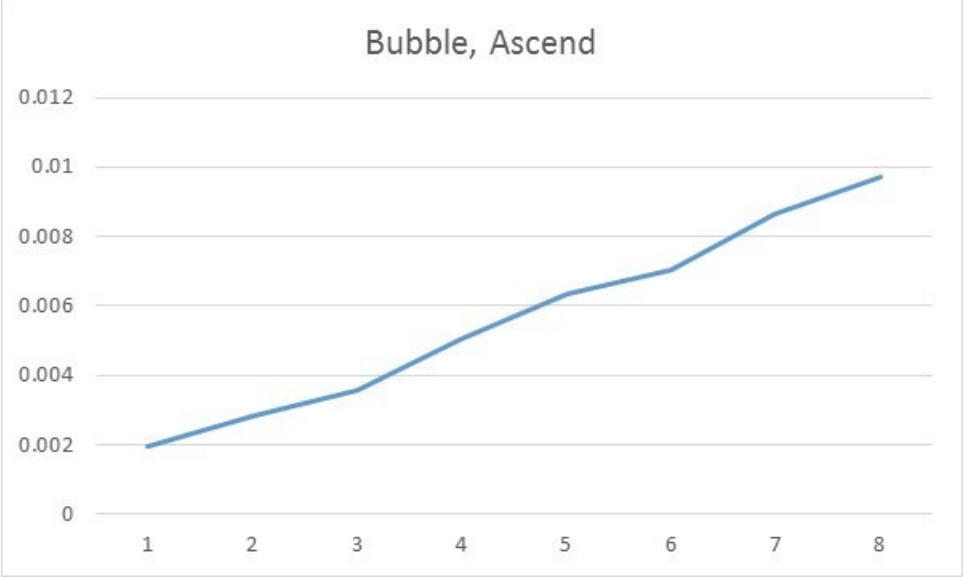
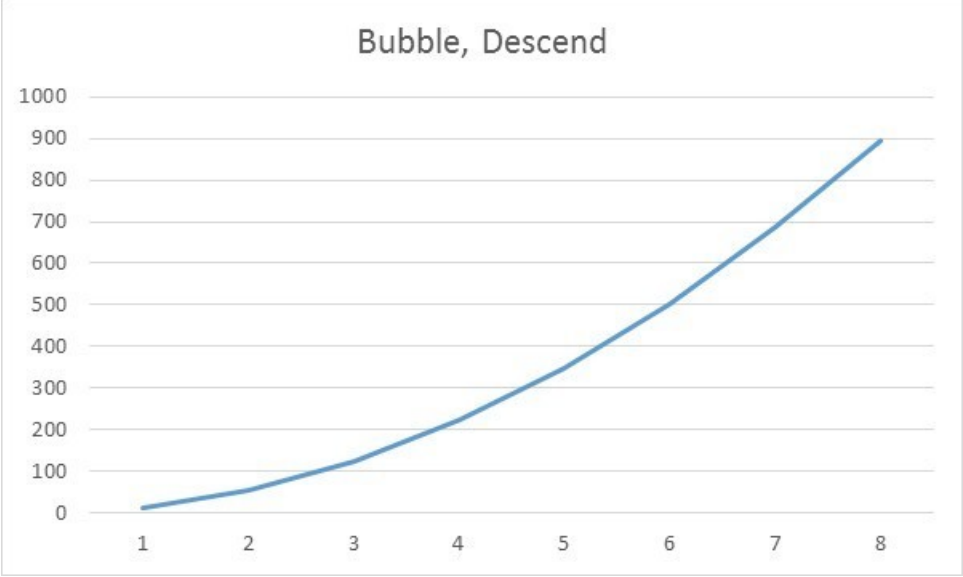
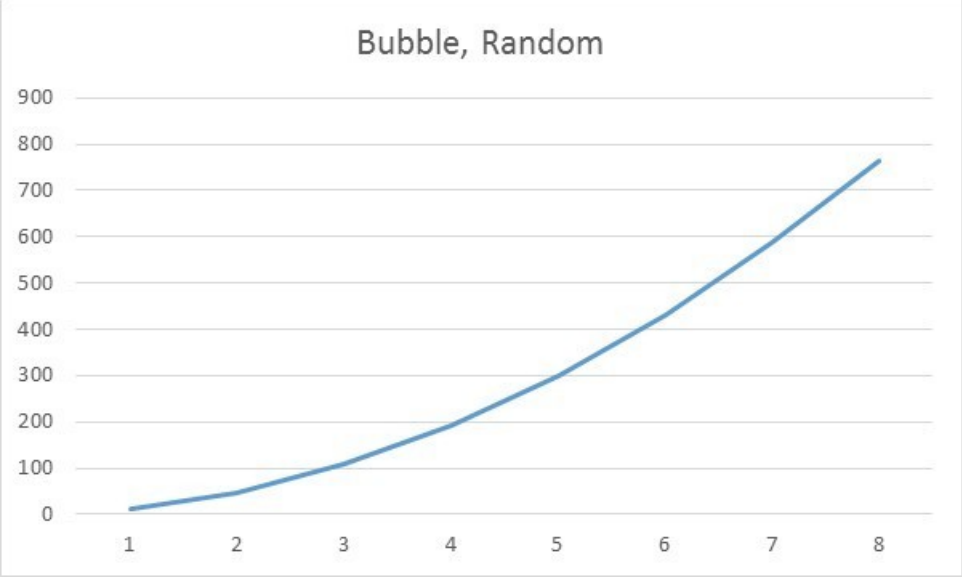
BUBBLE



