- **EASY**
- **Max Score: 30 Points**
-
-
-
-

Missing Numbers

Given two arrays of integers, find which elements in the second array are missing from the first array. If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number. Return the missing numbers sorted in ascending order. Only include a missing number once, even if it is missing multiple times.

Note

If there is no element that is to be printed, print -1.

Input Format

There will be four lines of input:

n - the size of the first array arr,

The next line contains n space-separated integers arr[i]

m - the size of the second array brr,

The next line contains m space-separated integers brr[i]

Output Format

Output array of integers (in increasing order) which tells you elements of second array not present in the first array

Example 1

Input

```
10
203 204 205 206 207 208 203 204 205 206
13
203 204 204 205 206 207 205 208 203 206 205 206 204
```

Output

```
204 205 206
```

Explanation

203 is in both arr and brr and its frequency is 2 in both arrays, hence it is not included in output

204 is included in both arrays but have different frequency hence, it is included in output

Similarly other numbers are considered

Example 2

Input

```
4
1 1 2 5
4
1 2 3 4
```

Output

```
1 3 4
```

Explanation

1 is in both arrays but frequency of 1 in both arrays is different, hence included in output

2 is in both arrays and have same frequency, hence not included in output

3 is in second array but not first, hence included in output

4 is in second array but not first, hence included in output

5 is in not in second array, hence not included in output

Constraints:

1 <= n, m <= 2 * 10^5

1 <= arr[i], brr[i] <= 10^4

My code

// in java
import java.util.*;

//test case wrong

```java
class Solution {
    static void missingNumbers(int n, int arr[], int m, int brr[]) {
        //Write your code here
        HashMap<Integer,Integer>hm=new HashMap<>();
        HashMap<Integer,Integer>hm2=new HashMap<>();
        for(int i=0;i<arr.length;i++)
            {
                hm.put(arr[i],hm.getOrDefault(arr[i],0)+1);
            }
        for(int i=0;i<brr.length;i++)
            {
                hm2.put(brr[i],hm2.getOrDefault(brr[i],0)+1);
            }
        for(int i:hm2.keySet())
            {
                if(hm.containsKey(i))
                {
                    if(hm.get(i)!=hm2.get(i))
                        System.out.print(i+" ");
                }
                else   System.out.print(i+" ");
            }
    }

}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();
        int arr[] = new int[n];
```

```
        for(int i=0; i<n; i++){
            arr[i]= sc.nextInt();
        }

        int m = sc.nextInt();
        int brr[] = new int[m];
        for(int i=0; i<m; i++){
            brr[i]= sc.nextInt();
        }

        Solution Obj = new Solution();
        Obj.missingNumbers(n,arr,m,brr);
    }
}
```