COS 364 Project SimulateQuickSort - COS364SimulateQuickSortProject2020.pdf
Due Date Thursday October 8 By 5 PM

You must work on this project alone. You may discuss the general ideas.

The goal of this project is to write a simulation to obtain the average case complexity of quick sort. The output from your code should include:

1. Values of n (size of the array to sort). The values should be 10, 50, 100, 500, 1000, 5000, 10000, 25000, 50000, 75000, 100000, 200000, 300000, 400000, and 500000.
2. Values of the average number of **exchanges** for each of the above n's. The **exchange** is the statement
   exchange (S[i] and S[j])
3. Values of the average number of exchanges divided by n for each of the above n's.

After obtaining the above outputs, plot the data points (n, average/n) for the above n's using any graphing utility you have. I used an online utility at

https://itools.subhashbose.com/grapher/index.php

For the actual quicksort part, please refer to the outline for QuickSort in the pdfs on my web page. These pdfs are QuickSortPartI2020.pdf and QuickSortPartII2020.pdf. Follow the latter pdf to structure the basic code for quick sort **except** for the following:

(1) For the partition method, use pivotpoint as the third parameter (using the keyword **out** to pass out any changes). In addition, omit the last line in the outline for partition since the partition method now returns **void**. This means omit the line: **return pivotpoint;**

private static void partition(**long** low, **long** high, out **long** pivotpoint)

Inside the partition method, declare your local variables to be of type **long**:

(2) For the quicksort method use:

private static void quicksort(**long** low, **long** high)

Inside the quicksort method, declare pivotpoint to be of type **long**:

Inside the quicksort method to call partition, replace the line in the outline with:

partition(low, high, out pivotpoint);

Note that your program will not read from a file nor write to a file. You capture the output from the console and paste it into a text editor. Then you type the data points (n, average/n) into your graphing utility.

As class members you will want to declare a **long** array S (for the data) and and a long array C (for the number of exchanges). This is because some of the integers we use (such as 500000) are relatively large. Also as a class member, you will have a **long** variable called count. How is count used? In the function partition, count will keep track of the number of exchanges for a run of partition. Thus, you need to add a line in partition to accomplish this. Then in the main, an element of the C array will get assigned count. Also have a class member that is of type Random. Decide on other class members.

In the main, you will:

1. Set up an array holding the above values from 10 to 500000 as follows:
   int [ ] lim = { 10, 50, 100, 500, 1000, 5000, 10000, 25000, 50000, 75000, 100000, 200000, 300000, 400000, 500000};
   and use the property of an array called Length as in lim.Length.
2. Use a random number generator to fill the array S each time. If rnd is of type Random, then you assign a variable to the next random number where you get the random number as follows:
   rnd.Next(n);
   where n is one of the elements in the lim array.
3. For a given n, you do quicksort 50 times, each time of course filling the S array with new random integers. For each of these 50 times, you keep track of the number of exchanges in the C array.

What to turn in:

(1) Share or email me a copy of your C# program (the .cs file)

(2) Turn in your output from your program.

(3) Turn in your graph.

(4) Turn in a writeup that:

   a. states the function that you think your plot represents and why;

   b. explain what this function has to do with the average case complexity of quicksort;

   c. explain how you designed the simulation part of your program. This paper should be at least 2 pages (8.5 by 11 inches average font).