COP3502C Computer Science 1
Sorting Algorithm Runtime Experiments

In this lab, you will need to write a c program and create an excel file. You will need to submit both your code and the excel file in the webcourses.

Experimental Study of the Execution Times of the Sorting Algorithms

In this lab, you will perform experimental studies on the sorting algorithms we have learned so far. You are asked to compare the execution times of the following sorting algorithms:

selection sort, bubble sort, insertion sort, merge sort, and quick sort

All of these algorithm source codes in c are available in your main course webcourses. So, copy them as you need.

In order to do the comparison, you will need to:

1) Create an array and fill it with random numbers
2) Pass this array to each sorting algorithm (copy the array before passing as the sorting algorithm going to change your array)
3) ***Write down the execution time for each algorithm in an excel file and plot them to see how the execution time changes with time.***

A good idea could be creating an array containing the total number of data you want to produce at each phase. For example, you can create an array just to store the sizes of eeach array you are going to create. Forexample, sizes array can include [1000, 10000, 20000, 40000, 50000, 100000]. This carray can be useful to generate 1000 integers, 10K integers, and so on.

You may find the following hints useful:

1) To generate random numbers between [1,maxVal] you can use the rand function:

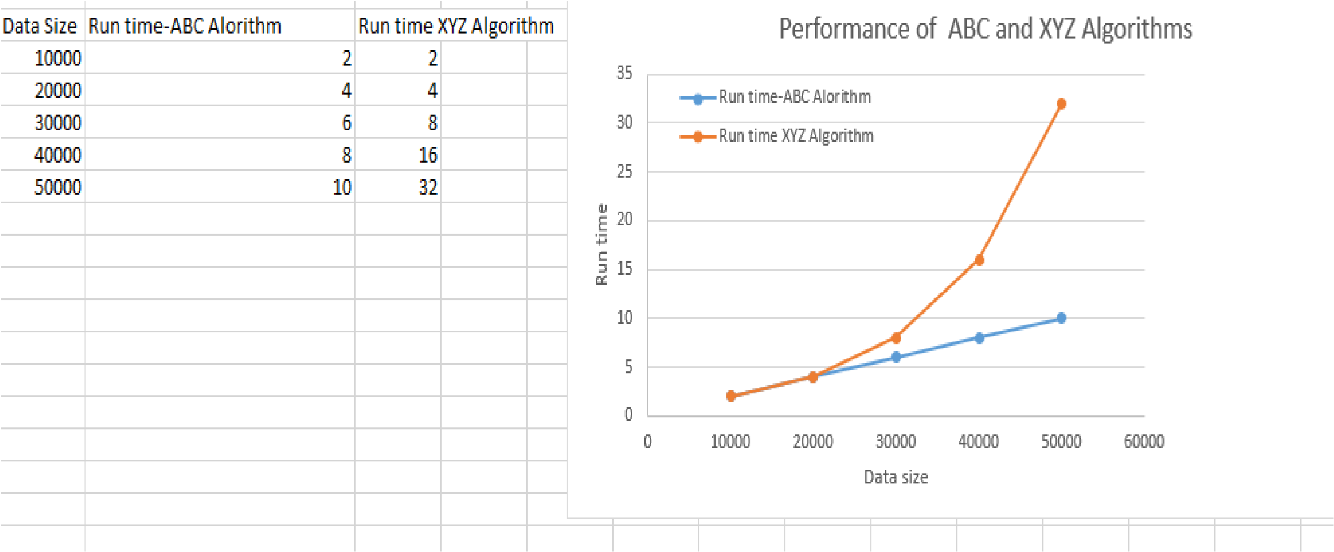
n = rand() % maxVal + 1;

Just put this line in your for loop and store n in the array. Don’t forget to include #include<stdlib.h>

2) To calculate the execution time you can use <time.h> library and the following technique:

```
    int start = time(0);
    bubbleSort(sortedArray, sizes[i]);
   int end = time(0);
  printf("Sorting %d values took %d milli seconds for Bubble sort.\n", sizes[i], timediff(start, end));
//timediff unction should be defined by us and you can simply copy the function from here:
--------------------------------------------
long timediff(clock_t t1, clock_t t2) {
   long elapsed;
   elapsed = ((double)t2 - t1) / CLOCKS_PER_SEC * 1000;
   return elapsed;
```

An example excel file content should look like this:



Data Size	Run time-ABC Alorithm	Run time XYZ Algorithm
10000	2	2
20000	4	4
30000	6	8
40000	8	16
50000	10	32