I believe this graph is trying to O(n^2), if I had chosen better numbers, the graph would have looked like n^2. The time would probably about double my time for 5,000,000, which was 0.009724. This would probably define the graph even more.

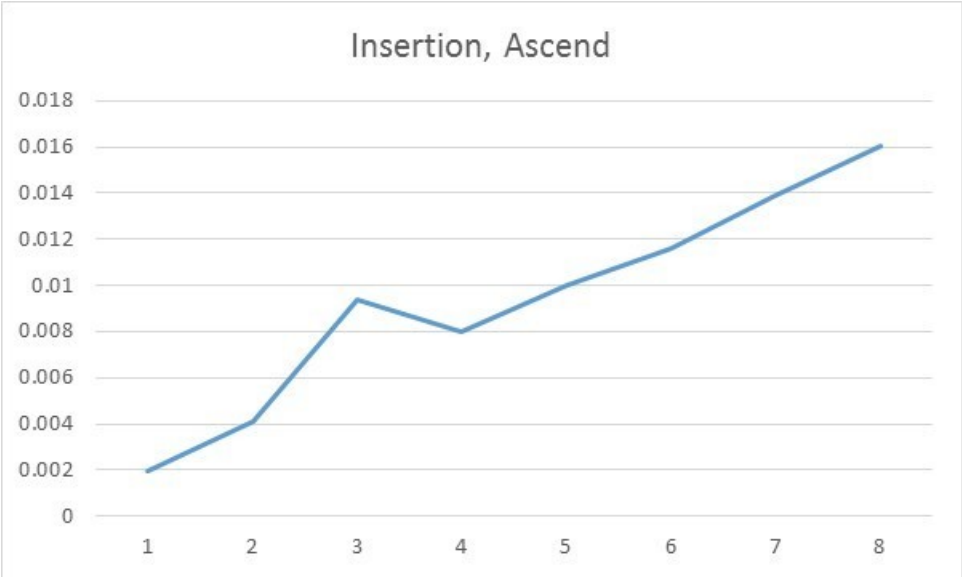


This graph is a better representation of O(n^2). This is understandable since it has a while loop to the maximum of n and a for loop within that going to a maximum of n. Thus, resulting in n^2. Although 10,000,000 would take a long time. My maximum is size 400,000 and that took 893.54 seconds. Thus, I believe it will be above 1,000 seonds. Probably close to 1,500 seconds.

