Gregory Jerian

12/3/15

Period 4

Median Finder Readme

OVERVIEW:

This program finds the median of an array in O(n) time. It works based on the idea that if a random value is chosen as a “pivot”, and three subarrays are drawn up of the values greater than, less than, and equal to the pivot, then the largest array contains the median. It then takes this array and does the same thing many times (through recursive calls) until the amount of values greater than the pivot is equal to the number of values less than the pivot. At this point, it is known that the pivot is the median.

This is similar to quicksort as quicksort utilizes the same concepts of breaking up an array into many subarrays based on their value, except in the case of quicksort it is to SORT the array. In this case, it is instead to find the median value. This “modified quicksort” way of finding medians is much faster than sorting an array and finding the middle value, which would take O(n log n) time, and while this optimization is not easily seen with arrays of 6 or 7 values it is much more apparent with arrays of several million or billion data points.

ERRORS:

With an even array, the program chooses the “lower” median out of the two possible instead of averaging the two like it should. This would have been very hard to implement with the recursive formula used and I could not figure out how to do it (even after talking with Rafi, who ran into the same problem).

CREDITS:

<http://java2novice.com/java-sorting-algorithms/quick-sort/> for quicksort code.

Rafi Long for explaining to me how the median finding algorithm works. Thanks Rafi!