CS111-Design and Analysis of Algorithms
2nd Semester, 2021-2022

Programming Project

Use the different sorting algorithms (Insertion Sort, Mergesort, Quicksort, Heapsort) to sort N randomly generated integers and then analyze their run time. The project will have two parts, one for the basic implementation of the program that sorts the integers, and second for analyzing what you see in terms of performance.

A: Desired Program Behavior:

1. User is prompted to enter a value for N (for number of integers) as input
2. The program generates N integers which are either
   a) all integers are picked randomly with a uniform distribution from the range $[0, MAXRANGE]$ (use MAXRANGE = 1000000) or
   b) sorted in the increasing order, wherein in addition to N, the program should also ask an input X (a positive number) from the user and set the first element of the array to $N+X$, the second element to $N+2X$ and so on (this will generate an increasing sequence of N numbers).
3. The program sorts these N integers using the different algorithms and outputs a file with original and sorted values.
4. The program also outputs the computation time $T(N)$ for sorting N integers. Be sure that this time includes only the computation time and not the time spent interacting with user and or generating the integers.
5. You may use either **C or C++** as a programming language for implementation. However, this program should **use standard functions** since I will be compiling and running it on my Linux system. Adequate comments in the code are expected.
6. The different algorithms can be implemented as separate functions in the program passing the input array as parameters to these functions.
7. To calculate time taken by a process or set of operations, you can use clock() function which is available in time.h. You can call the clock function at the beginning and end of the code for which you will measure time, subtract the values, and then divide by CLOCKS_PER_SEC (the number of clock ticks per second) to get processor time, like following:

```
#include <time.h>

clock_t start, end;
double cpu_time_used;

start = clock();
... /* Do the work. */
end = clock();
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
```

B: **Analysis**

1. Compute the average (over five different runs) of the actual running time $T(N)$ for five different N values, $N=10,100,1000,10000,100000$. Integers are to be picked randomly from [0, MAX_RANGE] (Part A.2.a above, i.e. input array is random). For example, for $N=100$, run your program "five times" to get five values of $T(N)$. Then take the average of these readings to get average $T(N)$ for $N=100$.
2. Repeat (1) using Part A.2.b as above (i.e., input array is sorted) to compute corresponding running time $T'(N)$ for $N=10,100,1000,10000,100000$.
3. Present the results in Table format, where rows of the tables will indicate different N values and columns will indicate the $T(N)$ for the different sorting algorithm.

Ex.

Average Running Time for an Input Array that is Random

N	Insertion sort	Mergesort	Quicksort	Heapsort
10				
100				
1000				
10000				
100000				

Average Running Time for an Input Array that is Sorted

N	Insertion sort	Mergesort	Quicksort	Heapsort
10				
100				
1000				
10000				
100000				

To be submitted are the following:

- Source code in C or C++
- A very detailed written report which contains the following:
  - documentation of your program describing the structure of your code
  - screenshots showing the actual exectution of the program
  - analysis of the outputs as specified above. Discuss the results of your analysis and provide your conclusion.
  - challenges you have encountered and the participation/contribution of each member in making the project

Note: Submissions that will include source code or program only will not be accepted. You will also be asked to demonstrate or defend your program.