

INDIAN INSTITUTE OF TECHNOLOGY JODHPUR



॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Data Structures and Algorithmic Techniques - CSL7561

Assignment - 3

ASSIGNMENT DEADLINE - 30th January, 11:59 PM

NOTE:

- You are allowed to use C programming language/ C++ to solve the problems. No other programming language is allowed.
- Use of API/ Libraries is strictly forbidden, you all need to write the code from scratch (except for input/ output header files).
- You need to submit the assignment before the deadline else your assignment will not be evaluated. No extension requests will be entertained.
- A strict plagiarism check using tools will be done on the code. If plagiarism is found in your code, you will be directly awarded zero for that question.
- The assignment needs to be submitted individually.
- You need to submit your programs (.c/ .cpp files) in a ZIP file.
- **The code needs to be well commented describing what your algorithm does.**

Common Instructions for all Sorting Questions

- Use integer values as elements of your array to perform sorting.
- Make sure that the size of your input is at least 10 elements.

Section - A (2 * 15 = 30 marks)

Implement the following sorting algorithms from scratch:

- 1) Heap Sort Algorithm
- 2) Counting Sort Algorithm

Input Format:

The first line contains the size of the input array (Integer)

The second line contains the input array (All Integers)

Output Format:

Program output should display the input array after sorting the input array using the specified algorithm.

Sample Input 1:

```
10
78 14 33 37 10 6 14 11 2 1
```

Sample Output 1:

```
1 2 6 10 11 14 14 33 37 78
```

Sample Input 2:

```
12
100 160 -23 448 129 -1 10 17 22 99 -5 0
```

Sample Output 2:

```
-23 -5 -1 0 10 17 22 99 100 129 160 448
```

Section - B (2 * 30 = 60 marks)

Implement the following graph algorithms from scratch. Use adjacency matrix to represent your graph:

- 1) Breadth-First Search Algorithm on an undirected graph
- 2) Depth First Search Algorithm on an undirected graph

Input Format:

The first line contains the number of nodes (positive integer)

The second line onwards contains the adjacency matrix (All positive integers)

The last line contains the starting node (positive integer)

Output Format:

Program output should display the DFS/ BFS sequence of nodes from starting node.

Sample Input 1:

```
5
0 1 1 0 0
1 0 0 0 1
1 0 0 1 0
0 0 1 0 1
0 1 0 1 0
0
```

Sample Output 1:

```
0 2 1 3 4 (or) 0 1 2 4 3 //for BFS question
0 2 3 4 1 (or) 0 1 4 3 2 //for DFS question
```

Section - C (1 * 10 = 10 marks)

- 1) Alice has a standard deck of 52 cards. She wants to play a card game with $K-1$ of her friends. This particular game requires each person to have an equal number of cards, so Alice needs to discard a certain number of cards from her deck so that she can equally distribute the remaining cards amongst her and her friends. Find the **minimum** number of cards Alice has to discard such that she can play the game.

Input Format:

The first line of input contains a single integer T , denoting the number of test cases. The description of T test cases follows. Each test case consists of a single line of input, containing a single integer K – the number of people playing the game (including Alice).

Output Format:

For each test case, output in a single line the minimum number of cards Alice has to discard from her deck such that she can play the game with her friends

Sample Input:

```
4
3
50
2
14
```

Sample Output:

```
1
2
0
10
```

Explanation (for testcase 1

Alice can discard any single card which will leave her with 51 cards. She can then give her friends and herself 17 cards each.