### List of Experiments - ADSA



|  |  |  |
| --- | --- | --- |
| LAB EXAM | | |
| 1 | L | **Find the Maximum and Minimum Elements in an Array:** Write a function to find the maximum and minimum elements in an array. |
| 2 | L | **Reverse an Array:** Write a function to reverse an array in place. |
| 3 | L | **Find the Kth Smallest/Largest Element in an Array:** Write a function to find the Kth  smallest or largest element in an array. |
| 4 | L | **Sort an Array of 0s, 1s, and 2s:** Given an array containing only 0s, 1s, and 2s, sort the array in linear time. |
| 5 | L | **Move All Zeroes to End of Array:** Write a function to move all zeroes in an array to the end while maintaining the relative order of other elements. |
| 6 | L | **Reverse a Linked List:** Write a function to reverse a singly linked list. |
| 7 | L | **Detect a Cycle in a Linked List:** Write a function to detect if a cycle exists in a linked list. |
| 8 | L | **Find the Middle of a Linked List:** Write a function to find the middle element of a linked list. |
| 9 | L | **Merge Two Sorted Linked Lists:** Write a function to merge two sorted linked lists into one  sorted linked list. |
| 10 | L | **Remove Nth Node from End of List:** Write a function to remove the Nth node from the start/end of a linked list. |
| 11 | L | **Implement a Stack Using Arrays/Lists:** Write a function to implement a stack using an array or list with basic operations: push, pop, peek, and isEmpty. |
| 12 | L | **Implement a Stack Using Linked List:** Write a function to implement a stack using alinked list with basic operations: push, pop, peek, and isEmpty. |
| 13 | L | **Check for Balanced Parentheses:** Write a function to check if a string containing parentheses is balanced. |
| 14 | L | **Evaluate Postfix Expression:** Write a function to evaluate a given postfix expression. |
| 15 | L | **Next Greater Element:** Write a function to find the next greater element for each elementin an array. |
| 16 | L | **Implement a Queue Using Arrays/Lists:** Write a function to implement a queue using an array or list with basic operations: enqueue, dequeue, front, and isEmpty. |
| 17 | L | **Implement a Queue Using Linked List:** Write a function to implement a queue using a linked list with basic operations: enqueue, dequeue, front, and isEmpty. |
| 18 | L | **Implement a Circular Queue:** Write a function to implement a circular queue with basic operations: enqueue, dequeue, front, rear, and isEmpty. |
| 19 | L | **Generate Binary Numbers from 1 to N:** Write a function to generate binary numbers from1 to N using a queue. |
| 20 | L | **Implement a Queue Using Stacks:** Write a function to implement a queue using two stacks. (vice-versa) |
| 21 | L | **Implement a Binary Tree:** Write a class to implement a basic binary tree with insert,delete, and traversal operations. |
| 22 | L | **Inorder Traversal:** Write a function to perform inorder traversal of a binary tree. |
| 23 | L | **Preorder Traversal:** Write a function to perform preorder traversal of a binary tree. |
| 24 | L | **Postorder Traversal:** Write a function to perform postorder traversal of a binary tree. |
| 25 | L | **Level Order Traversal:** Write a function to perform level order traversal of a binary tree. |
| 26 | L | **Height of a Binary Tree:** Write a function to find the height of a binary tree. |
| 27 | L | **Diameter of a Binary Tree:** Write a function to find the diameter of a binary tree. |
| 28 | L | **Check if a Binary Tree is Balanced:** Write a function to check if a binary tree is height balanced. |
| 29 | L | **Lowest Common Ancestor:** Write a function to find the lowest common ancestor of two nodes in a binary tree. |
| 30 | L | **Implement Graph Using Adjacency List:** Write a class to implement a basic graph usingan adjacency list with methods to add vertices and edges. |
| 31 | L | **Breadth-First Search (BFS):** Write a function to perform BFS on a graph from a givenstart vertex. |
| 32 | L | **Depth-First Search (DFS):** Write a function to perform DFS on a graph from a given start vertex. |
| 33 | L | **Detect Cycle in an Undirected Graph:** Write a function to detect if there is a cycle in an undirected graph. |
| 34 | L | **Connected Components in an Undirected Graph:** Write a function to find the number of  connected components in an undirected graph. |
| 35 | L | **Find MST Using Kruskal’s Algorithm:** Write a function to find the Minimum Spanning Tree of a graph using Kruskal’s algorithm. |
| 36 | L | **Find MST Using Prim’s Algorithm:** Write a function to find the Minimum Spanning Treeof a graph using Prim’s algorithm. |
| 37 | L | **Fibonacci Sequence:** Write a function to compute the nth Fibonacci number using dynamic programming. |
| 38 | L | **Climbing Stairs:** Write a function to determine how many distinct ways there are to climb a staircase with n steps if you can climb either 1 or 2 steps at a time. |
| 39 | L | **Min Cost Climbing Stairs:** Write a function to determine the minimum cost to reach thetop of a staircase given a list of costs associated with each step. |
| 40 | L | **House Robber:** Write a function to determine the maximum amount of money you can rob from a row of houses without robbing two adjacent houses. |
| 41 | L | **Maximum Subarray Sum (Kadane’s Algorithm):** Write a function to find the contiguous subarray with the maximum sum. |
| 42 | L | **Activity Selection:** Given a set of activities with start and end times, select the maximum number of activities that do not overlap. |
| 43 | L | **Fractional Knapsack Problem:** Given weights and values of items and the maximum capacity of a knapsack, determine the maximum value that can be obtained by including fractions of items. |
| 44 | L | **Huffman Coding:** Given a set of characters and their frequencies, construct the HuffmanTree to encode the characters. |
| 45 | L | **Job Sequencing Problem:** Given a set of jobs, each with a deadline and profit, maximize  the total profit by scheduling the jobs to be done before their deadlines. |
| 46 | L | **Minimum Number of Coins:** Given different denominations of coins and an amount, findthe minimum number of coins needed to make up that amount. |
| 47 | L | **N-Queens Problem:** Place N queens on an N×N chessboard so that no two queens threaten each other. |
| 48 | L | **Permutations:** Generate all possible permutations of a given list of numbers or characters. |
| 49 | L | **Subsets:** Generate all possible subsets of a given set of numbers. |