Bharatiya Vidya Bhavan's

# Sardar Patel Institute of Technology

(Autonomous College Affiliated to the University of Mumbai)

**End Semester Examination Synoptic**

May 2024

**Max. Marks: 100** **Duration: 180 min.** **Class: T.E.**

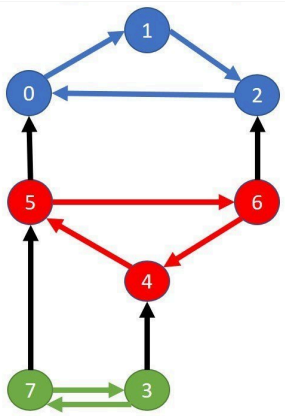
**Semester: VI** **Course Code: OECS3** **Branch: Extc/ETRX**

**Name of the Course: OE Data Structures And Analysis of Algorithms**

**Instructions:**

(1) **All Questions are Compulsory**
(2) **Draw neat diagrams**
(3) **Assume suitable data if necessary**

| Q. No. | Question | Max Mks | CO-BL-PI |
|---|---|---|---|
| *Q1 (a)* | Write a function to perform the following operations on a doubly linked list:<br>i) insertion at a specific location into the list.----**05 marks**<br>ii) delete the given data from the list.----------**05 marks**<br>**OR**<br>Write a function to perform the following operations on the Circular Singly linked list:<br>i) Insert after the given element into the list—---------**05 marks**<br>ii) delete the given element from the list---------**05 marks** | *10* | *1-3-1.4.1* |
| *Q1 (b)* | **[(5*(2+2)-7)*(2+7)] +(10 - [(6+8) *(-1)])**<br>I. Convert the provided infix expression into a postfix expression.<br>**5 2 2 + * 7 _ 2 7 + * 10 6 8 + -1 * - +** **---------05 marks**<br>II. Create a binary tree to represent the above postfix expression. Show the stepwise creation of a binary tree using a stack.---------**05 marks** | *10* | *2-3-4.1.2* |
| *Q2 (a)* | i) Arrange the following numbers: 9, 5, 1, 12, 6, 7, and 8 into a BST. Let the root be at position 1. —----------**02 marks**<br>Represent the BST using an array data structure. **---------05 marks**<br>Array size= A[31]<br>A[1]=9<br>a[2]=5<br>a[3]=12<br>a[4]=1<br>a[5]=6<br>a[11]=7<br>a[23]=8 | *04* | *2-3-4.1.2* |

| | | | |
|---|---|---|---|
| | ii) Given a Binary Search Tree (BST) and a positive number k=2, write a function to find the kth largest node in the BST. Assume the tree is already created.<br>**Note: The logic of In-order Traversal should not be used for the above function logic implementation.**<br><br>Logic of implementation by extracting successive largest and deleting the same and finding largest again in the updated tree<br>Find lagrgest—----------**3 marks**<br>Deletion—----------**3 marks** | ***06*** | |
| ***Q2 (b)*** | Use the DFS algorithm to find the graph's strongly connected components. Display each node's start and end times and represent the same into the balanced parentheses pattern that the DFS algorithm produced.<br>  <br><br>1- draw dfs tree with time—--------05 marks<br>2- showing balanced parenthesis—--02 marks<br>3- Identifying CC correctly–03 marks | ***10*** | |
| ***Q3 (a)-i*** | Given an array A[0---7] of **n** distinct elements, find the K= 4th smallest element using the Quick sort algorithm. Draw a recursive call tree structure for the same.<br>Assume:- Pivot element is extreme right<br><br>\| 7 \| 4 \| 1 \| 9 \| 5 \| 12 \| 10 \| 8 \|<br><br>**Draw a recursive call tree structure for the same. —03 marks**<br>**correctly placing pivot element —----------03 marks** | ***06*** | ***2-3-4.1.2*** |
| ***Q3 (a)-ii*** | Utilize the Binary Search Algorithm on the provided array to look for the following keys: -10, 130.<br>For every iteration, display the Low, Mid, and High index values. | ***04*** | ***2-3-4.1.2*** |

| | | | |
|---|---|---|---|
| | | | |

| -15 | -10 | 45 | 65 | 76 | 110 | 123 | 132 | 140 | 142 |
|---|---|---|---|---|---|---|---|---|---|

**Each Search with correct l m h indices----------02 marks(2*2=4)**

| Search X=-10<br>LOW | MID | HIGH |
|---|---|---|
| 0 | 5 | 9 |
| 0 | 2 | 4 |
| 0 | 1 | 1—----data found |
| Search X=130 | | |
| 0 | 5 | 9 |
| 6 | 7 | 9 |
| 6 | 6 | 6 |
| 7 | | 6—low>high Not found |

| | | | |
|---|---|---|---|
| ***Q3 (b)-i*** | Examine and contrast the strategies of Divide and Conquer, Greedy, and Dynamic Programming with example.<br>**Minimum 3 correct differences —----------5 marks** | ***05*** | ***4-3-4.1.2*** |
| ***Q3 (b)-ii*** | Solve the given recurrences using the Master method theorem. If the recurrence relation is solvable, specify the applicable case; otherwise, justify why it is not solvable.<br>i) $T(n)= 6T(n/3) + n^2 \log n$—**----------2.5 marks**<br>ii) $T(n)=2n\ T(n/2) + n^n$ **----------2.5 marks** | ***05*** | ***3-3-1.4.1*** |
| ***Q4 (a)*** | 1- Insert the given keys into a hash table of size 13, using Double Hashing to resolve any collisions that may occur. —<br>**3 collisions occur each 2 marks---------- 6 marks**<br>Use h1(k) = k mod 13, and h2(k) = 8 - (k mod 8). After successfully allocating all of the keys, display the final hash table. | ***10*** | ***4-3-4.1.2*** |

