This project had a lot of variables. These results are based on my computers capabilities and my choice of code implementation. For this project we had four different algorithms and where looking for how long will it take these algorithms to make the pivot index equal the given k. The four algorithms where merge sort, quick sort by recursion, quick sort by iterative, and the MM rule. The value k was calculated using the given formulas k = 1, n/4, n/2, 3n/4, and n. The minimum value for n was 10 and the maximum value was 1000.

I decided to fill the array with random values from 1 to 1000 since k can equal n. I wanted the possibility of k equaling a possible pivot index. I did not want the array full of small repetitive numbers but not too big, taking a lot of memory, for no reason. I made methods that did all the work so I can take the time from the main. As my code shows I simply called the appropriate method while passing the array, beginning index, end index, and the kth slot. I started the call before the call and ended it right after. Then I simply displayed my time, I recorded it, took the average, and made my data table.

To have an correct code, I also have return statements that end the sorting if ever the pivot equals k. After squashing some bugs I had carefully placed these statements in the appropriate location. If the kth slot equaled the pivot position then I have placed return statements that return all the way back to main to stop the timer. This is done because all we want is to find the kth slot. We were not sorting, we were searching. To obtain the most accurate results I ran this program ten times for each value of n. After many hours of program running, I have written the average in the table seen before my graphs.

From these results I made three tables. I compared all four algorithms, Select 2 vs. Select 3, and Select 4 vs. Select 1. When comparing quick sort recursively and iteratively, it’s no shocker that recursive is faster than iterative. In our level of study, this is a no brainer. But I did have to learn this new method, Select 4. While comparing Select 4 and Select 1, Select 4 wins as I highlighted in my tables. By choosing pivot more carefully, we can obtain a selection algorithm with Worst-Case complexity of O(n) while merge sort (Select 4) has a Worst-Case complexity of O(nlogn).