Advanced Data Structures and Algorithms

E2UC503C

ASSIGNMENT 1

Note: In case of programming problem, write your code in Java or Python Only.

1. Explain the concept of single-dimensional and multi-dimensional arrays.
2. Describe how arrays are represented in the row major order.
3. Explain how arrays are represented in the column major order.
4. Define what a sparse matrix is and describe its representation.
5. Explain how insertion and deletion operations work in a singly linked list. Write its code also.
6. Derive the index formulae for 1-D, 2-D, and 3-D arrays.
7. Evaluate the time complexity of the insertion operation in a singly linked list.
8. Given an array of size N with integers in the range of [1, N], the task is to find the missing number from the first N integers. There are no duplicates in the list. Write its program also.
9. Write a program that, given an array of n elements that contains elements from 0 to n-1, with any of these numbers appearing any number of times. Find these repeating numbers in O(n).

Advanced Data Structures and Algorithms

E2UC503C

ASSIGNMENT 2

Note: In case of programming problem, write your code in Java or Python Only.

1. Describe the process of converting an infix expression to a postfix expression. Write its program.
2. Define tail recursion and explain its significance.
3. Write a program to implement stack data structure using a linked list.
4. Write a program to implement queue data structure using a linked list.
5. Describe the steps involved in evaluating a postfix expression.
6. Explain the principles of recursion and how it differs from iteration.
7. How is a queue different from a stack, and in what scenarios is each useful?
8. Analyse the advantages and disadvantages of using recursion in problem-solving compared to iterative approaches.
9. Implement a recursive function to solve the Tower of Hanoi problem.
10. Create a program that demonstrates the use of a stack to check the balance of parentheses in an expression.
11. Write a program to implement stack using queue and vice versa.

Advanced Data Structures and Algorithms

E2UC503C

ASSIGNMENT 3

Note: In case of programming problem, write your code in Java or Python Only.

1. Explain the index sequential search and its applications.
2. Describe the basic concept of hashing in data structures.
3. Explain common collision resolution techniques used in hashing.
4. Compare and contrast sequential search and binary search algorithms.
5. Write code to insert, delete, and search for a node in a binary search tree.
6. Create a program that constructs a binary tree from given in-order and pre-order traversals.
7. Implement a basic AVL tree and demonstrate insertion, deletion, and searching operations.
8. Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, find the number of collisions.
9. Evaluate the efficiency of different types of search trees, such as AVL trees, B+ trees, Quad trees, and Oct trees, for specific applications.
10. Write a code to implement Quad tree, and Octree.