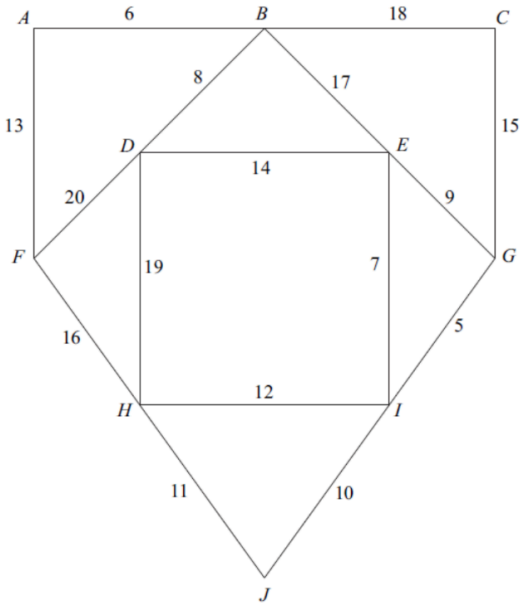
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 18 | 41 | 22 | 44 | 59 | 32 | 31 | 73 |

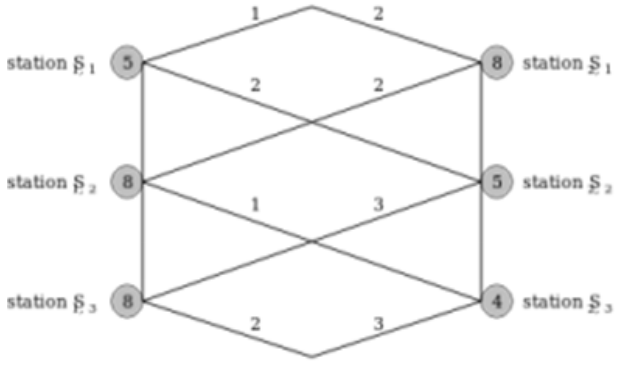
2- Delete the keys listed below. Show the steps for deleting keys and the final updated table after deletion.----------**02 marks**

Delete(31) and Delete(32).

3. Perform a search operation for the following keys on the hash table following a successful deletion. Outline the steps involved in the search and provide an explanation if a search is unsuccessful**.----------2 marks**

Search(73) and Search(31)

| | | | |
|---|---|---|---|
| ***Q4 (b)*** | Apply the Kruskal algorithm to the specified network. Show all the updated tables and draw the final MST with the cost of MST<br>  <br><br>1- **sorted edges—----------02 marks**<br>**2- successively, including edges with updated CC—----------04marks**<br>**3-Final MST with correct cost—----------4 marks** | ***10*** | ***3-3-1.4.1*** |
| ***Q5 (a)*** | Show that the Longest Common Subsequence problem has an optimal substructure.**----------04 marks**<br>Consider the two strings "LONGEST" and "STONE".<br>Create an LCS table using the dynamic programming approach. —**-04marks**<br>Determine the length of the longest common sequence and write it.**----02 marks**<br>**Answer ='ONE"**<br><br>**OR** | ***10*** | |

| | | | |
|---|---|---|---|
| | Write an algorithm for the Knapsack Problem using the Gregreedy approach and state its time complexity. —----------**05 marks**<br><br>Find the best solution to the following knapsack instance, where n=5 and the knapsack capacity = 80. —----------**05 marks**<br><br>Objects 1 2 3 4 5<br><br>Weight 20 30 40 10 7<br><br>Profit 7 8 9 1 6<br><br>**Items in Sack<I5,I1,I3,I4*23/4> profit==26.17** | | |
| ***Q5(b)*** | From Isengard, Iron Man and Captain America set out on a mission to reach Middle Earth. There are two distinct routes from Isengard to Middle-Earth, as Fig. 1 illustrates. Each track has three pay stations where users must stop; however, they can change tracks at any time by paying an additional fee. One can reach the (i+1)th toll of the second track by switching tracks from the ith toll of the first track, and vice versa.<br><br>1-Demonstrate the property of overlapping substructure that the given problem detected.----------**03 marks**<br>**2- shwing updated table—----------05 marks**<br>**3- showing final path—----------02 marks**<br>Utilize the dynamic programming technique to determine the lowest possible cost of traveling from Isengard to Middle-Earth. Show the calculation.<br>Isengard<br>  <br><br>Middle Earth<br><br>Final answer with path L1-L2-L2==20 | ***10*** | ***2-3-1.4.1*** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | F1 | 6 | 14[1] | 24[1] | =17+3<br>=20[2] | | |
| | F2 | 10 | 13[1] | 17[2] | | | |