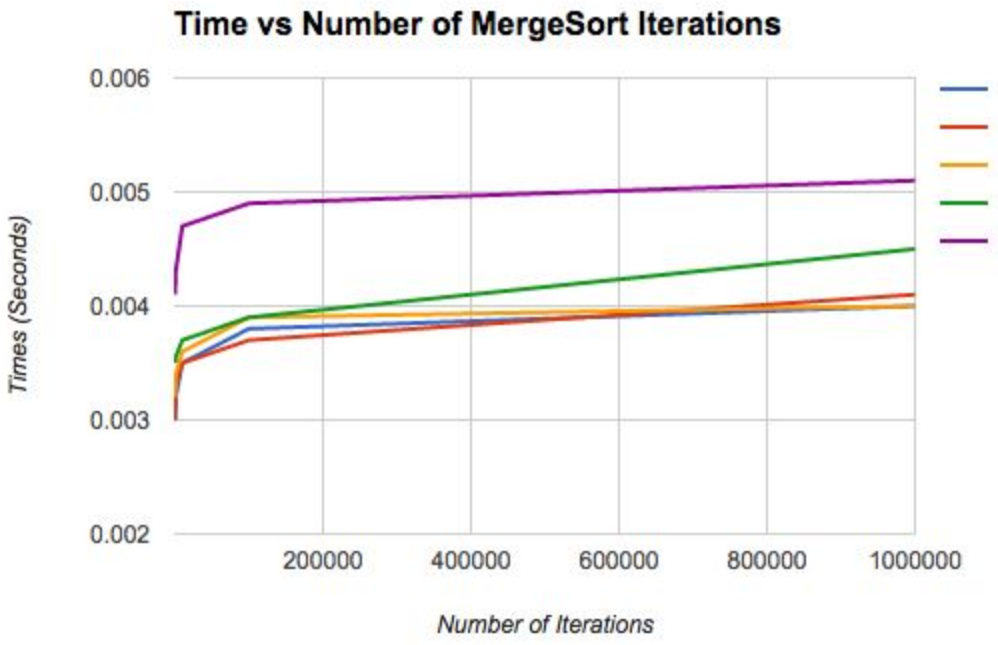
Will Stout
u10116888

1. Sean Scanlon is my partner, he will be submitting our program

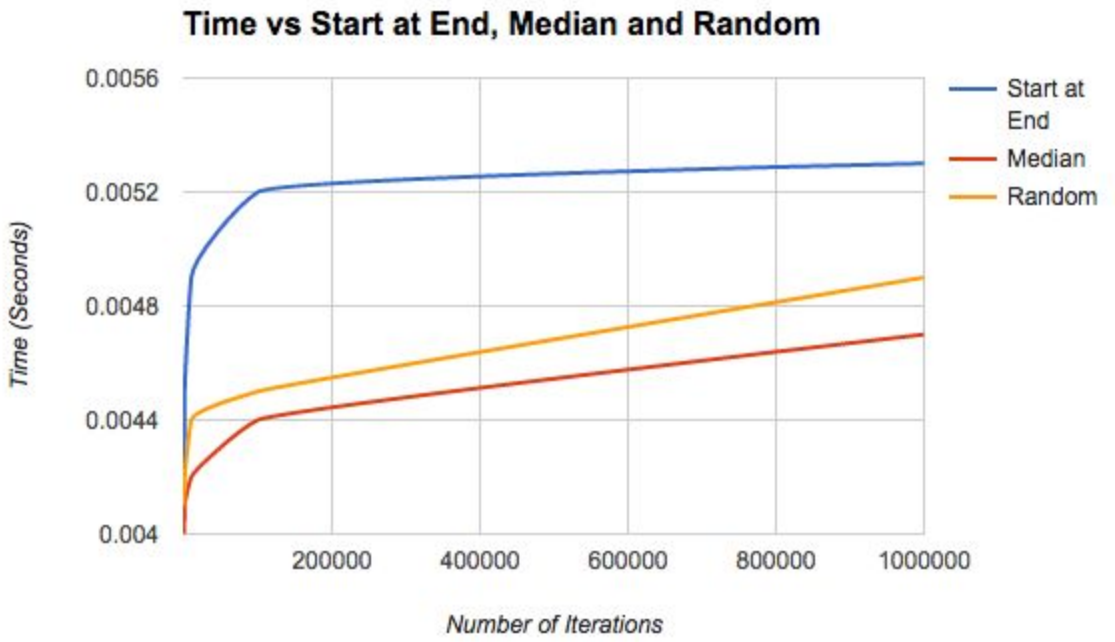
2.He is a hard working, and saves me when I need it. Good team player and one big plus is that he is very humble.

3. We work well together but could use better communication. He is diligent in his work so when it comes time to explain his work to me, he does so very proficiently. In order to work better together we could be more open about what and when we want to work on the assignment.

