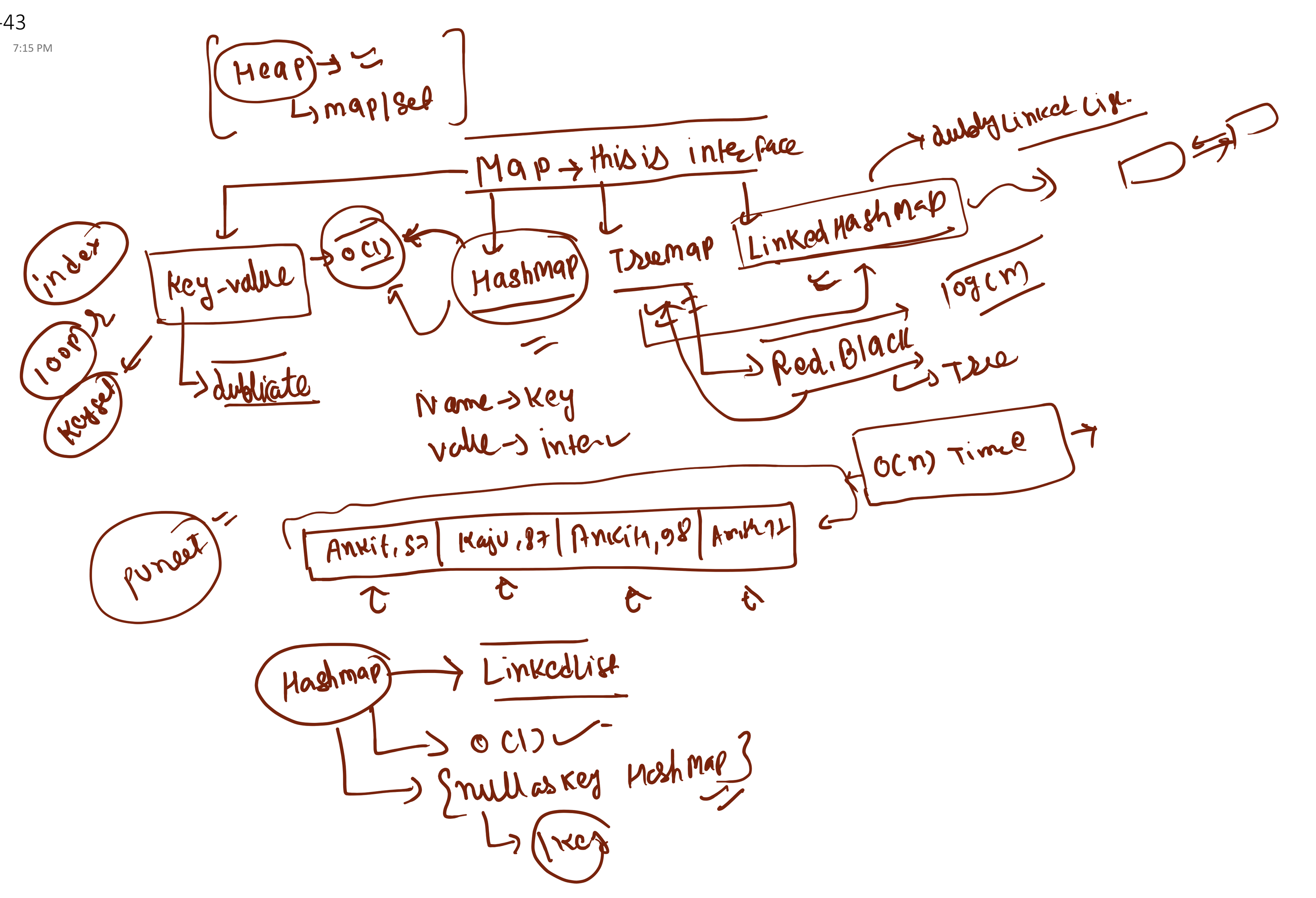
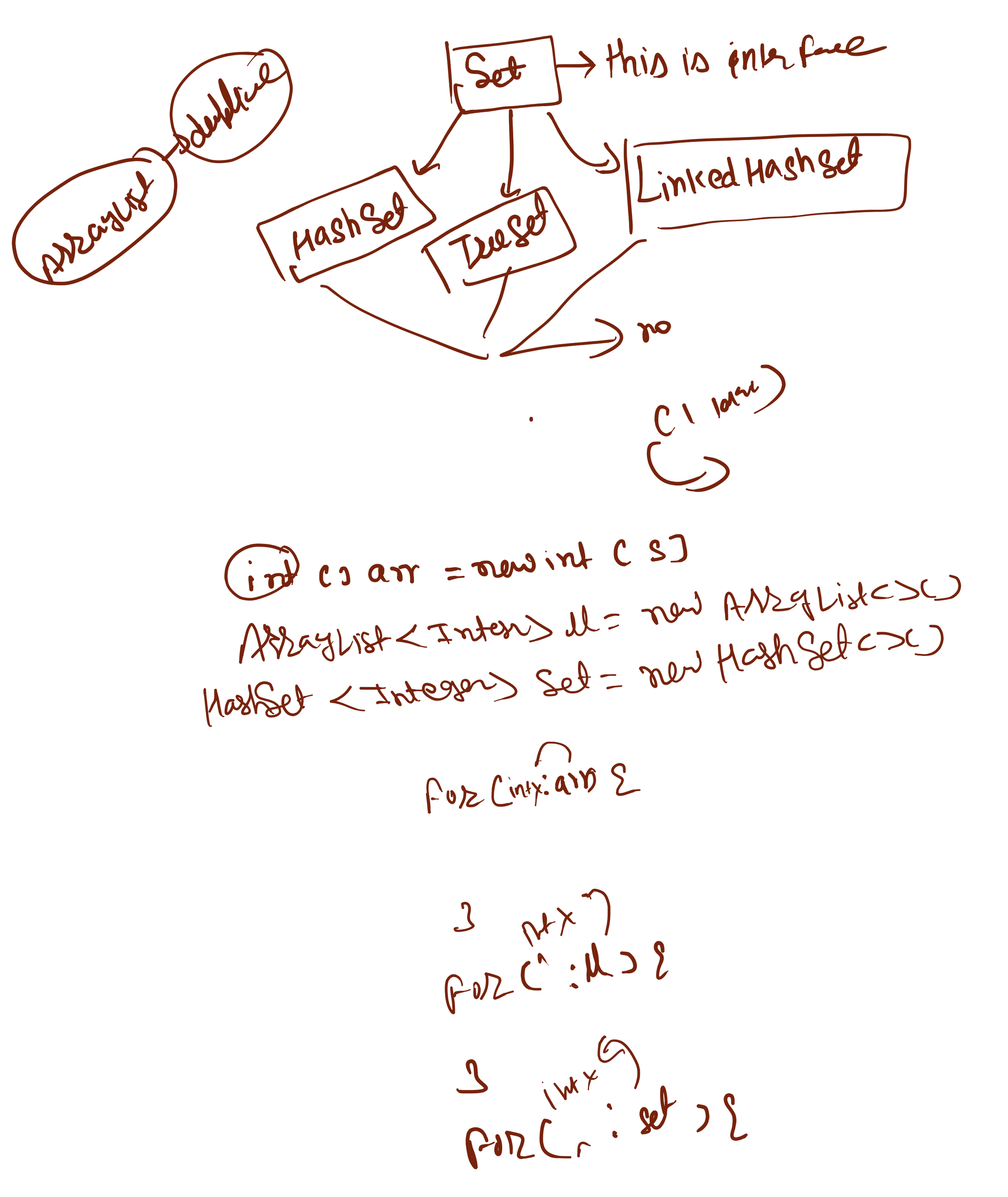
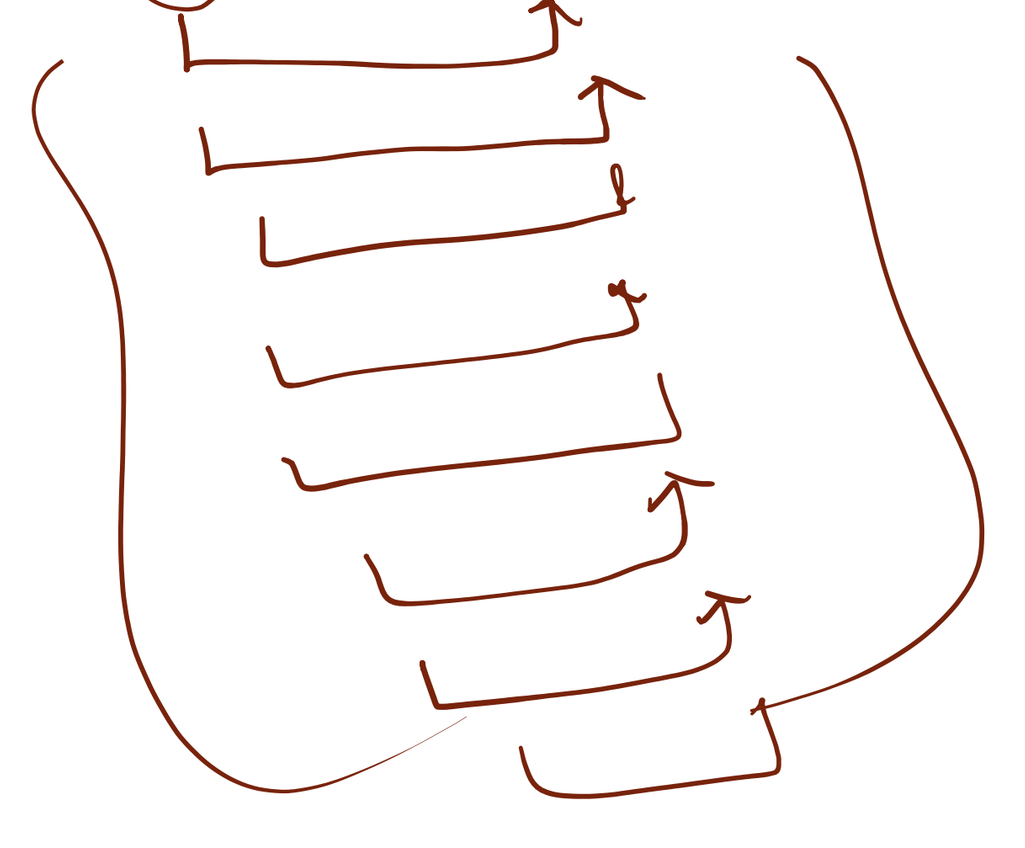
Lec-43





```
