CSCI 230 Summer 2019

Due, Saturday, July 20, 11:59 pm

HW 4, 100 points

In this assignment you will do an empirical study of the performance of the quick sort algorithm, when different pivot values are chosen.

Create an integer array of size 5,000. Fill in the array with random numbers. Apply the quick sort with each of the each of the following pivot selections to sort the array.

1) Quicksort with pivot as the middle element, as demonstrated in our text
2) Quicksort with pivot as the median of the first, last and middle elements of the array
3) Quicksort with pivot as a random element of the array (use the Random class to choose a random index)

Before applying each algorithm, initialize the array to its original state (randomly filled values) by saving a copy of the original array.

After applying each sort function printout the first 20 elements of the array to make sure that it is really sorted (your sort algorithm is working correctly).

Also, measure and record the (CPU) time each each algorithm takes. Look into Java's

Repeat this process for arrays of size i* 5,000 for i =1, 2, 4, 6, 10

Have your program generate the times, and create an easy to read table with the data. Make sure your name is on this document, and generate it in pdf and name it HW4.pdf

To submit: Create a folder in the usual way. Include your Quicksort.java files (all code in one file) and your HW4.pdf document. Zip the folder and upload to OAKS by the due date and time.

Note: the textbook version of the code is provided. The partition algorithm given chooses the middle element as the pivot.

You can duplicate the code and have multiple quicksort and partition methods if you wish, or organize in some other clever way. Just clearly document your code so we can see what you did.