

**DATA STRUCTURES AND ADVANCED ALGORITHMS  
(IOT2101)**

**Time Allotted : 2½ hrs**

**Full Marks : 60**

*Figures out of the right margin indicate full marks.*

*Candidates are required to answer Group A and  
any 4 (four) from Group B to E, taking one from each group.*

*Candidates are required to give answer in their own words as far as practicable.*

**Group – A**

1. Answer any twelve:

**12 × 1 = 12**

*Choose the correct alternative for the following*

- (i) Let A be a two-dimensional array declared as follows:  
A: array [1...10] [1...15] of character;  
Assuming that each character takes one memory locations the array is stored in row-major order and the first element of the array is stored in location 100, what is the address of the element A[i][j] ?  
(a)  $15i + j + 84$  (b)  $15j + i + 84$   
(c)  $10i + j + 89$  (d)  $10j + i + 89$
- (ii) The load factor in hashing is defined as:  
(a) Table size / Number of keys (b) Number of keys / Table size  
(c) Number of collisions / Table size (d) None of these
- (iii) The height of a complete binary tree with n nodes is approximately:  
(a)  $\log_2 n$  (b)  $n/2$   
(c) n (d)  $\sqrt{n}$
- (iv) What is the worst-case time complexity of quick sort?  
(a)  $O(n \log n)$  (b)  $O(n^2)$   
(c)  $O(\log n)$  (d)  $O(n)$
- (v) Red-Black trees guarantee which property?  
(a) All leaves are black (b) Root is always black  
(c) Red nodes have black children (d) All of these
- (vi) Dijkstra algorithm's running time depends on:  
(a) Graph representation (b) Choice of priority queue  
(c) Number of vertices and edges (d) All of the above
- (vii) In union-find, path compression helps:  
(a) Avoid cycles (b) Speed up find operation  
(c) Balance the tree (d) Delete nodes

- (viii) Matrix chain multiplication problem aims to optimize:  
 (a) Number of elements multiplied (b) Order of multiplication  
 (c) Resultant matrix size (d) None
- (ix) The time complexity of Floyd-Warshall algorithm with n number of nodes is:  
 (a)  $O(n)$  (b)  $O(n^2)$   
 (c)  $O(n^3)$  (d)  $O(n \log n)$
- (x) In branch and bound, the node whose bound is the best is:  
 (a) Expanded first (b) Ignored  
 (c) Deleted (d) None

*Fill in the blanks with the correct word*

- (xi) The primary data structure used to implement recursion is a \_\_\_\_\_.
- (xii) The traversal of a binary search tree that produces nodes in sorted order is \_\_\_\_\_ traversal.
- (xiii) Topological sort can be applied only on \_\_\_\_\_ graphs.
- (xiv) The asymptotic time complexity of building a max-heap is \_\_\_\_\_.
- (xv) KMP algorithm uses the \_\_\_\_\_ function to avoid unnecessary comparisons.

### Group - B

2. (a) How do you merge two sorted linked lists into a single sorted linked list? Provide an algorithm and analyze its time complexity. [[CO1](Understand & Apply/IOCQ)]
- (b) State the advantages and disadvantages of linked list over array. [[CO1](Remember/LOCQ)]
- (c) Find the Big-Oh for  $T(n) = 48n^{100} + 2^{n+2} + 3n^2 + 100$ . Justify your answer. [[CO1](Understand/IOCQ)]
- (d) Is Double Ended Queue a FIFO data structure? Explain. [[CO1](Understand/IOCQ)]
- 4 + 3 + 3 + 2 = 12**
3. (a) Convert the infix expression  $A*(B+C/D)-E^F$  to postfix, showing stack contents at each step. [[CO1](Apply/IOCQ)]
- (b) Evaluate the postfix expression  $5 2 3 + * 4 2 ^ -$ ; show stack transitions. [[CO1](Apply/IOCQ)]
- (c) Why is it beneficial to use a circular array to implement a queue? Explain with an example. [[CO1](Analyse/HOCQ)]
- 4 + 4 + 4 = 12**

### Group - C

4. (a) Define binary tree, full binary tree, complete binary tree and binary search tree with examples. [[CO2](Remember/LOCQ)]
- (b) Show each and every step to construct a binary search tree (BST), for which the in-order and post-order traversal sequences are as follows:  
**In-order:** 20 30 50 75 80 100 180 190 195 200 250 295  
**Post-order:** 30 20 75 80 50 195 190 200 180 295 250 100

Delete the data items 100 and 50 (one at a time) from the above constructed BST.