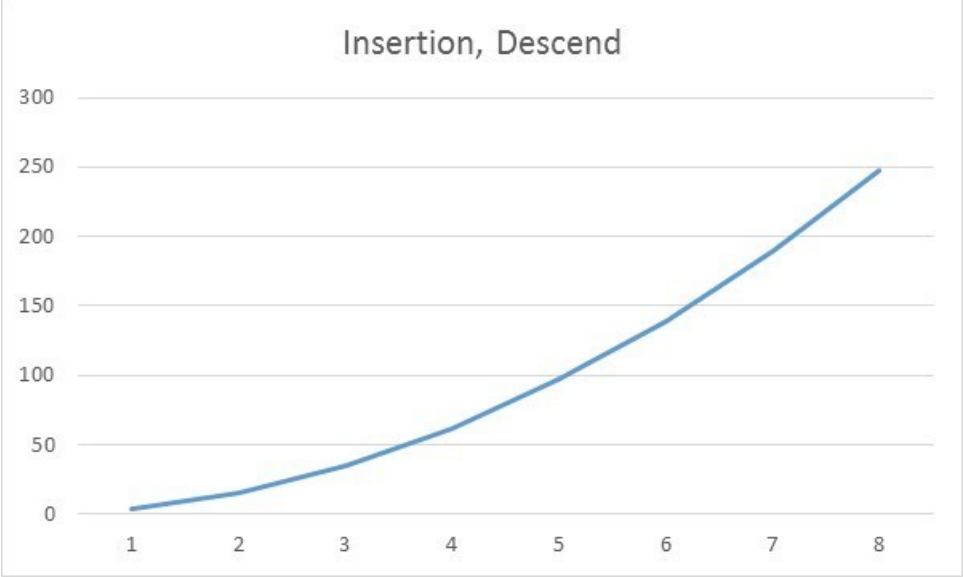


This graph is clearly O(n^2) as well. Considering the loops implementing the sort. This shows the average case for bubble sort. Sorting an array of 10,000,000 would probably result in a time around 900 to 1,000 seconds.

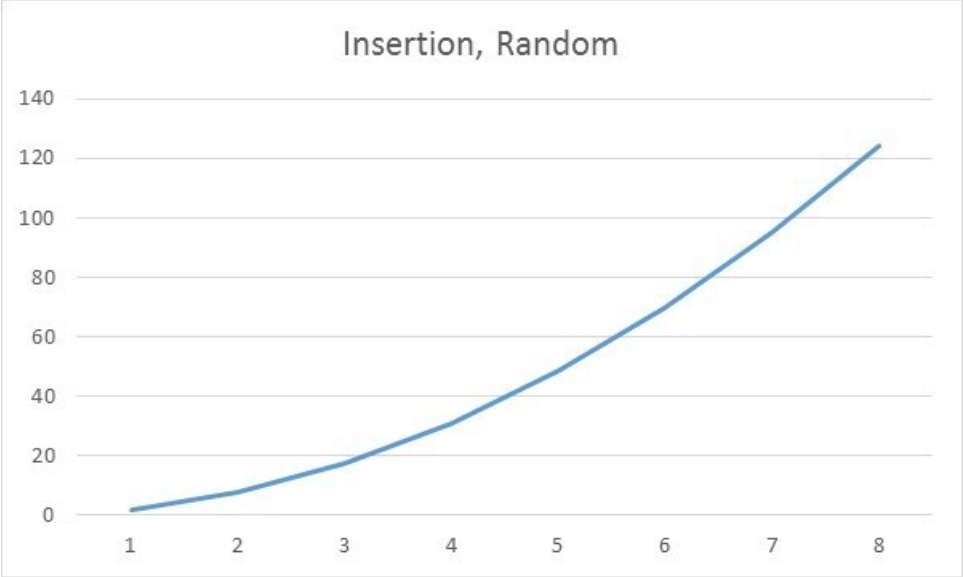
INSERTION



This is the best case scenario for insertion. I believe it is trying its best to be O(n^2), but I could have used better numbers to represent that. As for inputing an array size 10,000,000, it wouldn't even go over 0.1 second.

