

JK LAKSHMIPAT UNIVERSITY, JAIPUR

INSTITUTE OF ENGINEERING AND TECHNOLOGY

End Term Examination December 2022

B. Tech in Computer Science Engineering, Semester III (CSE)

Roll No.....

CS1102: DATA STRUCTURES

Time: 03 hours

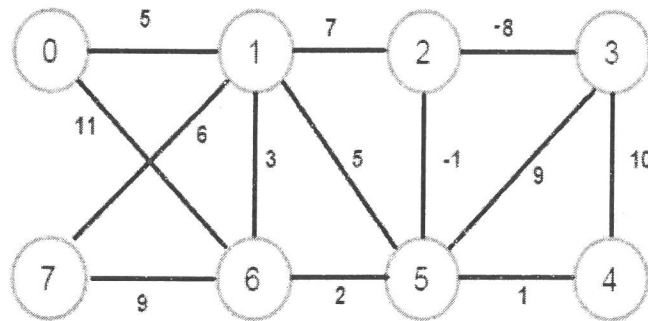
Max. Marks: 30

Instructions to students:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever required and mention it clearly.
3. Writing appropriate units, nomenclature, and drawing neat sketches/schematics/ flow chart wherever, required are an integral part of the answer.

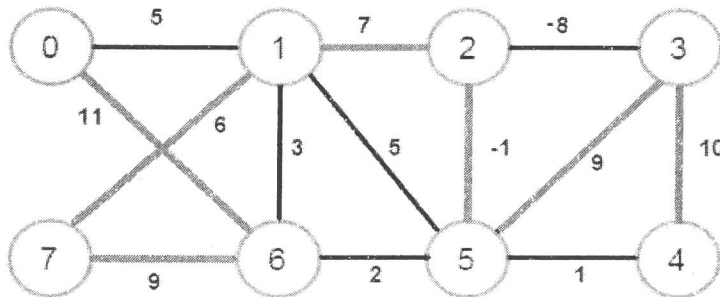
Q.1A Given a weighted undirected graph, Write Pseudo Code/ Algorithm that will find the maximum cost path from a given source to any other vertex in the graph which is greater than a given cost. The path should not contain any cycles. 05

For example, consider the following graph,



Input Let source = 0 and cost = 50.

The maximum cost route from source vertex 0 is 0 → 6 → 7 → 1 → 2 → 5 → 3 → 4, having cost 51, which is more than cost 50. The solution should return 51.



Q.1B	Consider the stack of plates in a marriage party. If the stack gets too high, it might topple. Therefore, in real life, we would likely start a new stack when the previous stack exceeds some threshold count. Implement a data structure SetOfStacks that mimics this. SetOfStacks should be composed of several stacks and should create a new stack once the previous one exceeds capacity. SetOfStacks.push() and SetOfStacks.pop() should behave identically to a single stack (that is, pop() should return the same values as it would if there were just a single stack). Note: Implement a function popAt(int index) which performs a pop operation on a specific sub-stack.	05
Q.2A	There are 25 horses among which we need to find out the fastest 3 horses. In each race, only 5 horses can run simultaneously because there are only 5 tracks. What is the minimum number of races required to find the 3 fastest horses without using a stopwatch? Write your algorithm.	04
Q.2B	Write Pseudo Code/ Algorithm to rearrange the given linked list in a fashion that all odd position nodes precede all even positioned node. E.g.: Input: 1→2→3→4→5→ NULL Output: 1→3→5→2→4→ NULL	04
Q.3A	Write a Pseudo Code/ Algorithm to test whether a given Binary Tree is a Binary Search Tree or not. Sample Input <pre> graph TD 3((3)) --> 5((5)) 3 --> 2((2)) 5 --> 1((1)) 5 --> 4((4)) 2 --> 6((6)) </pre> Sample Output : NO <pre> graph TD 3((3)) --> 2((2)) 3 --> 4((4)) 2 --> 1((1)) 4 --> 5((5)) 4 --> 6((6)) </pre> Sample Output: YES <pre> graph TD 3((3)) --> 2((2)) 3 --> 5((5)) 2 --> 1((1)) 5 --> 6((6)) 5 --> 4((4)) </pre> Sample Output : NO	04

Q.4A	You are having an array of n distinct numbers. You have to write Pseudo Code/Algorithm to sort all even-placed numbers in increasing and odd-placed numbers in decreasing order. The modified array should contain all sorted even-placed numbers followed by reverse sorted odd-placed numbers. Note that the first element is considered as even placed because of its index 0. Examples: Input: arr[] = {0, 1, 2, 3, 4, 5, 6, 7} Output: arr[] = {0, 2, 4, 6, 7, 5, 3, 1} Even-place elements : 0, 2, 4, 6 Odd-place elements : 1, 3, 5, 7 Even-place elements in increasing order : 0, 2, 4, 6 Odd-Place elements in decreasing order : 7, 5, 3, 1 Input: arr[] = {3, 1, 2, 4, 5, 9, 13, 14, 12} Output: {2, 3, 5, 12, 13, 14, 9, 4, 1} Even-place elements : 3, 2, 5, 13, 12 Odd-place elements : 1, 4, 9, 14 Even-place elements in increasing order : 2, 3, 5, 12, 13 Odd-Place elements in decreasing order : 14, 9, 4, 1	04
------	---	----

Q.4B	Draw an AVL tree that results from inserting into an initially empty tree with keys given below – Sequence:- E, A, S, Y, Q, U, E, S, T, I, O, N and then delete Q. Consider the A is smaller than B and so on.	04
OR		
Q.4B	A Binary tree has 12 nodes. The in-order and preorder traversals of the tree yields the following sequence of nodes: Preorder: G, B, Q, A, C, K, F, P, D, E, R, H In order: Q, B, K, C, F, A, G, P, E, D, H, R Draw the Binary tree and find Post order of the tree.	04

