CS2123 Data Structures - Spring 2020

Assignment 1: Function Runtimes Table

Due 2/7/20 by 11:59pm

Completing the Program (15 points)

This program prints a table of runtimes (these are displayed in seconds) for given functions on arrays. The program tests different array sizes to establish a relationship between input size and runtime. It tests each array size multiple times and then takes an average of the times. Here are example calls to the timing functions:

```
int sizes[] = { 1000, 2500, 5000, 7500, 10000};
char str1[] = "Insertion Sort";
char str2[] = "quicksort (uses insertion sort when sorting <30 numbers)";

fRT = timeAlgorithm(str1, 10, 5, sizes, insertionSortInitial );
printRuntimeTable(fRT);
freeFunctionRuntimes(fRT);

fRT = timeAlgorithm(str2, 10, 5, sizes, quickSortOptInitial );
printRuntimeTable(fRT);
freeFunctionRuntimes(fRT);
```

This results in following table:

```
Insertion Sort
Test size Test #0   Test #1   Test #2   Test #3   Test #4   Test #5   Test #6   Test #7   Test #8   Test #9   Average
10000     0.223     0.214     0.216     0.213     0.220     0.240     0.212     0.212     0.212     0.214     0.218
25000     1.457     1.405     1.436     1.426     1.408     1.410     1.405     1.466     1.389     1.341     1.414
50000     5.385     5.356     5.347     5.356     5.356     5.370     5.354     5.358     5.351     5.371     5.360
75000     12.071    12.099    12.168    12.210    13.097    13.214    13.122    13.064    12.403    12.093    12.554
100000    23.288    22.105    21.737    21.624    21.616    21.669    23.189    23.144    23.176    21.710    22.326

quicksort (uses insertion sort when sorting <30 numbers)
Test size Test #0   Test #1   Test #2   Test #3   Test #4   Test #5   Test #6   Test #7   Test #8   Test #9   Average
10000     0.002     0.003     0.002     0.003     0.002     0.002     0.002     0.003     0.004     0.003     0.003
25000     0.007     0.008     0.007     0.007     0.007     0.007     0.007     0.007     0.007     0.007     0.007
50000     0.014     0.014     0.014     0.014     0.014     0.014     0.015     0.015     0.014     0.014     0.014
75000     0.020     0.021     0.025     0.022     0.021     0.023     0.022     0.027     0.024     0.023     0.023
100000    0.032     0.043     0.029     0.028     0.028     0.029     0.031     0.029     0.030     0.029     0.031
```

Note your runtimes may vary since the test data is randomly generated.

The runtimes are stored in a functionRuntimes struct. You are completing a program to create and fill data in this struct, print the data of this struct, and free this struct.

You are given a partial implementation in the fle “cs2123HW1-LastName.c”. Specifically you are tasked to complete the funtctions below the heading “Functions for finding and printing runtime”. No other functions should be changed.

Using the Program (5 points)

After you have the program completed, you should use it to help determine the asymptotic runtimes of the three mystery functions (i.e., mysteryRuntime1, mysteryRuntime2, mysteryRuntime3).

Be sure to also examine the code of the mystery functions to confirm/improve your estimations.

Fill in the following table in the provided file:

```
/*
TODO: Give your asymptotic estimates for the runtimes of the following 3 functions:

mysteryRuntime1:  O(   )
mysteryRuntime2:  O(   )
mysteryRuntime3:  O(   )
*/
```

Deliverables:

Your solution should be submitted as "cs2123HW1-LastName.c" where "LastName" is replaced with your last name. Be sure to fill in the table of runtimes described above:

Upload this file to Blackboard under Assignment 1. **Do not zip your file**.

To receive full credit, your code must compile and execute. You should use valgrind to ensure that you do not have any memory leaks.

Remember:

The program you submit should be the work of only you. Cheating will be reported to SCCS. Both the copier and copiee will be held responsible.