Mr.Cooper Practice Questions

Date : 01-05-2025
Submitted by : Srinithi S - IT CIT

Technical Practice day 1:
1.Given a number of ropes denoted as 'N' and an array containing the lengths of these ropes, your task is to connect the ropes into one single rope. The cost to connect two ropes is determined by the sum of their lengths. Your goal is to find the minimum cost required to connect all the ropes.

INPUT:
The first line of input consists of a single integer T, indicating the number of test cases to handle.

Each test case includes:
- The first line specifies the number of ropes, N.
- The second line provides the rope lengths as N space-separated integers.

OUTPUT:
For each test case, output a single integer representing the minimum cost to connect all the ropes. Print each result on a new line.

SAMPLE TEST CASES:
1.
Input:
T = 1
N = 4
Lengths = [4, 3, 2, 6]

Output:
29

Constraints:
1 <= T <= 10
1 <= N <= 10^4
1 <= length[i] <= 10^4

CODE

problem1.java

01-05-2025 > problem1.java > problem1 > getSequenceSum(int[])

```java
import java.util.PriorityQueue;

public class problem1 {
    public static void main(String[] args) {
        // int t = 1 ;
        int[] arr = {4,3,2,6} ;
        System.out.println(getSequenceSum(arr));

    }
    public static int getSequenceSum(int[] n){
        int res =  0 ;
        PriorityQueue<Integer> prq = new PriorityQueue<>();
        for (int i = 0; i < n.length; i++) {
            prq.add(n[i]);
        }
        while (prq.size() > 1) {
            int a = prq.poll();
            int b = prq.poll();
            res += a + b;
            prq.add(a + b);
        }
        return res ;
    }
}
// Time complexity: O(nlogn) where n is the length of the array
// Space complexity: O(n) for the priority queue
```

Output

```
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025> javac problem1.java
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025> java problem1
29
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025>
```

Technical Practice Day 1:
2. Determine if a given N * N matrix is an idempotent matrix. A matrix is considered idempotent if it satisfies the following condition:
M * M = M

INPUT:
The first line contains a single integer N, denoting the size of the matrix. The next N lines each contain N space-separated integers, representing the elements of the matrix.

OUTPUT:
Output 'true' if the matrix is idempotent; otherwise, output 'false'.

SAMPLE TEST CASES:
1.
INPUT:
2
1 0
0 1

OUTPUT:
true

Constraints:
1 <= N <= 10^3

CODE

problem2.java problem2.class

01-05-2025 > problem2.java > problem2 > main(String[])

```java
public class problem2 {
    public static void main(String[] args){
        int[][] arr = {
            {1, 0, 0},
            {0, 1, 0},
            {0, 0, 1}
        };
        System.out.println(checkIdempotent(arr));

        int[][] arr2 = {
            {1, 2},
            {3, 4}
        };
        System.out.println(checkIdempotent(arr2));
    }

    public static boolean checkIdempotent(int[][] arr){
        int n = arr.length;
        int[][] temp = new int[n][n];
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                temp[i][j] = 0;
                for(int k = 0; k < n; k++){
                    temp[i][j] += arr[i][k] * arr[k][j];
                }
            }
        }
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                if(temp[i][j] != arr[i][j]){
                    return false;
                }
            }
        }
        return true ;
    }
}
```

OUTPUT:

```java
    public static void main(String[] args){
        int[][] arr = {
            {1, 0, 0},
            {0, 1, 0},
            {0, 0, 1}
        };
        System.out.println(checkIdempotent(arr));

        int[][] arr2 = {
            {1, 2},
            {3, 4}
        };
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS powershell - 01-05-2025

```
-a----        01-05-2025     10:14           1013 problem3.java


(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025> javac problem1.java
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025> java problem1
29
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025> javac problem2.java
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025> java problem2
true
false
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025>
```

Technical Practice Day 1:
3. Given an integer array arr of size 'N' containing only 0s, 1s, and 2s, write an algorithm to sort the array.

INPUT:
The first line contains an integer 'T' representing the number of test cases. Each test case consists of two lines. The first line contains an integer 'N', the size of the array. The second line contains 'N' space-separated integers, the elements of the array.

OUTPUT:
For each test case, output the sorted array as space-separated integers in a new line.

SAMPLE TEST CASES:

INPUT:
2
5
0 2 1 2 0
3
2 1 0

OUTPUT:
0 0 1 2 2
0 1 2

Constraints:
$1 \leq T \leq 10$
$1 \leq N \leq 5 * 10^5$
$0 \leq arr[i] \leq 2$

CODE

```java
// Problem 3: Sort an array of 0s, 1s, and 2s
public class problem3 {
    public static void main(String[] args) {
        int[] arr = {0, 1, 2, 0, 1, 2, 0, 1, 2};
        System.out.println(x:"Original array: ");
        for (int num : arr) {
            System.out.print(num + " ");
        }
        System.out.println();
        sortArray(arr);
        System.out.println(x:"Sorted array: ");
        for (int num : arr) {
            System.out.print(num + " ");
        }
    }

    public static void sortArray(int[] arr) {
        int low = 0, mid = 0, high = arr.length - 1;

        while (mid <= high) {
            if (arr[mid] == 0) {
                swap(arr, low++, mid++);
            } else if (arr[mid] == 1) {
                mid++;
            } else {
                swap(arr, mid, high--);
            }
        }
    }

    private static void swap(int[] arr, int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}
```

Java: Ready

OUTPUT

```
false
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025> javac problem3.java
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025> java problem3
Original array:
0 1 2 0 1 2 0 1 2
Sorted array:
0 0 0 1 1 1 2 2 2
(base) PS D:\SDE TRAINING\SDE_Training-main\MAY\01-05-2025>
```