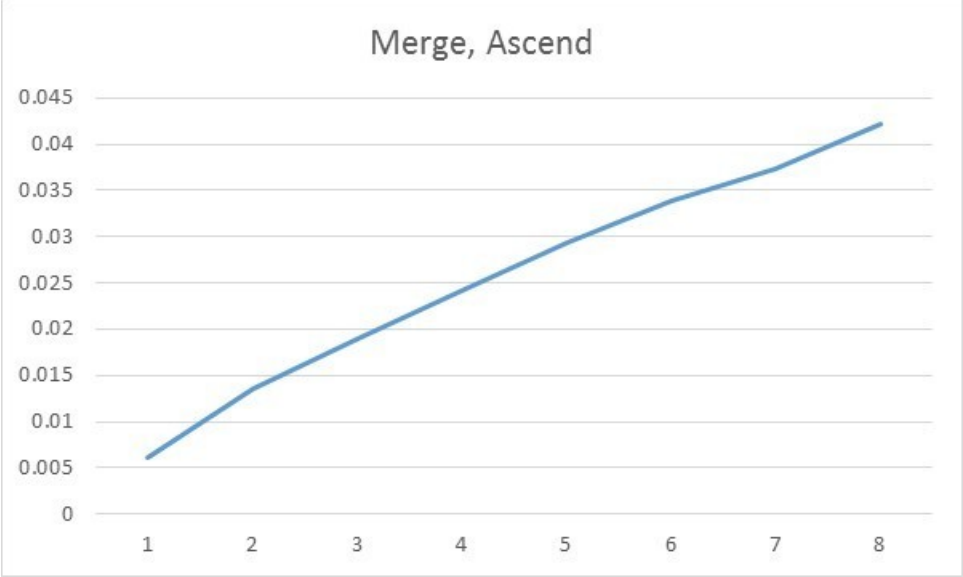


This graph, obvoiusly, is O(n^2). The numbers used worked out better in this case. This would represent the worst case scenario. 10,000,000 would probably take around 1000 seconds to sort.

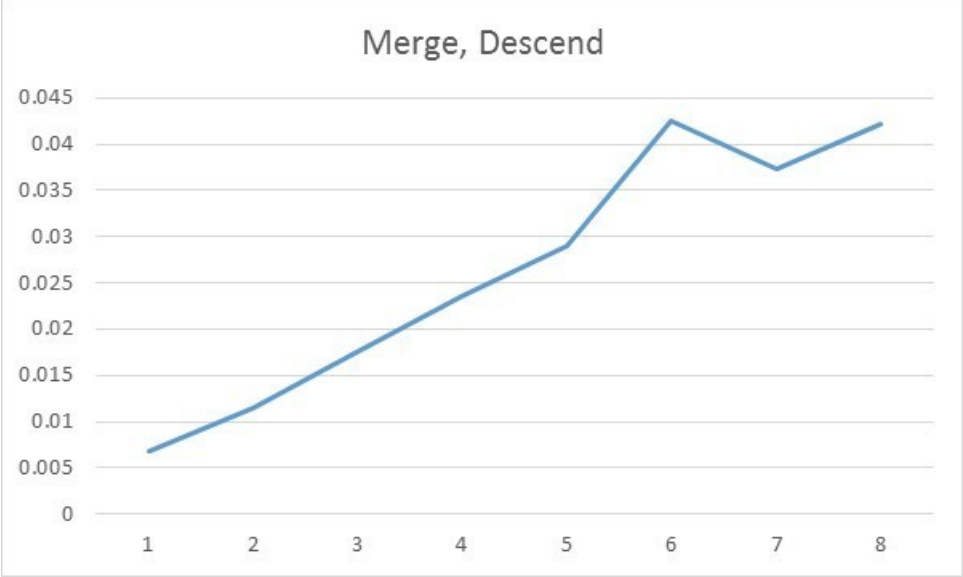


This representing the average case, would show O(n^2). That being said, it still makes sense due to the implementation used for the sort. 10,000,000 would most likely result in around 700 seconds.