System.out.println(map2);
        // System.out.println(map.keySet());
        Set<String> key = map.keySet();
        for (String k : key) {
            System.out.println(k + " " + map.get(k));
        }
```

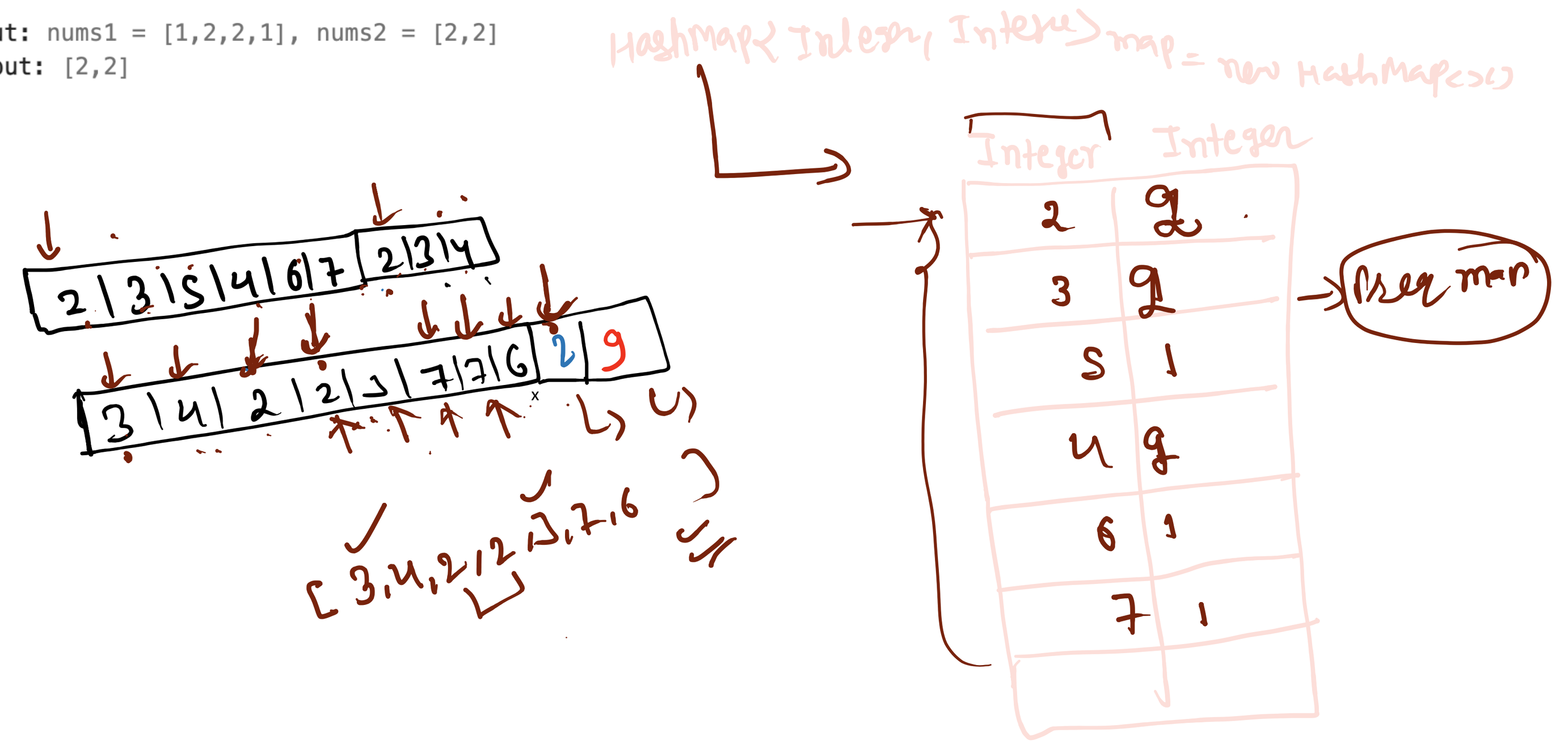
[Kaju, Ankita, Amisha, Kunal, Raj, Puneet, Jaspal, Ankit]



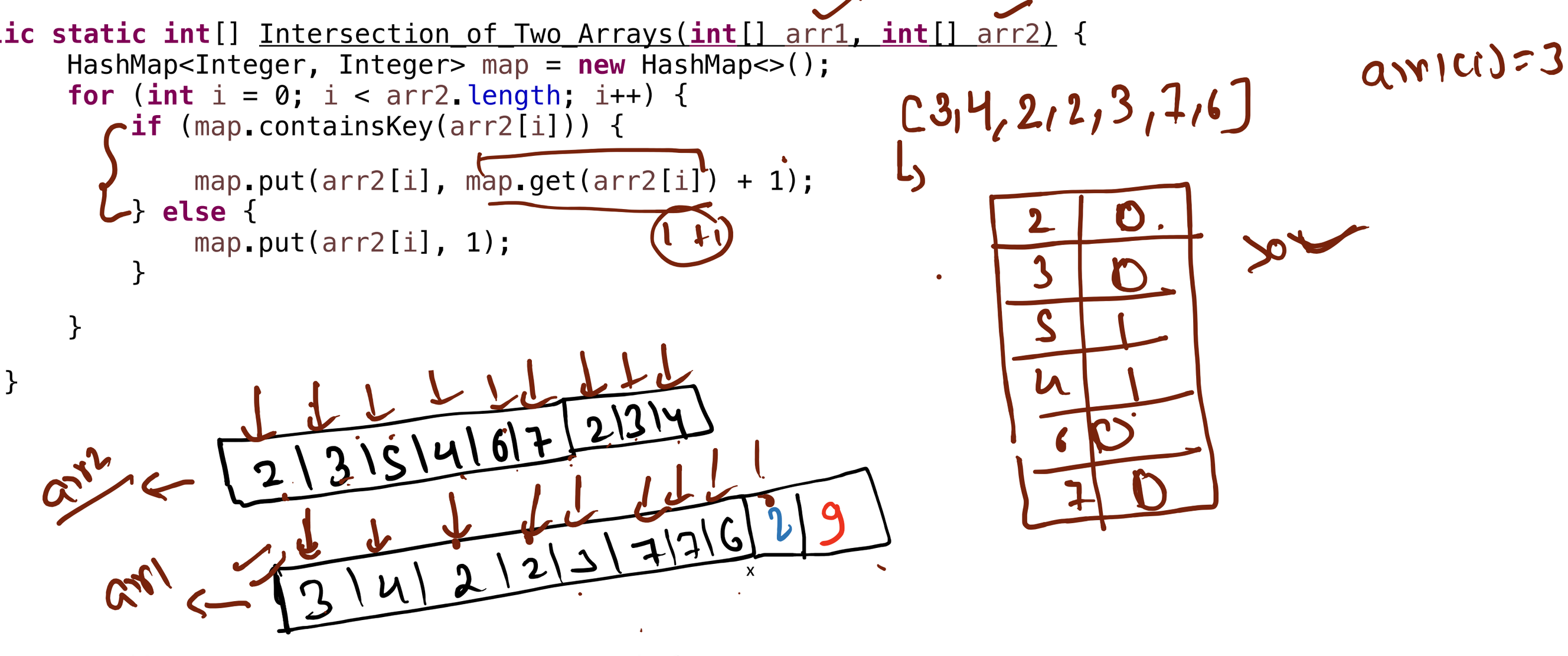
Given two integer arrays `nums1` and `nums2`, return *an array of their intersection*. Each element in the result must appear as many times as it shows in both arrays and you may return the result in **any order**.

