

LAB 2 Write Up

Romeo Perlstein, section 0102

6.1 - Sorting

From analyzing the table, you are able to extract which sort methods are faster than the other. As we discussed in class, `Bubble Sort` is the slowest out of the 4, with an average run time of 5 seconds for the 10,000 value array. However, for smaller array's, such as the 8 index or 100 index, the time difference is rather negligible between each sorting method. In fact, almost all of the sorting algorithms complete sorting the 8 index array in less than 1 second. You'll also notice that even for the higher index array's, The `Insertion Sort` method remains is quite fast.

These results were concluded by sorting 100 different arrays (for each index length), for each sorting method. A conclusion that could be drawn from analyzing the table would be that `Bubble Sort` and `Selection Sort` are the slowest out of the two sorting methods, with `Insertion Sort` being the fastest

The output file, named `timeFile.csv`, can be found in the working directory after running the program

6.2

This program works quite simple. The input argument is a filepath to your file of unsorted numbers. The program will check to see if the first line of the page is a number, or if it is some sort of header, and then act accordingly. The program will then read in the contents of the file into an array, and pass that array to the `merge_sort()` Function

6.3

This program is somewhat straight forward as well. Utilizing two dictionaries, one containing letters as keys with their corresponding values as numbers, and one containing numbers as keys with their corresponding values as letters. This allows us to search for the number of a letter, and search for the letter of a number, quite easily. It also let's up check to see if the character of a string is a letter, or if it is some sort of other ascii character. We check each character of the string until we've constructed a new encrypting string using the Vigenere Cipher method

6.3 - Challenge

This program utilized the same algorithm as the one implemented in `vigenere()`, except now, the number of our character corresponds to the original numbers + the shifted key number, so we simple subtract the shifted key number to get our original number. We can then reconstruct the array as follows. In order to find the write hidden phrase, we use common english words like `the`, `it`, `and`, `a` and others to try and pinpoint the phrase.

After I found the phrase, I adjusted some of the search string to match words in the phrase, to guarentee the code would find it

6.4

This program initializes takes two `Runner` classes as inputs, and races them against each other. In a while loop, the distance traveled is calculated by multiplying the given speed of the animal, and the time (being tracked through an integer) it's been since starting the loop. Sprinkled throughout are print statements that makes the race feel a little less robotic and more realistic, as well as utilization of the `speak()` property, where applicable