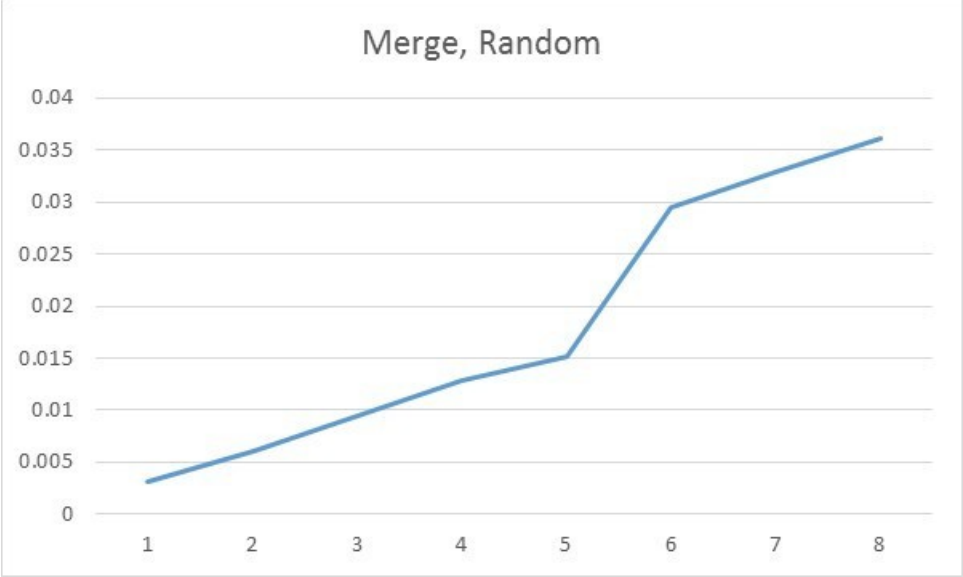
MERGE



My initial guess would be -n^2, but after thinking about it, I believe it is,indeed, showing O(nlog(n)) for this best case. That is what it should be for this sort. Doing 10,000,000 in this case would potentially show 0.09 seconds perhaps.

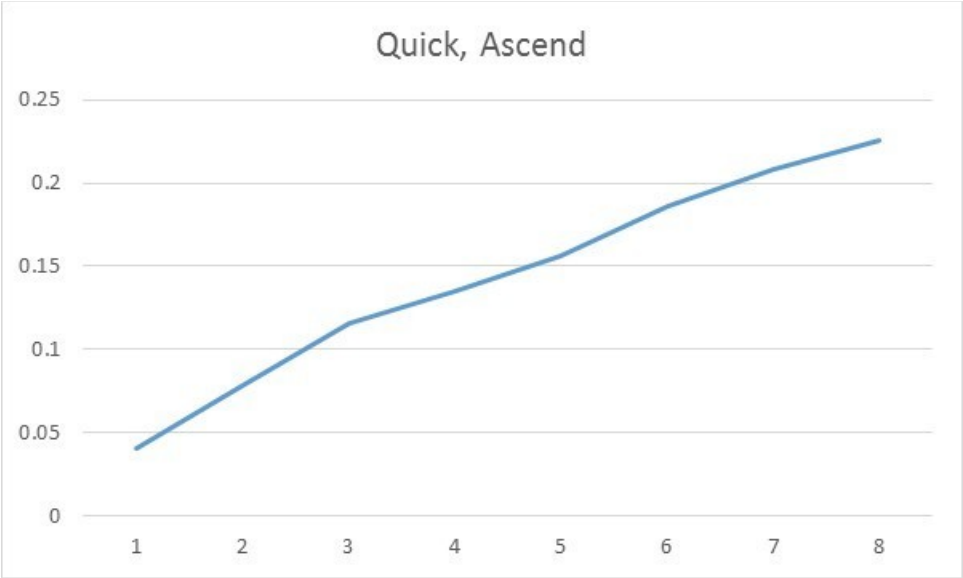


For this worst case scenario, the graph is trying its hardest to be O(nlog(n)). It would have been if I had chosen better numbers. As for 10,000,000, it would most likely result around 0.08 seconds.

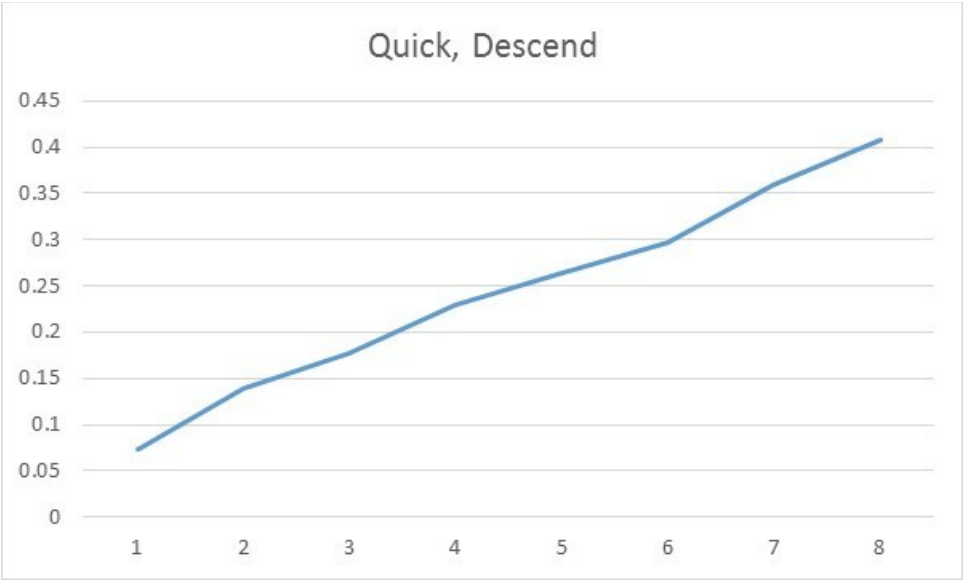


For this average case, It is trying to be O(nlog(n)) as well. Perhaps the number choice could have been a better range. 10,000,000 would probably be 0.05 seconds at most.

