

**BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI
(END SEMESTER EXAMINATION)**

CLASS: BTECH/IMSC
BRANCH: CSE & AIML/PHYSICS

SEMESTER : III/V/ADD
SESSION : MO/2025

SUBJECT: CS24201 DATA STRUCTURES AND ALGORITHM

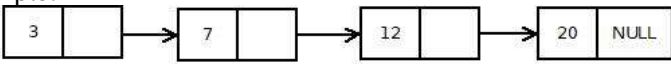
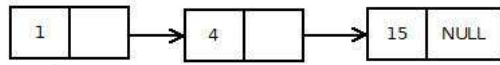
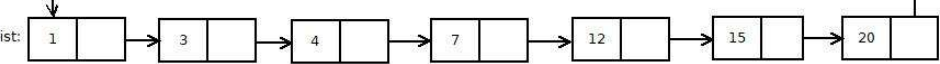
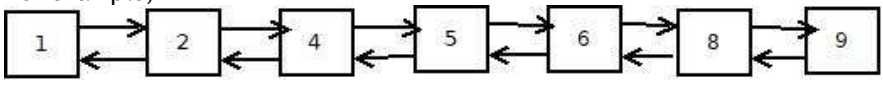
TIME: 3 Hours

FULL MARKS: 50

INSTRUCTIONS:

1. The question paper contains 5 questions each of 10 marks and total 50 marks.
2. Attempt all questions.
3. The missing data, if any, may be assumed suitably.
4. Before attempting the question paper, be sure that you have got the correct question paper.
5. For answering algorithm/pseudocode, writing language specific syntax will be penalized.
6. While writing algorithm/pseudocode, you can only use the data structures covered, i.e., array, linked list, stack, queue, tree and graph. Use of any other language-specific data structure or undefined built-in functions, e.g., length(), maxsize(), sort(), etc., will be penalized.

READ THE ABOVE INSTRUCTIONS CAREFULLY

- | | | CO | BL |
|--|-----|----|-----|
| Q.1(a) Solve the following recurrence by expansion and write the final answer in terms of n:
$T(n) = T(n-1) + 3, T(1) = 2.$
Prove that $3n^2 - 5n + 7 = \Omega(n^2)$ using the definition of notation Ω . | [5] | 1 | 1,2 |
| Q.1(b) Represent the following polynomial using an array $P(x) = 12x^9 - 5x^4 + 7x^2 - 3.$
A polynomial is represented using a singly linked list, where each node stores: coefficient, exponent, and pointer to next term represented by c, e and next. You are given the following partially completed pseudocode for inserting a node in non-increasing order of exponents.
Insert(head, c, e):
<ol style="list-style-type: none"> 1. if head = NULL then 2. A 3. If $e > \text{head} \rightarrow e$ then 4. B 5. C Write the pseudocode completing the missing parts A, B, & C. Consider all possible cases. | [5] | 2 | 1 |
| Q.2(a) You are given two singly linked lists, L1 and L2, each sorted in non-decreasing order. The two lists may be of different sizes. Both are ordinary non-circular singly linked lists. Write pseudocode for a function that merges the two sorted lists into a single circular linked list, also sorted in non-decreasing order.
Example:
L1: 
L2: 
Final List:  | [5] | 2 | 3 |
| Q.2(b) Given a sorted doubly linked list of positive distinct elements, write pseudocode for a function that finds pairs in the list whose sum is equal to a given input value x in sorted order. For example,
Input: 
Target x = 7
Output: 1, 6 and 2, 5 | [5] | 2 | 3 |

PTO

- Q.3(a) Following is an implementation of an unbounded stack S using a queue Q: [5] 3 2
- | | |
|--------------------------------|----------------------------|
| PUSH(S, x) | POP(S) |
| 1. ENQUEUE(Q, x) | 1. If Q is empty |
| 2. For i = 1 to SIZE(Q) - 1 do | 2. print "Stack Underflow" |
| 3. temp = DEQUEUE(Q) | 3. Else |
| 4. ENQUEUE(Q, temp) | 4. return DEQUEUE(Q) |

Let the stack is initially empty and n number of push followed by m number of pop operations ($m < n$) are performed. Find the total number of enqueue and dequeue operations in terms of n and m.

