Data Structures and Algo in Java

Learnt some easy Array Questions to solve Leet code Problems.

1. Find the missing number in an Array
2. Maximum Consecutive Ones
3. Find the number which appear ones from twice appearing numbers.

import java.util.HashSet;

public class arrleet

{

public static void main(String[] args)

{

//int arr [] = {9,9,4,4,3,3,7,0,0};

int arr [] = {1,1,0,0,1,1,1,0,1};

int n = arr.length;

System.out.println(repeating1s(arr,n));

}

public static int missing(int arr[] , int n)

{

HashSet<Integer> set = new HashSet<>();

for(int num:arr)

{

set.add(num);

}

for(int i=0;i<=n;i++)

{

if(!set.contains(i))

{

return i;

}

}

return -1;

}

public static int missingxor(int arr [] , int n)

{

int xor1 = 0;

int xor2=0;

for(int i=0;i<=n;i++)

{

xor1=xor1^i;

}

for(int i=0;i<n;i++)

{

xor2=xor2^arr[i];

}

int miss=xor1^xor2;

return miss;

}

public static int missXorOpt(int arr[], int n)

{

int xor1=0;

int xor2=0;

for(int i=0;i<n;i++)

{

xor1=xor1^(i+1);

xor2=xor2^arr[i];

}

int miss = xor1^xor2;

return miss;

}

public static int singleOpt(int arr[], int n)

{

int xor = 0;

for(int i=0;i<n;i++){

xor=xor^arr[i];

}

return xor;

}

public static int repeating1s(int arr [], int n)

{

int count = 0;

int maxi = 0;

for(int i=0;i<n;i++)

{

if(arr[i]==1)

{

count++;

maxi=Math.max(count,maxi);

}

else{

count =0;

}

}

return maxi;

}

}