

Topics

ASK/VIEW DOUBT

SOLUTION

HINT

Your Code

Main Code

Code Pair

C++ (g++ 5)

Lecture 14 : Hashmaps

Deadline

Sep 21, 2020, 11:59 PM

Lecture

Score 200.00/200

100.0%

Introduction to Hashmaps

Inbuilt Hashmap

Remove Duplicates

Code : Maximum Frequency Number 40.0/40

Code : Print Intersection 80.0/80

Code : Pair Sum to 0 80.0/80

Iterators

Problem

Result

Code : Print Intersection

Send Feedback

You have been given two integer arrays/list (ARR1 and ARR2) of size N and M, respectively. You need to print their intersection; An intersection for this problem can be defined when both the arrays/lists contain a particular value or to put it in other words, when there is a common value that exists in both the arrays/lists.

Note :

Input arrays/lists can contain duplicate elements.

The intersection elements printed would be in the order they appear in the first sorted array/list (ARR1).

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

The first line of each test case or query contains integer 'N' representing the size of the first array/list.

The second line contains 'N' single space separated integers representing the elements of the first array/list.

The third line contains an integer 'M' representing the size of the second array/list.

The fourth line contains 'M' single space separated integers representing the elements of the second array/list.

Output format :

For each test case, print the intersection elements in a row, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

1 <= t <= 10^2

0 <= N <= 10^6

0 <= M <= 10^6

Time Limit: 1 sec

Sample Input 1 :

2
6
2 6 8 5 4 3
4
2 3 4 7
2
10 10
1
10

Sample Output 1 :

2 3 4
10

Sample Input 2 :

1
4
2 6 1 2
5
1 2 3 4 2

Sample Output 2 :

1 2 2

```
1 // input1 - first array
2 // input2 - second array
3 // size1 - size of first array
4 // size2 - size of second array
5 #include<unordered_map>
6 void intersection(int input1[], int input2[], int size1, int size2) {
7     /* Don't write main().
8      * Don't read input, it is passed as function argument.
9      * Print the output and don't return it.
10     * Taking input is handled automatically.
11     */
12     unordered_map<int,int> seen;
13
14     for(int i=0;i<size1;i++){
15         if(seen.count(input1[i]))
16             seen[input1[i]]++;
17         else
18             seen[input1[i]]=1;
19     }
20     for(int i=0;i<size2;i++){
21         if(seen.count(input2[i]) && seen[input2[i]]!=0){
22             cout<<input2[i]<<endl;
23
24             /*cout<<endl;
25             cout<<seen[2]<<endl;*/
26             seen[input2[i]]--;
27         }
28     }
29 }
30
```



Bucket Array and hash function


Explanation for Sample Output 2 :

Since, both input arrays have two '2's, the intersection of the arrays also have two '2's. The first '2' of first array matches with the first '2' of the second array. Similarly, the second '2' of the first array matches with the second '2' if the second array.



< PREVIOUS

> NEXT

 CUSTOM INPUT

 SUBMIT SOLUTION

