

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY AND RESEARCH

Subject: Data Structure and Algorithms

Semester: 3

Subject Code: CE261

Academic Year :2023-24(ODD)

Course Outcome (COs):

At the end of the course, the students will be able to:

- CO1 Understand and Implement Algorithms and core Data Structures such as stack, queue, hash table, priority queue, binary search tree and graph in programming language.
- CO2 Analyze data structures in storage, retrieval and computation of ordered or unordered data.
- CO3 Compare alternative implementations of data structures with respect to demand and performance.
- CO4 Describe and evaluate the properties, operations, applications, strengths and weaknesses of different data structures.
- CO5 Apply and select the most suitable data structures to solve programming challenges.
- CO6 Discover advantages and disadvantages of specific algorithms.

Practical List

Sr. No.	AIM	Hrs.	CO
1.	Implement Linear Search and Binary Search using array data structure. Supplementary Experiment: [L: A] 1. https://www.codechef.com/problems/SEGM01 2. https://www.hackerrank.com/contests/launchpad-1-winter-challenge/challenges/binary-search-advanced 3. https://www.codechef.com/problems/CHEFHCK2 4. https://www.codechef.com/problems/SNAKEEAT 5. https://www.codechef.com/problems/DIVSET	2	1
2.	For a given array B1, B2, ..., BM of length at least 3, let's define its weight as the largest value of (Bi-Bj)·(Bj-Bk) over all possible triples (i, j, k) with 1 ≤ i, j, k ≤ M and i ≠ j, j ≠ k, k ≠ i. You are given a sorted array A1, A2, ..., AN (that is, A1 ≤ A2 ≤ ... ≤ AN).	2	2

	<p>Calculate the sum of weights of all contiguous subarrays of A of length at least 3. That is, count the sum of weights of arrays $[A_i, A_{i+1}, \dots, A_j]$ over all $1 \leq i < j \leq N$ with $j-i \geq 2$.</p> <p>Input:</p> <p>The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows. The first line of each test case contains an integer N. The second line of each test case contains N space-separated integers A_1, A_2, \dots, A_N.</p> <p>Output:</p> <p>For each test case, print a single line containing the sum of weights of all subarrays of A of length at least 3.</p> <p>Constraints</p> <ul style="list-style-type: none"> $1 \leq T \leq 1000$ $3 \leq N \leq 3000$ $1 \leq A_1 \leq A_2 \leq \dots \leq A_N \leq 10^6$. Sum of N over all test cases won't exceed 6000 <p>Sample Input 1</p> <pre>2 4 1 2 3 4 5 1 42 69 228 2021</pre> <p>Sample Output 1</p> <pre>4 1041808</pre>		
3.	<p>3.1 Implement following operations of singly linked list.</p> <ol style="list-style-type: none"> Insert a node at front Insert a node at end Insert a node after given node information Delete a node at front Delete a node at last <p>3.2 Implement following operations of doubly linked list.</p> <ol style="list-style-type: none"> Insert a node at front Insert a node at end Insert a node after given node information Delete a node at front Count number of nodes <p>3.3 Implement following operations of circular singly linked list.</p>	4	1,2

	(a) Inserting a node at front (b) Delete a node at end Note: Display content of linked list after each operation.		
4.	Implement Sorting Algorithm(s). (a) Bubble Sort (b) Selection Sort (c) Quick Sort (d) Merge Sort Supplementary Experiment: <ol style="list-style-type: none"> https://www.codechef.com/problems/TSORT [L: M] https://www.codechef.com/problems/MRGSRT [L: A] 	4	3
5.	<p>Chef and his little brother are playing with sticks. They have total N sticks. Length of i-th stick is A_i. Chef asks his brother to choose any four sticks and to make a rectangle with those sticks its sides. Chef warns his brother to not to break any of the sticks, he has to use sticks as a whole. Also, he wants that the rectangle formed should have the maximum possible area among all the rectangles that Chef's brother can make. Chef's little brother takes this challenge up and overcomes it. Can you also do so? That is, you have to tell whether it is even possible to create a rectangle? If yes, then you have to tell the maximum possible area of rectangle.</p> <p>Input</p> <ul style="list-style-type: none"> The first line contains a single integer T denoting the number of test-cases. T test cases follow. The first line of each test case contains a single integer N denoting the number of sticks. The second line of each test case contains N space-separated integers A_1, A_2, \dots, A_N denoting the lengths of sticks. <p>Output</p> <ul style="list-style-type: none"> For each test case, output a single line containing an integer representing the maximum possible area for rectangle or -1 if it's impossible to form any rectangle using the available sticks. <p>Input</p> <pre>2 5 1 2 3 1 2 4 1 2 2 3</pre> <p>Output</p> <pre>2</pre>	2	5