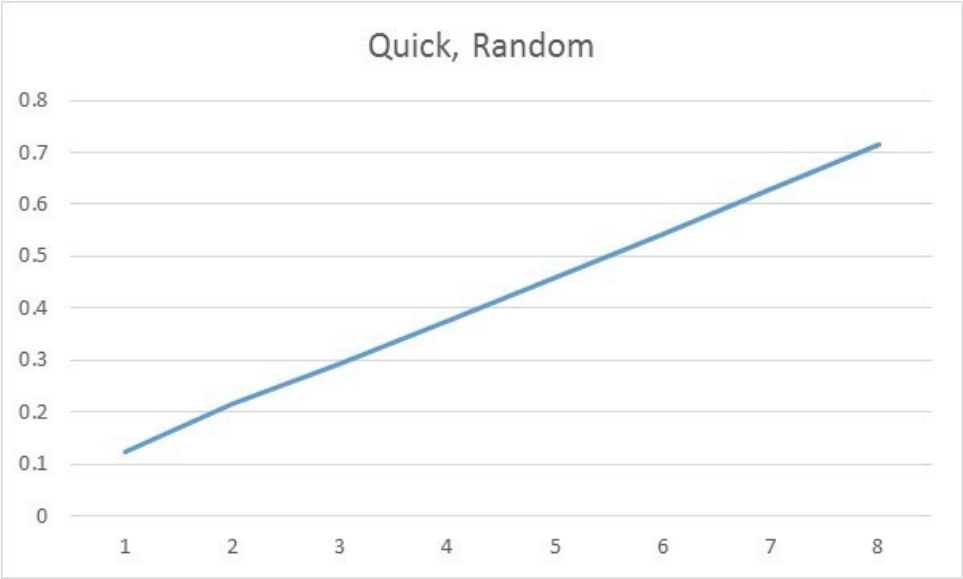
QUICK



Similar to merge sort, this best case for quick sort is O(nlog(n)). The representation isn't too bad. 10,000,000 probably wouldn't break above 1 second.

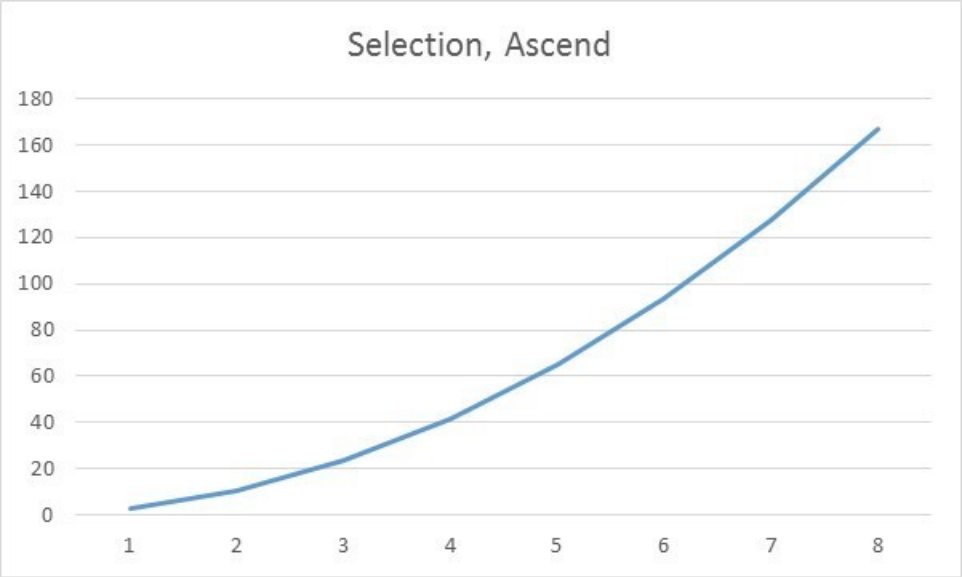


This worst case shows a shoddy example of O(nlog(n)). 10,000,000 would be 1 second to 1.2 seconds.

