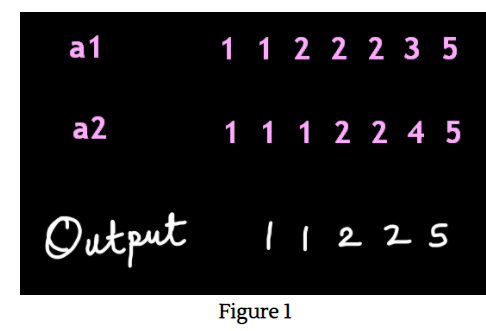
**1. PROBLEM DISCUSSION**

● You are given 2 arrays a1 and a2 of sizes n1 and n2 respectively.

● You are required to find the intersection of a1 and a2 as seen in figure 1.

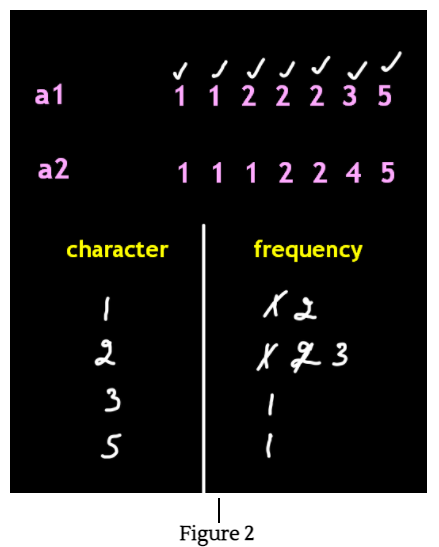
● Don't assume the arrays to be sorted.

● The elements of intersection should be printed in order of their occurence in a2.

****

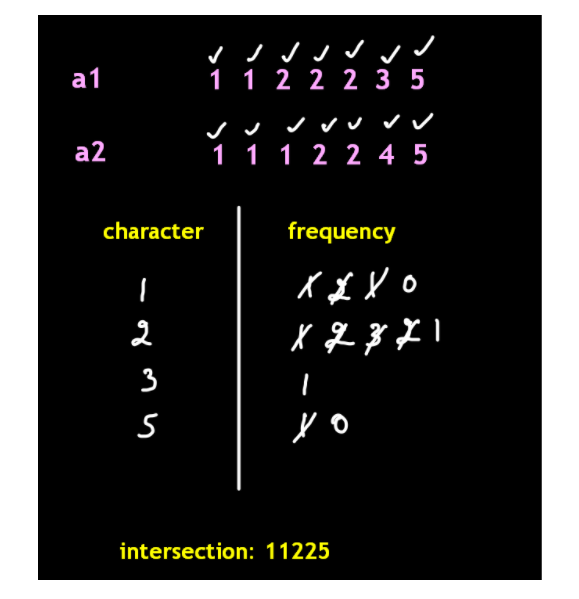
**2. Approach :**

● We make a Hashmap table for every unique element of a1 vs. its frequency.

****

● In figure 2, we iterate through every element of the array a1 and subsequently increase the frequency against it.

● Now, we iterate through each element of the array a2. If that element is present in our Hashmap, then it is printed and the record of that element is removed from the hashmap as shown in figure 3. If the element is not present then we do nothing. This is shown in figure 3.

****

**3. Code**

If you have done the previous question, then this problem's code will be a breeze for you.

ConsoleJava

import java.io.\*;

import java.util.\*;

public class Main {

public static void main(String[] args) throws Exception {

Scanner scn = new Scanner(System.in);

int n1 = scn.nextInt();

int[]a1 = new int[n1];

for (int i = 0; i < n1; i++)

{

a1[i] = scn.nextInt();

}

int n2 = scn.nextInt();

int[]a2 = new int[n2];

for (int i = 0; i < n2; i++)

{

a2[i] = scn.nextInt();

}

HashMap< Integer, Integer> hm = new HashMap();

for (int val : a1)

{

if (hm.containsKey(val)) {

int old = hm.get(val);

int now = old + 1;

hm.put(val, now);

}

else {

hm.put(val, 1);

}

}

for (int val : a2) {

if (hm.containsKey(val)) {

System.out.println(val);

int old = hm.get(val);

if (old > 1)

hm.put(val, old - 1);

else

hm.remove(val);

}

}

}

}

For more clarity of the question, watch the question video

Play Video

**4. TIME & SPACE COMPLEXITY**

TIME COMPLEXITY- O(n)

SPACE COMPLEXITY- O(n)

where n= number of entities in the Hashmap.