Bill Parrott
EECS 660, HW5
Select Algorithm Report

I chose to implement the k-select algorithm in PHP because of my familiarity with the language and its similarity to C++.

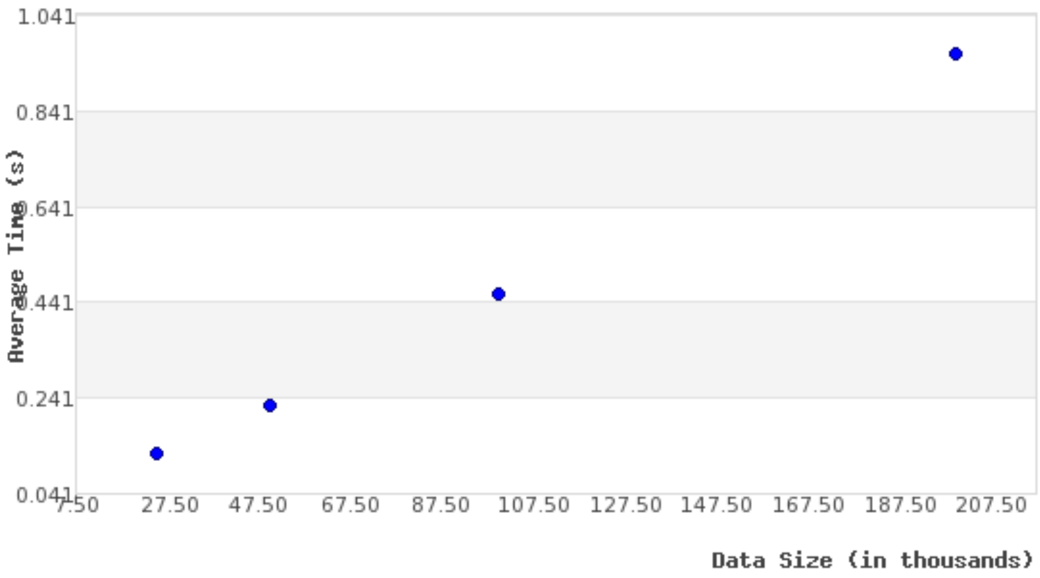
To run the program on the cycle servers, type “php select.php <args>”, where the <args> are chosen from the valid arguments that can be passed to the program. To see a list of valid arguments, type “php select.php --help”.

When designing the algorithm, I abstracted out some of the repeated sections, such as partitioning and insertion sort, into their own functions. This kept the main select() function to a manageable size.

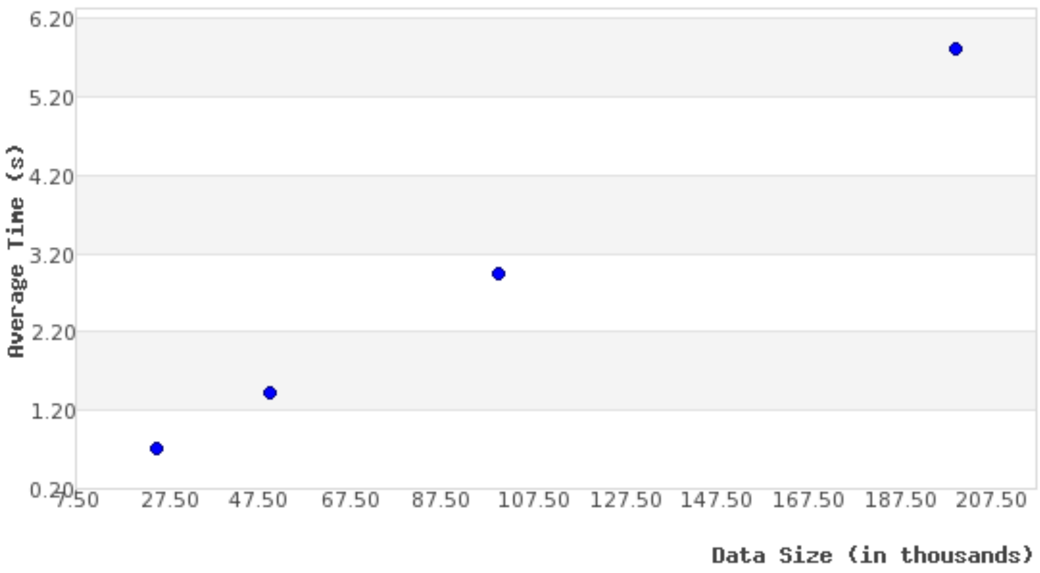
The biggest issue I ran into while debugging the program was dealing with the changing values of $left, $right, and $k. Because the program recursively called the select() function to choose both pivot values as well as the final k-th item, these variables changed over the course of the program’s execution. This made things somewhat difficult during the early development stages, because I had to tweak the way the array indices were passed with each recursive call.

The results of my tests show a linear relationship between the number of items in the array and the time it took to find the k-th item. This corresponds to the O(n log n) complexity of the algorithm.

The complete results of the tests can be found in output.txt and output2.txt. The first file is the result of tests performed on my personal laptop, while the second is the result of tests performed on the cycle server. Each has corresponding graphs with the results of the tests. The graphs containing the average values are included below.



K-Select Algorithm Average Timings (Cycle Server)



K-Select Algorithm Average Timings