Notes

Date __/__/____

Problem Name : Repeat and Missing

N = 2
Arr[] = {2, 2}

// sort array
// create an ans array of size (2)
// ans[0] = 0, ans[1] = 0
// for (i = 1 to n)

```
if (arr[i] == arr[i-1])
    ans[0] = arr[i];
    arr[i] = 0;
}
```

for duplicate or repeat

```
if (arr[i] - 1 != i)
    {
        ans[1] = i + 1;
    }
return ans;
```

missing

emarks

(: O(NlogN))
SC : O(1) // www.lawrencemayo.ac.in

Problem Name : Inversion of Arr

// Brute force

// condition for inversion is

arr[i] > arr[j] and i < j

// for (i = 0 to n-1)

for (j = i+1 to j < n)

if (arr[i] > arr[j] and i < j)

inversion cnt ++;

return inversion cnt;

}

TC : O(N^2)

SC : O(1)

Remarks

// Better approach

// using merge sort

// TC : O(logN)

SC : O(1)

Notes Date__/__/_____

Day-3

Problem Name: Search a 2D matrix

1	3	5	7
10	11	16	20
23	30	34	60

matrix = [[1,3,5,7], [10,11,16,20], [23,30,34,60]]
target = 3

// First logic → Brute force { TC: O(n*m), SC: O(1) }
// using for loop one for → row and for → colm
// if mat == target → true
// else return false.

// Second logic / Approach { TC: O(NlogM), SC: O(1) }
// Using mid
// if (mat[mid/m][mid%m] == target)

Remarks

```
return true;
if (mat[mid/m][mid%m] < target)
    low = mid+1;
```