	-1		
6.	1. Implement basic operations (push (), pop () and display ()) of stack using array. 2. Implement basic operations (push (), pop () and display ()) of stack using linked list. Supplementary Experiment: 1. https://www.codechef.com/problems/SUDBOOKS [L: A]	2	1,2
7.	<p>Chef has a string which contains only the characters '{', '}', '[', ']', '(' and ')'. Now Chef wants to know if the given string is balanced or not. If is balanced then print 1, otherwise print 0.</p> <p>A balanced parenthesis string is defined as follows:</p> <ul style="list-style-type: none"> The empty string is balanced If P is balanced then (P), {P}, [P] is also balanced if P and Q are balanced PQ is also balanced <p>For example "()", "({})[()]" are balanced parenthesis strings while "([{}])", "())" are not balanced.</p> <p>Input The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows. The first and only line of each test case contains a single string</p> <p>Output For each test case, print a single line containing the answer.</p> <p>Input: 4 () (D] ({})(){} [{}]</p> <p>Output: 1 0 1 0</p>	2	5,6
8.	1. Implement basic operations (enqueue (), dequeue () and display ()) of queue using array. 2. Implement basic operations (enqueue (), dequeue () and display ()) of queue using linked list. 3. Implement basic operations (enqueue (), dequeue () and display ()) of circular queue using array. Supplementary Experiment: 1. https://www.codechef.com/problems/CHFQUEUE [L: A]	2	1,5

9.	<p>There are N people, numbered from 11 to N. They go to a cinema hall. Each of them buys a ticket, which has a number written on it. The number on the ticket of the i^{th} person is A_i.</p> <p>There are infinite seats in the cinema hall. The seats are numbered sequentially starting from 11. All the N people stand in a queue to get their respective seats. Person 11 stands at the front of the queue, Person 22 stands in the second position of the queue, so on up to Person N who stands at the rear of the queue. They were given seats in this manner:</p> <p>Let the number on the ticket of the person currently standing in front of the queue be X. If the X^{th} seat is empty, the person gets out of the queue and takes the X^{th} seat. Otherwise the person goes to the rear of the queue, and the number on his ticket is incremented by one - that is, it becomes $X+1$.</p> <p>Print the seat number occupied by each of the N people.</p> <p>Input</p> <ul style="list-style-type: none"> The first line of input contains a single integer T denoting the number of test cases. The description of T test cases follows. The first line of each test case contains a single integer N. The second line of each testcase line contains N space-separated integers A_1, A_2, \dots, A_N. <p>Output</p> <ul style="list-style-type: none"> For each test case, print a single line containing N space-separated integers, where the i^{th} integer denotes the seat number finally occupied by the Person i. <p>Input</p> <pre>4 5 1 2 3 2 4 4 4 1 3 2 3 1 1 1 5 2 5 1 5 2</pre> <p>Output</p> <pre>1 2 3 5 4 4 1 3 2 1 2 3 2 5 1 6 3</pre>	2	4
10.	Implement Binary Search Tree (BST) using following operations.	4	1,4

	(a) Insert (b) Search (c) Traversal (Inorder, Preorder, Postorder)		
11.	Implement a Graph to perform following operations. 1. Adjacency list representation 2. Apply DFS and BFS on the given graph.	2	2,6
12.	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. $H(x) = (x \bmod 18) + 2$. 2. Write a program to input and display the final values of array.	2	1,3

Initialisms:

L: Level

M: Medium

A: Advanced