(II) 10/10/2025

NEP - Semester End Examination – October 2025

Program: SY.B.Sc.IT SEM-III **Course:** Data Structure for Problem Solving

Program Code: UGIT01 **Course Code:** NUIT302

Duration: 1 Hour **Max. Marks:** 30

Instructions:

1. All questions are compulsory.
2. Figures to the right indicate full marks.
3. Draw neat diagrams wherever necessary.

Q. 1		**Attempt any THREE of the following. (5 Marks each)**	**[15]**	**Course Outcome**	**Knowledge Level**
	(a)	Define a stack. List its main operations and explain each operation with one example.		**CO1**	**L1**
	(b)	Explain the difference between best-case, worst-case, and average-case time complexity.		**CO2**	**L2**
	(c)	Write the algorithm to insert an element into an array at a given position. Demonstrate with an example input.		**CO1,** **CO3**	**L3**
	(d)	Compare arrays and linked lists. Analyze which operations (insertion, deletion, traversal) are more efficient in each case and justify your answer.		**CO2**	**L4**
	(f)	Evaluate the behavior of a linear queue of size 3 for the following sequence of operations: Enqueue 5, Enqueue 15, Display, Dequeue, Enqueue 25, Display, Dequeue, Dequeue, Display, Dequeue. Criticize any overflow or underflow conditions that occur. Also design and implement a simple algorithm or C/C++ program to demonstrate the Enqueue(x) operation.		**CO3**	**L5**
	(g)	Design and construct a singly linked list that allows insertion at the end of the list. Formulate a small example to demonstrate the working of the operation. You may either provide a C/C++ program or a step-by-step algorithm.		**CO3**	**L6**
Q. 2		**Attempt any THREE of the following. (5 Marks each)**	**[15]**	**Course Outcome**	**Knowledge Level**
	(a)	Write a C/C++ program or algorithm to perform Insertion Sort on an array of integers. Apply Insertion Sort algorithm to sort the array 64, 34, 25, 12 and show the array after each pass.		**CO3**	**L3**
	(b)	What is hashing in data structures? Explain collision in a hash table and describe in simple terms how separate chaining resolves it.		**CO1**	**L2**

	(c)	Implement and demonstrate one searching technique using an algorithm or C/C++ code. Use an example array to show how the search works.		CO3	L3
	(d)	Design and implement a Binary Search Tree (BST) in C/C++ for the sequence 50, 30, 20, 80, 60, 40, 70. Generate and present its Inorder, Preorder, and Postorder traversals.		CO4	L6
	(e)	Design and construct a Huffman coding scheme for the given character frequencies A(5), B(9), C(12), D(13), and E(16). Formulate the Huffman codes for each character and develop the total number of compressed bits required.		CO2	L6
	(g)	Evaluate the use of the modulo division method for inserting the values 99, 33, 23, 44, 56, 43, and 19 into a hash table of size 7. Discuss the problems caused by collisions and explain how linear probing helps resolve them. Present the final arrangement of the hash table.		CO2	L5

--X--X--