*[[CO2)(Understand & Apply/IOCQ]]*

**4 + (5 + 3) = 12**

5. (a) Show each step of insertions (along with any height balancing steps, if required), to construct an AVL tree using the following keys, inserted one at a time and as per the given order:

45, 80, 70, 40, 30, 65, 35, 25, 27, 15.

Show each step of deletion (along with any height balancing steps, if required) from the above constructed AVL tree, where each of the following keys to be deleted (one at a time): 70, 80.

*[[CO2) (Apply/IOCQ]]*

- (b) For a binary tree with n internal nodes where every internal node has exactly two children, prove the number of leaves is n+1.

*[[CO2) (Apply/IOCQ]]*

**8 + 4 = 12**

**Group - D**

6. (a) Describe how to count connected components in an undirected graph using Breadth First Search (BFS) or Depth First Search (DFS) algorithm. Analyze the complexity of this algorithm.

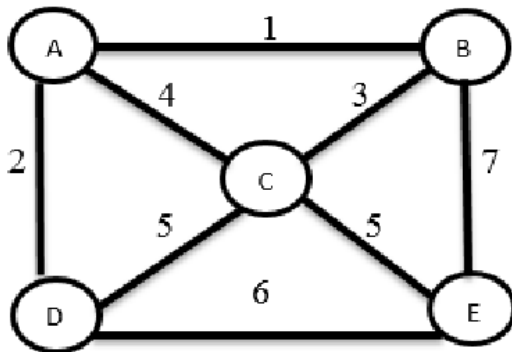
*[[CO4)(Understand & Apply/HOCQ]]*

- (b) Define adjacency matrix and adjacency list. Compare the time and space complexity of these two graph representations.

*[[CO4)(Understand/IOCQ]]*

**(4 + 2) + (3 + 3) = 12**

7. (a) Apply Prim’s algorithm to find Minimum Spanning Tree (MST) of the given graph. Show all steps.



*[[CO4)(Apply/IOCQ]]*

- (b) Explain the Union-Find (Disjoint Set Union) data structure. Illustrate union by rank and path compression with an example.

*[[CO4)(Understand/LOCQ]]*

**6 + (3 + 3) = 12**

**Group - E**

8. (a) Describe the differences between 0/1 Knapsack and Fractional Knapsack problems.

*[[CO5)(Analyse/IOCQ]]*

- (b) Explain the greedy vs Dynamic Programming approaches for the Knapsack problem.

*[[CO5) (Analyse/IOCQ]]*

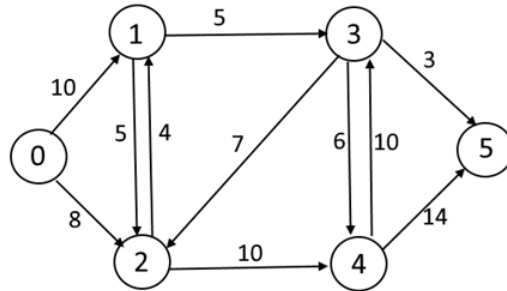
- (c) Solve the 0/1 Knapsack problem using Dynamic Programming for the given items and knapsack capacity=11. The weight and profit of each of the items are as follows:

$P_i$ (Rs)	50	30	32	27
$W_i$ (Kg)	5	6	4	3

[[CO5] (Apply/HOCQ)]

**3 + 3 + 6 = 12**

9. (a) Define the concept of residual graph and augmenting path in the context of network flow. [[CO6](Understand/IOCQ)]
- (b) Apply Ford–Fulkerson algorithm on a given flow network and compute the maximum flow where source = 0 and sink = 5.



[[CO5,CO6] (Apply/IOCQ)]

[[CO6] (Understand/LOCQ)]

- (c) Mention real-world applications of max-flow algorithms.

**3 + 6 + 3 = 12**

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	16.67	66.67	16.67