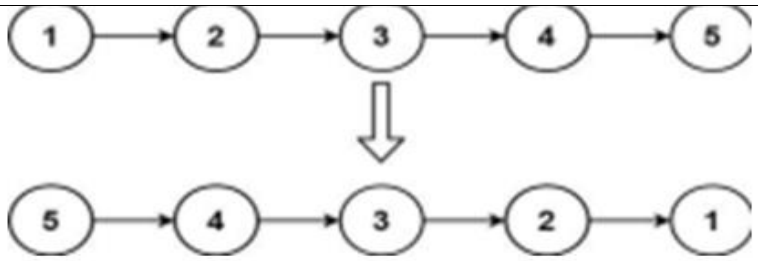
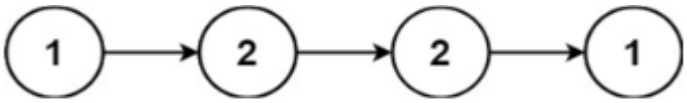
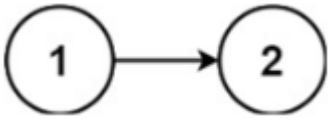
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY (CHARUSAT)
DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY AND RESEARCH
DEPSTAR
Department of Computer Science and Engineering

Subject Name	:	Data Structures & Algorithms	**Class**	:	B.Tech. (III Sem)
Subject Code	:	CSE203	**Academic year**	:	2024-2025

Practical List

Sr. No.	Aim of the Practical	Hours	COs
1.	Implement Linear Search and Binary Search using an array data structure. **Supplementary Experiment:** **Implement Ternary Search Algorithm**. It is a search algorithm that is used to find the position of a target value within a sorted array. It operates on the principle of dividing the array into three parts instead of two, as in binary search.	2	CO1
2.	2.1 You are designing a simple text editor that uses an appropriate data structure to manage undo operations. Every time a user makes a change to the text, the state of the text before the change is inserted onto a data strucure. If the user decides to undo a change, the editor will delete the top state from the data structure and revert the text to that state. Additionally, the user should be able to see the list of recent changes that can be undone. 2.2 Implement a program to convert infix notation to postfix notation using appropriate data structure.	4	CO1
3.	Imagine you are a wizard tasked with moving a set of magical rings from one tower to another. You have three towers: • Tower A (the source) • Tower B (the auxiliary) • Tower C (the destination) There are n rings of different sizes on Tower A. Your goal is to move all the rings to Tower C using the following rules: 1. You can only move one ring at a time. 2. You can only move the top ring of any tower.	2	CO1

	3. No ring may be placed on top of a smaller ring. To help you in this magical task, you will use a recursive algorithm. Implement it.		
4.	4.1 Implement basic operations (enqueue (), dequeue () and display ()) of queue using an array. 4.2 Implement basic operations (enqueue (), dequeue (), and display ()) of circular queue using array.	4	CO1
5.	5.1 Implement the following operations of a singly linked list. (a) Insert a node at front (b) Insert a node at end (c) Insert a node after given node information (d) Delete a node at front (e) Delete a node at last 5.2 Implement following operations of doubly linked list. (a) Insert a node at front (b) Insert a node at end (c) Insert a node after given node information (d) Delete a node at front (e) Count number of nodes 5.3 Implement following operations of circular singly linked list. (a) Inserting a node at front (b) Delete a node at end **Supplementary Experiment:** (a) Reverse Linked List Given a singly linked list, reverse the list, and return the reversed list. Input: head = [1,2,3,4,5] Output: [5,4,3,2,1]	8	CO2

	 (b) Palindrome Linked List Given a singly linked list, return true if it is a palindrome or false otherwise. (without using extra data structure) Example 1: Input: head = [1,2,2,1]  Output: true Example 2: Input: head = [1,2]  Output: false		
6.	Write a program to concatenate two doubly linked lists.	2	CO2
7.	7.1 Implement basic operations (push (), pop () and display ()) of stack using linked list. 7.2 Implement basic operations (enqueue (), dequeue () and display ()) of queue using linked list.	4	CO3
8.	Implement Binary Search Tree (BST) using following operations. (a) Insert (b) Search (c) Traversal (Inorder, Preorder, Postorder)	4	CO3/CO4