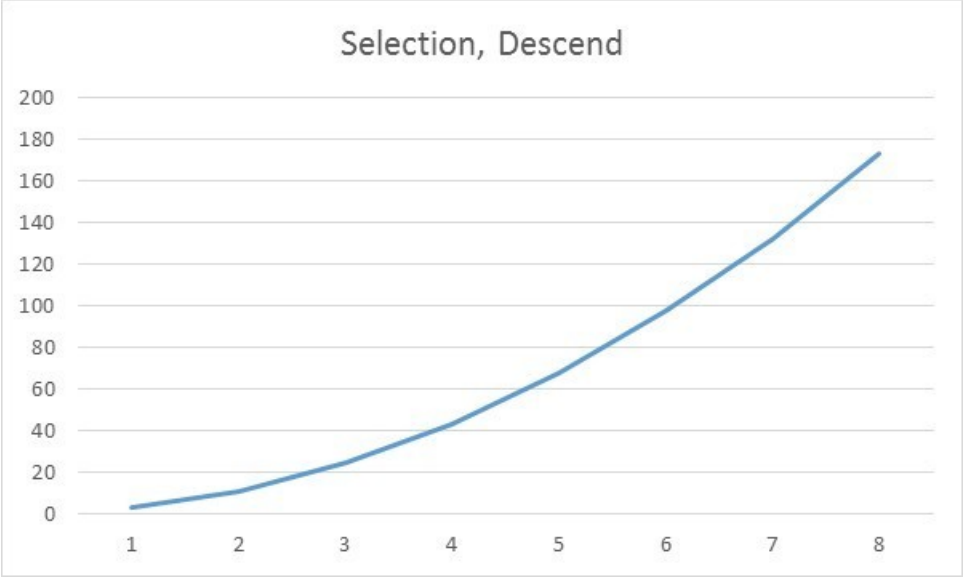


Althoug it looks like a linear equation, the graph should be nlog(n). 10,000,000 would probably be about 1.5 seconds.

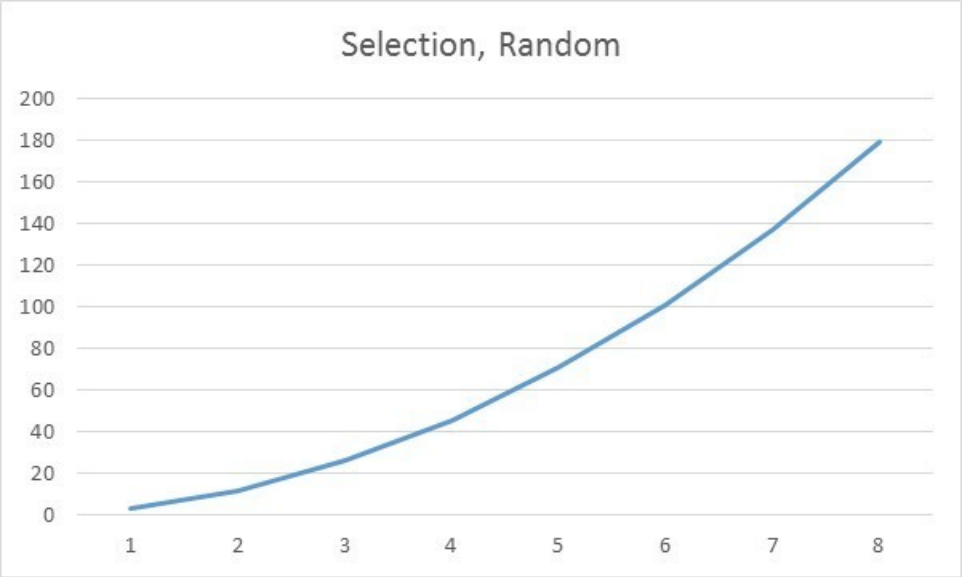
SELECTION



The graph for best case is n^2 rightfully so. 10,000,000 would probably result in about 500 seconds.



The worst case's graph shows n^2 as well. 10,000,000 would probably be about 600 seconds. Only a little more than the best case.



This is the average case. It is obviously n^2 as well. 10,000,000 should result somewhere between the best and worst case. Thus, somwhere between an estimate of 500 and 600 seconds.

P.S. apologies for not enough analyzation