



IT12 Data Structure and Algorithms

Hands-On Questions

Q.1 Singly Linked List

Write a program to perform the following operations on a Singly Linked List:

1. Append a node at the end of the list. (Easy)
2. Print the list. (Easy)
3. Find the Minimum and Maximum node value. (Easy)
4. Count Even and odd value nodes. (Easy)
5. Insert a node at a specific position. (Medium)
6. Delete the last node. (Medium)
7. Delete the middle node (Medium)
8. Search for an element in the list. (Medium)
9. Merge two lists. (Medium)
10. Reverse the list. (Hard)
11. Swap node values of consecutive nodes (Pairwise Swapping) (Hard)

Q2. Doubly Linked List

Write a program to perform the following operations on a Doubly Linked List:

1. Append a node at the end of the list. (Easy)
2. Print the list. (Easy)
3. Count total nodes. (Easy)
4. Print node values in reverse order. (Medium)
5. Insert a node at any position of the list. (Medium)
6. Delete a node at a specific position. (Medium)
7. Search for an element in the list, print its position (Medium)
8. Delete alternate nodes. (Hard)
9. Reverse the list. (Hard)



Q3. Stack

Write a program to perform the following operations on a Stack:

1. Push an element onto the stack. (Easy)
2. Pop an element from the stack. (Easy)
3. Peek the top element of the stack. (Easy)
4. Reverse the stack using an array. (Medium)
5. Reverse the stack using another stack. (Medium)
6. Reverse a string using stack. (Medium)
7. Check for balanced parenthesis. (Hard)

Q4. Linear Queue

Write a program to perform the following operations on a Linear Queue:

1. Enqueue an element into the queue. (Easy)
2. Dequeue an element from the queue. (Easy)
3. Peek the front element of the queue. (Easy)
4. Reverse the queue using stack. (Medium)
5. Count no occurrences of an element in a queue. (Medium)
6. Search for an element in the queue. (Medium)
7. Merge two queues into one. (Hard)
8. Check if two queues are identical (Hard)



Q5. Priority Queue

Write a program to perform the following operations on a Priority Queue:

1. Enqueue an element with a given priority. (Easy)
2. Display Queue. (Easy)
3. Dequeue the element with the highest priority. (Medium)

Q6. Binary Search Tree

Write a program to perform the following operations on a Binary Search Tree:

1. Insert a node into the BST. (Easy)
2. Find the minimum value in the BST. (Easy)
3. Find the maximum value in the BST. (Easy)
4. Search for a specific node and print its level. (Medium)
5. Count the number of nodes in the BST. (Medium)
6. Count the number of nodes in the left subtree. (Medium)
7. Count the number of nodes in the right subtree. (Medium)
8. Delete the specific node. (Hard)

Q7. Algorithms

1. Write a program to search element using Binary search
2. Write a program to generate Pascal Triangle
3. Write a program of Tower of Hanoi for n=3
4. Write a program to find GCD using Euclidean Algorithm
5. Write a program to sort elements in the list using Merge Sort
6. Write a program to sort elements in the list using Quick Sort
7. Write a program to implement Jump Game
8. Write a program to find unique paths for m*n matrix