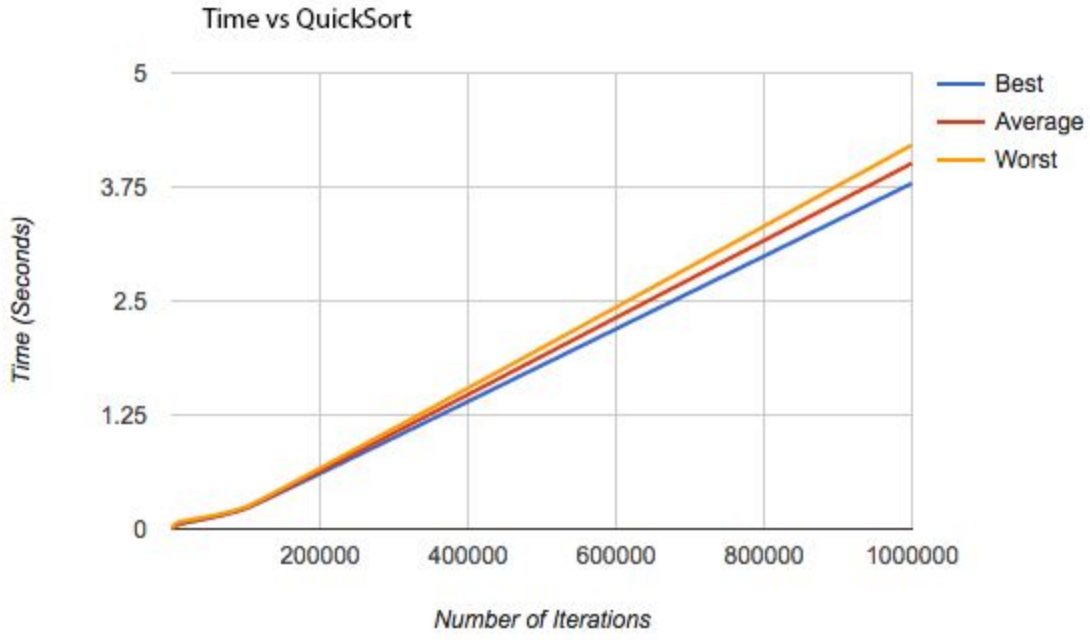
4. The merge sort experiment shows that even though merge sort has the better average case performance than insertion sort, it performs worse when n is small. So in this case it is best to switch over to insertion sort, which handles small n very well compared to merge sort.

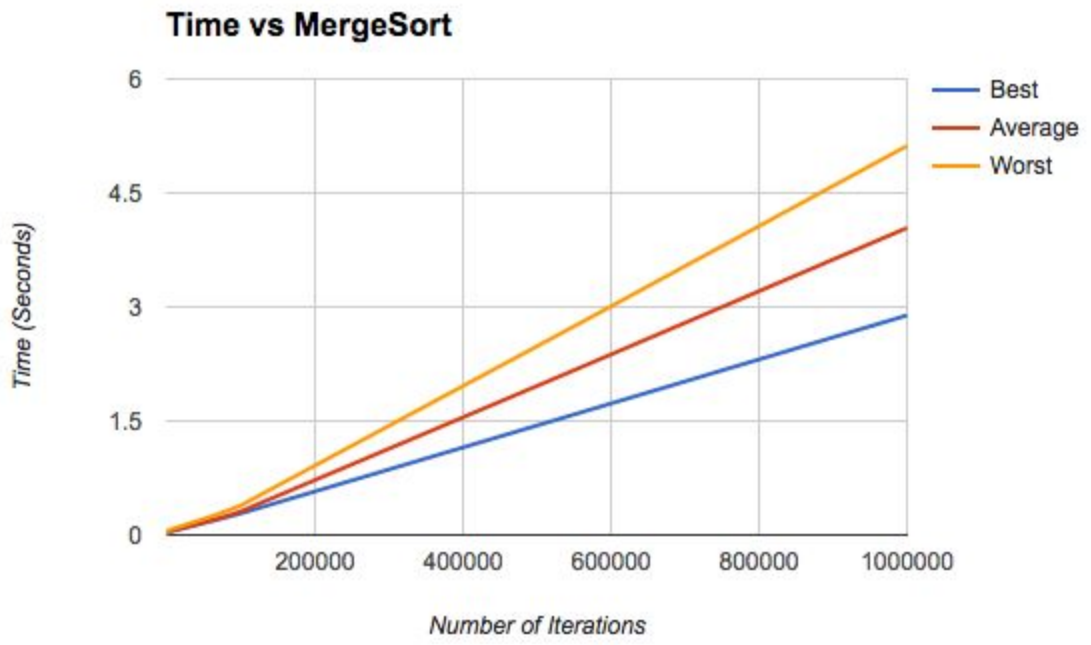


5. The best pivot will be determined by what kind of list there is. One of the ways to help mostly avoid running into the worst case pivot (which is choosing an extreme), is to choose a random pivot. While this is good for reducing the chances of running into worst case, it is a slightly less reliable method of choosing a pivot. Another way of choosing a pivot is to take the median of three. This makes it far more unlikely that the worst casescenario is run into, although if comparisons are more costly they can eat up time. A final way of choosing a pivot is to just run from the back, this is unadvisable as it can easily run into worst case conditions, which is O(n^2).



6. I have found that through my plotting that since quicksort tends to stay slow than quicksort until going and testing mergesort's worst case, which performs the worst out of any of the 6 cases. But on average mergesort out performs quicksort.





7. Actual running times are close to growth rates but definitely not exact. Because big O notation only cares about the largest order of n, all constants are thrown away. So even if we were to run into the average case for quicksort, n*logn , there is the possibility that actual runtime looks more like 4*n*logn.

8. Roughly twenty five hours.