DSA Refresher Module

Assignment-1

Max Marks: 40 **Due Date**: 28/07/2024, 11:59 PM

Instructions

- Use C++ programming language only.
- Plagiarism might be checked.
- Assignments can be done individually or in pair
- Upload a Zip folder named <RollNo>_<YourName> if you are doing individually. If you are doing it in pair use the naming format as <RollNo1>_<RollNo2>.zip. If not named appropriately, itwill not be evaluated.
- For the coding questions, you must ensure that your code compiles and runs otherwise these will be awarded **zero marks**.
- Students might be required to give a demo of their codes.
- Inside the Zip folder, there should be files named<RollNo>_<RollNo2>_Sec1_Q1.cpp, <RollNo>_<RollNo2>_Sec1_Q2.cpp, ….., <RollNo1>_<RollNo2>.pdf.
- <RollNo>_<YourName>.pdf - Report including answers which require theoretical answer
- For all Coding Question, do not include object files.

Section 1 (15 Marks):

1. **(7 points)** (a) (3 marks) Write a code for the Tower of Hanoi Algorithm using two temporary pegs I.e. There is a source pole/peg named T1, two temporary pegs named T2 and T3; a destination pole/peg named T4. Take the number of disks as input from the user.

   (b) (2 marks) Give the sequence of moves required to move 5 disks from source to destination, using the above algorithm. Explain how the code is working with 5 disks.

   (c) (2 marks) Compare the answer you have received in Q1.2 with the answer you will receive with the traditional Tower of Hanoi Setup (1 source pole, 1 temporary pole, 1 destination pole)I.e. compute and compare their time complexities.

```
Output format for Q1.1
>> Enter the number of disks : 1
>> The sequence of steps are:-
T1 -> T4
```

```
>> Enter the number of disks : 2
>> The sequence of steps are:-
T1 -> T2
T1 -> T4
T2 -> T4

>> Enter the number of disks : 3
>> The sequence of steps are:-
T1 -> T2
T1 -> T3
T1 -> T4
T3 -> T4
T2 -> T4
```

2. **(3 points)** (a) (2 marks) Write a recursive and an iterative code for Merge sort. Take the array to be sorted as input from the user. Assume it to be an integer array only. Print the sorted array from both recursive call and iterative call.

   (b) (1 marks) Explain the iterative code by an example.

```
Output format for Q2.1
>> Enter the array to be sorted with elements separated by comma: 32,42,1,4,32,15,6
Sorted array by Recursion is : 1,4,6,15,32,42
Sorted array by Iteration is : 1,4,6,15,32,42
```

3. **(5 points)** (a) (2 marks) Shinigami took Light Yagami's special notebook and issued a challenge. To get it back, Light needs to solve the "Min Max" problem. Light's task: Given an array of 'N' integers, determine the maximum values obtained by taking the minimum element over all possible subarrays of varying sizes from 1 to 'N'.

   (b) (3 marks) Help Light Yagami to solve this problem in **O(n)** time.

```
Consider an array, Arr = [3, 1, 2, 4].
For subarrays of size 1: max(min(3), min(1), min(2), min(4)) = 4
For subarrays of size 2: max(min(3, 1), min(1, 2), min(2, 4)) = 2
For subarrays of size 3: max(min(3, 1, 2), min(1, 2, 4)) = 1
For subarrays of size 4: max(min(3, 1, 2, 4)) = 1

The resulting array is [4, 2, 1, 1].
```

Section 2 (25 Marks):

1. **(15 points)** For the given three sorting algorithms: insertion sort, selection sort and merge sort select the most suitable sorting algorithm for the scenarios provided below and provide a rationale for your choice. Your justification carries more weight than your selection. Each sorting algorithm might be used more than once. If multiple sorting algorithms could fit a scenario, outline their advantages and disadvantages for each of them, and then pick the one that aligns best with the application. Clearly state and justify any assumptions you make and perform asymptotic analysis. "Best" should be evaluated based on **asymptotic running time**.

   (a) (5 marks) You have a custom data structure, *DS*1, that organizes k items and supports two operations:
   `DS1.get_at_index(j)` takes constant time, and
   `DS1.set_at_index(j, x)` takes $\Theta(k \log k)$ time.
   Your task is to sort the items in DS1 in-place.

   (b) (5 marks) Imagine you possess a fixed array A containing references to n comparable objects, where comparing pairs of objects takes $\Theta(\log n)$ time. Select the most suitable algorithm to sort the references in A in a manner that ensures the referenced objects appear in non-decreasing order.

   (c) (5 marks) Suppose you are given a sorted array A containing n integers, each of which fits into a single machine word. Now, suppose someone performs some $\log \log n$ swaps between pairs of adjacent items in A so that A is no longer sorted. Choose an algorithm to best re-sort the integers in A.

2. **(10 points)** (a) (5 marks) Implement stack and queue from scratch using pointers in C++. They should have all the functions supported by stacks and queues, e.g., insert, pop, top, etc. (Plagiarism will be strictly checked.)

   (b) (5 marks) Use your stack implementation from above and solve the following problem:- Given a circular integer array nums (i.e., the next element of nums[nums.length - 1] is nums[0]), return the next greater number for every element in nums. The next greater number of a number x is the first greater number to its traversing order next in the array, which means you could search circularly to find its next greater number.

```
Input: [1,2,3,4,3]
Output: [2,3,4,-1,4]
```