CSCI 232 - Lab 8: Sorting

Part 1 – Sorting Algorithm Benchmark

Create a module called “benchmark”. Inside, provide implementations for the following sorting algorithms (you can use the code provided for these – you may have to make some small tweaks like removing debugging statements):

- Bubble Sort
- Selection Sort
- Insertion Sort
- Shell Sort
- Merge Sort
- Quick Sort

To test these algorithms, create a list of 500 random integers. You can either create each number randomly and put them in a list, or you can create a list from 1 to 500 and use the shuffle function (from the random module) in Python.

Much like our Big-O lab from earlier this semester, use the timeit module to find out how long it takes to run each sorting algorithm 10 times. Print the results to the console after running each sorting algorithm on a mixed set of numbers.

Part 2 – Bi-Directional Bubble Sort

Create a second module called bd_bubblesort.

Implement a bi-directional bubble sort that “bubbles” in two directions. The original bubble sort algorithm can be modified to scan the list from right-to-left as well as left-to-right. After a relatively high value bubbles to the right, a relatively low value will bubble to its appropriate location on the left. In other words, bi-directional bubble sort scans in both directions, alternately.

Test the speed of this algorithm by running it separately from the other benchmarks.

Grades:

Bi-directional bubble sort – 40

Sorting algorithms – 60 (10 points each)