Example 1:

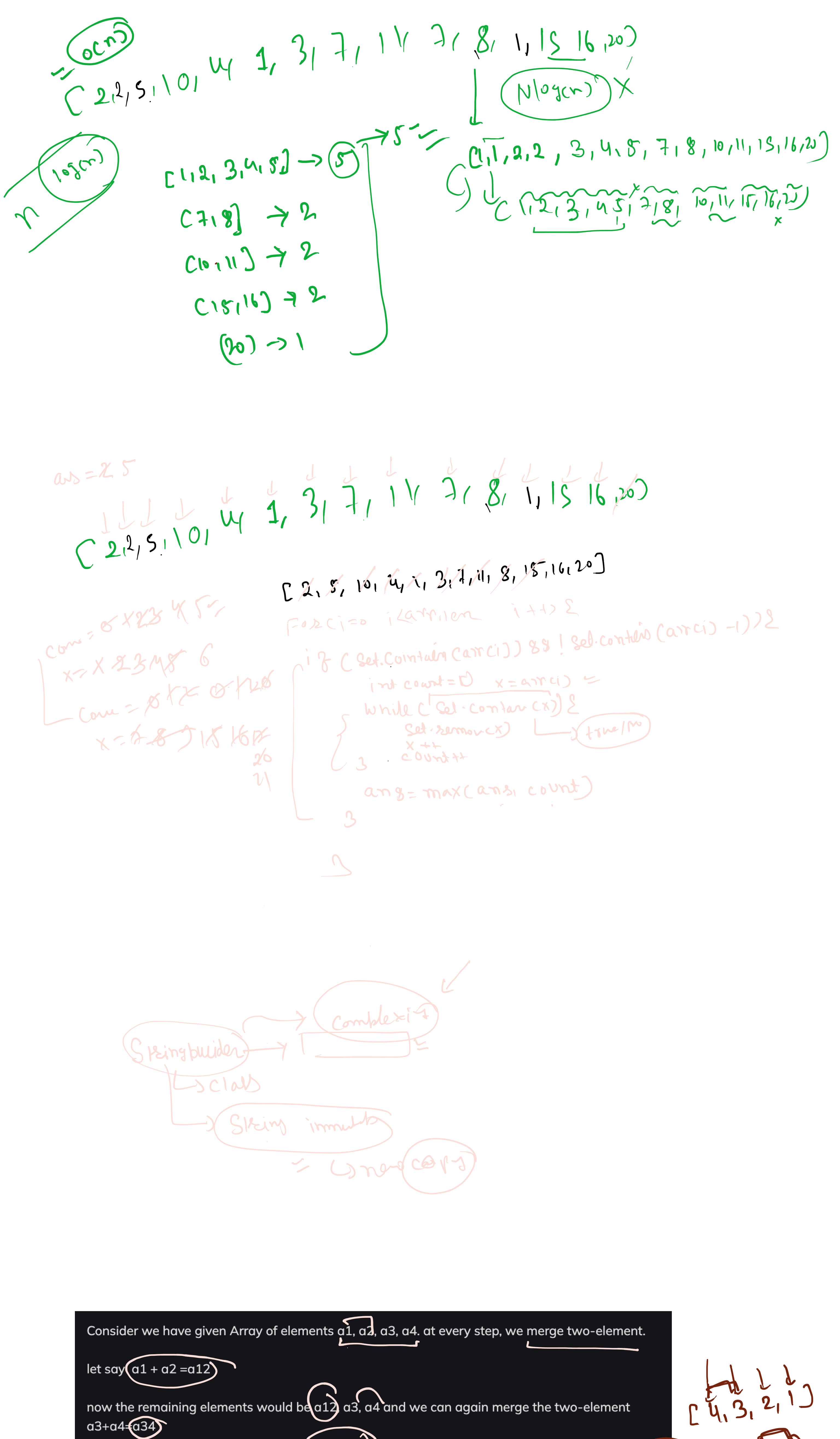
```
Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2,2]
```



```
public static int[] Intersection_of_Two_Arrays(int[] arr1, int[] arr2) {
        HashMap<Integer, Integer> map = new HashMap<>();
        for (int i = 0; i < arr2.length; i++) {
            if (map.containsKey(arr2[i])) {

                map.put(arr2[i], map.get(arr2[i]) + 1);
            } else {
                map.put(arr2[i], 1);
            }

        }

    }
```



```
        for (int i = 0; i < arr1.length; i++) {
                if (map.containsKey(arr1[i]) && map.get(arr1[i]) > 0) {
                    ll.add(arr1[i]);
                    map.put(arr1[i], map.get(arr1[i]) - 1);
                }

            }
```



Consider we have given Array of elements a1, a2, a3, a4. at every step, we merge two-element.

let say a1 + a2 =a12

now the remaining elements would be a12, a3, a4 and we can again merge the two-element a3+a4=a34

now the remaining elements would be a12, a34

finally, we will add the last two elements to get an a1234.
your task is to find the minimum sum of all the pairs getting added.