CSCI 230 -- PA 5

Selection Problem

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab.

You can work with a lab partner and each one must submit the same PDF file (include both names in the submission file). Each person must include a brief statement about your contribution to this assignment.

You must use an appropriate provided template from Canvas and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Exercise 1: Modify selection sort to find the k^{th} smallest element (do not sort the whole list). Output both k^{th} smallest element and number of compares. Test it with a small file with a few integers first like n = 10 (try k = 1, k = n/2, and k = n) and then try it to find the medians (k = n/2) of the two data files from previous PA, *small1k.txt* and *large100k.txt*.

Exercise 2: Implement the randomized quick select algorithm to find k^{th} smallest element as discussed in lecture/book. Output both k^{th} smallest element and number of compares. Test it with a small file with a few integers first like n = 10 (try k = 1, k = n/2, and k = n) and then try it to find the medians (k = n/2) of the two data files from previous PA, *small1k.txt* and *large100k.txt*.

Question 1: Provide at least two good reasons that you might not want to sort the data to determine the median value.

Question 2: Explain how the prune-and search design pattern works. Is recursion always necessary? Explain.

Extra Credit: Output run times in ms or ns for exercise 2 and confirm that it should be roughly linear by comparing run times for the 1k file and 100k file. Determine the maximum depth of recursive calls (not number of recursive calls) for exercise 2 and print it out. Confirm that it should be O(logn).

Fill out and turn in the PA submission file for this assignment (save as PDF format)