Department of Computer Science and Engineering
Indian Institute of Technology (ISM) Dhanbad

End-Semester Lab Examination, Session 2023-24

Course Name (Code): Data Structures Lab (CSC204) **Time: 1 Hr. 45Mins.**
Subject: III B. Tech. (CSE) **Max. Marks: 40**

Instructions:

1. Answer any **one** question from **Qs No. 1-2** and **Qs No. 3 is Compulsory**.
2. After completion of the examination, do the following:
   - Program file should be named as $< \boldsymbol{AdmissionNo\_QuestionNo} >$, e.g., **20JE0014_Q1**. Save your program files into a folder, named as $< \boldsymbol{AdmissionNo\_endSem} >$ and zip the folder.
   - Send the Zip folder to the e-mail id: dsalab2023@gmail.com.
3. Use of Internet is strictly prohibited. You must not copy code from the internet. Plagiarism of the submitted code will be checked and if your code found to be copied from any source, your exam will be cancelled. If your code is found copied from any other student, then both students will get **zero** marks.
4. Submit the code within the time limit.

Questions:

1. A class of students where details of each student is stored using a doubly linked list. Each node of the doubly linked list contains the information such as $Regn\ No.$ (in integer), $Name$ (in string), $Branch$ (in string), and $CGPA$ (in float) of a particular student. Your task in this problem is to create a doubly linked list which takes the input for at least 5 students and then perform the following task:

   a) Write a function $\boldsymbol{arrange\_ascend}()$ to arrange data of students according to $Regn\ No.$ using Insertion sort.

   Suggested Output Format:

$Regn\ No.$	$Name$	$Branch$	$CGPA$
2	John	CSE	7.2
5	Ram	ECE	8.0
7	Hari	ME	6.5
8	Raj	CSE	7.0
10	Ali	ECE	7.6

   b) Write a function $\boldsymbol{search\_data}()$ to display data of a student with a particular $Regn\ No.$

   Sample Input:

   Enter the $Regn\ No.$: 8

   Sample Output:

$Regn\ No.$	$Name$	$Branch$	$CGPA$
8	Raj	CSE	7.0

[15+5]

2. Given an adjacency matrix of a directed graph $G(V, E)$ where V and E represents the vertices and edges of the graph. The first row of the adjacency matrix indicates the all outgoing edges for 1st vertex. Complete the following tasks regarding the given adjacency matrix.

a) Write a function ***DFS_time***( ) to print the *discovery time* and *finishing time* for all the vertices while performing DFS traversal on the given graph. During the traversal, prioritize exploring the vertex with the lowest number whenever possible.

Sample Input:

Enter the number of vertices: 6

Enter the 1st row of the adjacency matrix: 0 1 1 0 0 0
Enter the 2nd row of the adjacency matrix: 0 0 0 1 0 0
Enter the 3rd row of the adjacency matrix: 0 0 0 1 1 0
Enter the 4th row of the adjacency matrix: 0 0 0 0 0 1
Enter the 5th row of the adjacency matrix: 0 0 0 1 0 0
Enter the 6th row of the adjacency matrix: 0 0 0 0 1 0

Enter the source vertex: 3

Sample Output:

1st Vertex: discovery time: 9; Finishing time: 12
2nd Vertex: discovery time: 10; Finishing time: 11
3rd Vertex: discovery time: 1; Finishing time: 8
4th Vertex: discovery time: 2; Finishing time: 7
5th Vertex: discovery time: 4; Finishing time: 5
6th Vertex: discovery time: 3; Finishing time: 6

b) Write a function ***Simple_path***( ) to check whether there exists any simple path between given pair of vertices or not. Your output should be either "YES" or "NO" as per the context.

Sample Input:

Enter the source vertex: 2
Enter the destination vertex: 5

Sample Output:

YES

[10+10]

3. Given two arrays of integers, create two different binary search trees by inserting the elements of those arrays from lower index to higher index. Then do the following task.

a) Write a function ***Similar_tree***( ) to determine whether two trees are structurally similar or not. In this context, "structurally similar" means that the two trees have the same organization of nodes, but the values of the nodes may differ. Your output should be either "YES" or "NO" as per the context.

Sample Input:

Enter the elements of the first array: $20, 15, 5, 30, 22, 32$
Enter the elements of the second array: $10, 25, 8, 23, 6, 30$

Sample Output:

YES

b) Write a function ***zigzag_fashion***( ) which will traverse any tree data structure in zigzag fashion. The zigzag traversal is defined as:

- Start from level 0 of the tree, then
- for level 1, the traversal direction will be from right-to-left, for level 2 this will be from left-to-right, for level 3 the traversal direction will be again from right-to-left, and so on.

Now, show the outputs of zigzag fashion traversal for the trees given in part (a). You are allowed to utilize two additional data structure, if required.

Sample Output:

Zigzag traversal of first tree is: $20, 30, 15, 5, 22, 32$
Zigzag traversal of the second tree is: $10, 25, 8, 6, 23, 30$

[10+10]