Program Structure and Algorithm

Assignment 3 Benchmark

Shouting Lyu

001822094

In general, I completed 3 parts of the assignment: Benchmark, InsertionSort and SelectionSort, and Measuring the running time of two sorting algorithms with random, ordered, reverse-ordered and partial-ordered arrays. In detail, I run 10 times before starting to calculate the running time of each sorting method, and then run 100 times and finally calculate the average time for 7 different values of n: 50, 100, 200, 400, 800, 1600, 3200.

The figures show the ‘run()’ method and three shuffle method. In the shuffle method, I used Knuth algorithm to guarantee the array is randomly shuffled, and they are in the ‘Helper’ class. As you can see, I implemented the run method and call the function 10 times before the real experiment, and I used ‘nanoTime()’ to calculate the average running time.

A screenshot of a cell phone screen with text

Description generated with very high confidenceA picture containing clock

Description generated with high confidence

The other four figures below show the comparison of two sorting algorithms with different inputs. As you can see, the advantages of insertion sorting are obvious with ordered and partial ordered arrays. Then, the insertion sort is still faster than selection sort, though the efficiency is not as perfect as sorting the ordered array. In the worst case, while sorting a reverse-ordered array, the insertion sort is close to the selection sort.

A close up of a map

Description generated with high confidenceA close up of a map

Description generated with high confidenceA close up of a map

Description generated with high confidenceA close up of a map

Description generated with high confidence

At last, the following figure shows my passing the unit test.

A screenshot of a cell phone screen with text

Description generated with very high confidence