- Q.3(b) A circular queue is implemented using an array $Q[0...n-1]$, along with two indices: front (points to the first element), rear (points to the last element). [5] 3 2,3
- Initially, front = rear = -1 and the queue size is n. Explain why a circular queue is preferred over a linear queue when using an array.

Write pseudocode for the following operations of a circular queue:
 Enqueue(Q, x), Dequeue(Q), IsFull(Q), IsEmpty(Q).

- Q.4(a) The following post-order traversal of a Binary Search Tree is partially incorrect. Identify the incorrect part of the order: 20, 40, 30, 55, 65, 60, 90, 80, 70, 50. [5] 5 4

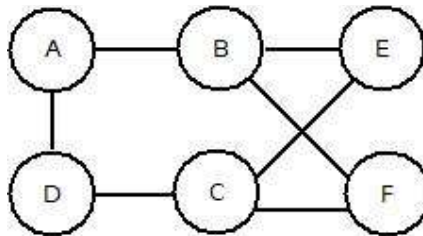
You are given the sequence of integers: {50,30,70,20,40,60,80,35,45,65,75,85}. Insert the above in the given order (left to right) into an initially empty Binary Search Tree (BST). Draw the final BST. Next, you are asked to delete the node with key 70 in a way such that the number of comparisons needed to search for the key 75 in the modified BST is minimized. Describe the deletion strategy you follow and how many comparisons would be needed to find the element 75 in the new BST. No need to write pseudocode.

- Q.4(b) Consider an array A of size n, where all elements are same, e.g., $A=[2,2,2,\dots, 2]$. Two sorting algorithms are applied to this array: insertion sort and quick sort. In quick sort the pivot is always chosen as the last element of the current subarray. Compare the number of comparisons performed by each sorting in terms of n. You do not need to write pseudocode. [5] 4 3,4

You are given the following set of keys to be inserted into a hash table: {4733, 4743, 3587, 5832, 2866, 4186, 4183, 1389}. A hash function is defined as: $h(x)=x \text{ mod } 11$, where 11 is a prime number used as the table size. (a) Construct the Separate Chaining hash table for the above keys; (b) Construct the Open Addressing hash table using Linear Probing for the same set of keys where all keys are stored in the same table even though a collision occurs.

- Q.5(a) [5] 5 3

Consider the given undirected graph. Construct the adjacency matrix of the graph G, using the vertex order [A,B,C,D,E,F].



A binary heap is stored in an array using level-order order indexing, where positions start from 1 (i.e., the root is stored at index 1). For a heap with 13 elements find the indices of the leaf nodes.

For a heap with n elements stored as $H[1...n]$, the structural relationships are:

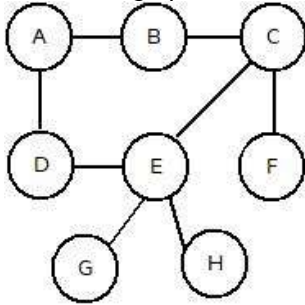
Left child of index i: $2i$; Right child of index i: $2i + 1$; Parent of index i: $\lfloor i/2 \rfloor$

Determine a general formula giving the range of array indices that correspond to all leaf nodes in the heap.

Q.5(b) In an AVL tree, the minimum number of nodes $N(h)$ for a tree of height h satisfies the recurrence relation: $N(h) = N(h-1) + N(h-2) + 1$, where $N(0) = 1$ and $N(1) = 2$. Given that an AVL tree has 50 nodes, determine the maximum possible height of the tree. [5]

5 3

Consider the following connected undirected graph G:



Construct a Spanning Tree of the graph using Depth First Search starting from vertex A. Use adjacency lists in alphabetical order, i.e., explore neighbours in sorted order. From the resulting DFS Spanning Tree, determine the number of leaf nodes in the spanning tree.

.....20/11/2025 E.....