9.	Implement a Graph to perform following operations. 9.1 Adjacency list representation 9.2 Apply DFS and BFS on the given graph.	2	CO5
10.	A library sort its collection of books. Each book has a unique ID number, and the library wants to organize these books in ascending order of their IDs. 10.1 The library has a small collection of rare books. Since the collection is small, efficiency is not the primary concern. **Steps to be followed:** 1. Start with the first book and assume it has the smallest ID. 2. Compare this book with the rest of the books to find the one with the smallest ID. 3. Swap the book with the smallest ID with the first book. 4. Move to the next book and repeat the process until all books are sorted. The library receives new books regularly, and you need to integrate them into an already sorted collection quickly. **Steps to be followed:** 1. Start with the second book, assuming the first book is sorted. 2. Compare the second book with the first and insert it in the correct position. 3. Move to the next book and repeat the process, inserting each book into the correct position in the sorted part of the array. 4. Continue until all books are sorted.	4	CO5/CO6
11.	11.1 The library has a small collection of books, and they want to organize these books in ascending order of their IDs. **Steps to be followed:** 1. Start with the first pair of books and compare their IDs. 2. If the first book's ID is greater than the second book's ID, swap them.	4	CO5/CO6

	3. Move to the next pair of books and repeat the process until you reach the end of the list. 4. Repeat steps 1-3 for the entire list until no swaps are needed in a complete pass. 11.2 The library has just received a large donation of books from a benefactor. The collection consists of thousands of books, each with a unique ID number. Given the size of the collection, the library needs an efficient algorithm to sort these books quickly and ensure they are organized properly on the shelves. **Steps to be followed:** 1. Choose a 'pivot' book from the list. 2. Partition the other books into two sub-arrays: books with IDs less than the pivot and books with IDs greater than the pivot. 3. Recursively apply the same process to the sub-arrays. 4. Combine the sub-arrays and pivot to get the sorted list.		
12.	The library has a massive collection of books that is too large to handle efficiently with simple sorting algorithms. Additionally, the library frequently merges collections from different donors, each already sorted. To handle these large collections and merging tasks, the library should use appropriate sorting technique. Suggest a sorting technique while ensuring efficient and stable nature of sorting technique.	2	CO5/CO6
13.	In an array of 20 elements, arrange 15 different values, which are generated randomly between 1,00,000 to 9,99,999. Use hash function to generate key and linear probing to avoid collision. $H(x) = (x \bmod 18) + 2$. Write a program to input and display the final values of array.	2	CO1
Hacker rank problem			
1	First Problem **1D-Array DS** [https://www.hackerrank.com/domains/data-structures]	2	CO1
2	First Problem of **2D-Array DS** [https://www.hackerrank.com/domains/data-structures]	2	CO2/CO4

3.	Insert a node at the tail of the Linked List [https://www.hackerrank.com/domains/data-structures]	4	CO4/CO5
4	Cycle Detection [https://www.hackerrank.com/domains/data-structures]	2	CO5
5	Merge two sorted linked list [https://www.hackerrank.com/domains/data-structures]	2	CO6
6	In a faraway Galaxy of Tilky Way, there was a planet Tarth where the sport of Competitive Coding was very popular. According to legends, there lived a setter known for loving knapsack type problems. Given N objects in a row, with weights W1,W2,…,WN, you need to find the maximum number of consecutive objects you can fill in a bag of maximum capacity C such that the total weight of objects taken is at least K. In other words, pick objects such that-The total weight of collected objects is at least K. The total weight does not exceed C. The objects picked must be consecutive (i.e. a subarray of the objects need to be picked) The number of objects is maximized. You need to print this maximum value. Note-If no such object could be picked, then the answer is obviously 0. Input • The first line of input contains T, number of test cases in a file. • The next line contains three integers, N, C and K, as described in the problem statement. • The next line contains N space separated integers, denoting Wi, i.e. weight of the object. Output • For test case, output the maximum number of objects you can pick. Input 2 5 5 5 5 4 3 2 1 5 5 4 1 4 1 1 1 Output 2 2	4	CO2/CO4