CS 3130 Design and Analysis of Algorithms Spring 2019

Programming Project #2 [90 points].

Due date: Thursday, March 21.

The purpose of the programming assignment is to perform empirical analysis of the following sorting algorithms on integer arrays:

(a) selection sort;
(b) insertion sort;
(c) two versions of a bubble sort: with and without counting the number of swaps on each pass through an array;
(d) quicksort;
(e) mergesort.

Requirements:

- functions from standard libraries implementing sorting algorithms are NOT allowed; however, you can use any code from any other sources;
- for obtaining the working time of each algorithm for a particular array, use the same function as for part (C) of Project #1; call this function before and after a call to the function implementing a sorting algorithm;
- run your functions for the following types of arrays containing 1000, 10000 and 100000 integers: random numbers, sorted list, almost sorted list (say, each 10th number is out of order);
- the size of integers is supposed to be from 1 to 10000;
- you may store your (non-random) arrays in the FILES so that you won't need to create them every time when you call your functions;
- submit: (1) source code with the results; (2) the analysis of your experiments.
- requirements to the analysis: (a) the text must be TYPED and submitted electronically or in person before the class on Thursday, 03/21; (b) your analysis must include theoretical information about the efficiency of each sorting algorithm; (c) experimental results must be clearly presented in the form of the table or graphs; (d) I expect to see your conclusions on how well the experimental results correspond to the theory, and which sorting algorithms work better for specific types of input [**analysis itself will be 15 points out of 90**];
- *please, do NOT include original and sorted arrays in your output!*
- Your source code MUST include information on the name of the programmer and the purpose of the project, as well as some comments.