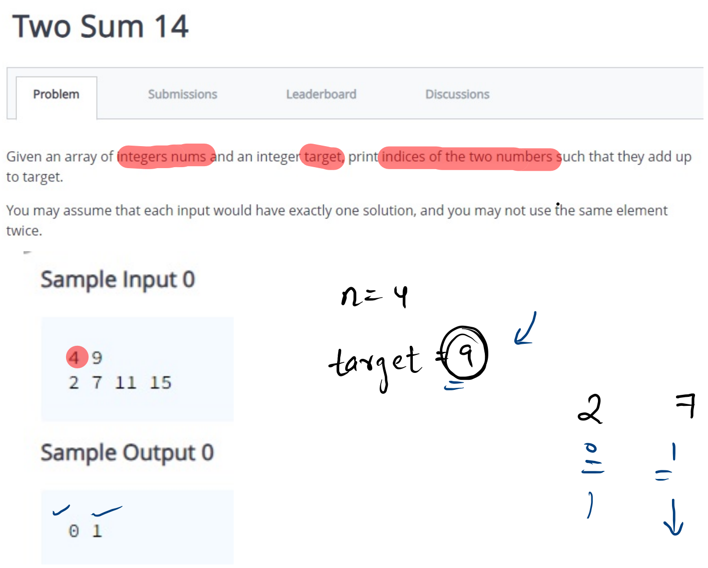


Two Sum 14

Problem | Submissions | Leaderboard | Discussions

Given an array of integers nums and an integer target, print indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

Sample Input 0

```
4 9
2 7 11 15
```

Sample Output 0

```
0 1
```

n= 4

target = $\boxed{9}$ ✓

2	7	11	15
0	1	2	3

tar = 11

eg.

15	8	6	1	5	7
0	1	2	3	4	5

one.
unique.

15	0
8	1
6	$\boxed{2}$

tar curr

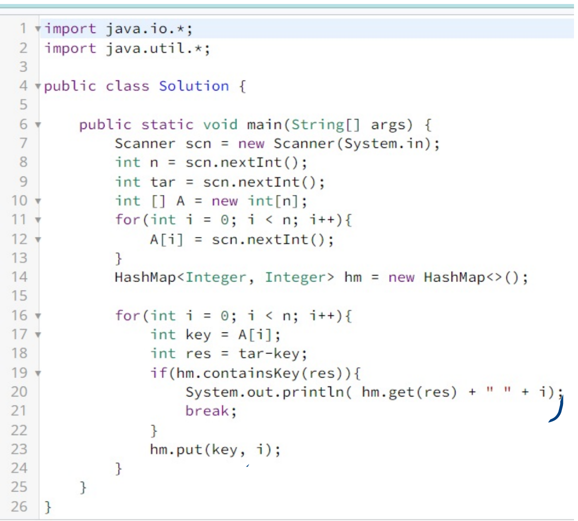
$11 - 6 = 5$

$11 - 1 = \boxed{10}$

$11 - \boxed{5} = \boxed{6}$ ✓

$\boxed{\text{'4' '2'}}$ ✓

$\boxed{2 \quad 4}$ ✓

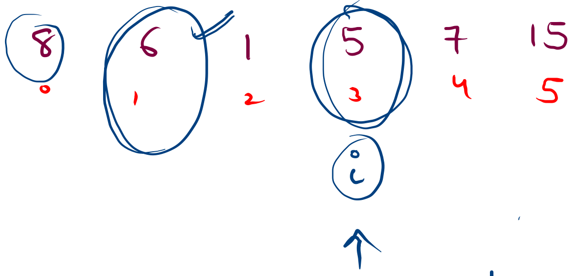


```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int tar = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }
        HashMap<Integer, Integer> hm = new HashMap<>();

        for(int i = 0; i < n; i++){
            int key = A[i];
            int res = tar-key;
            if(hm.containsKey(res)){
                System.out.println( hm.get(res) + " " + i);
                break;
            }
            hm.put(key, i);
        }
    }
}
```



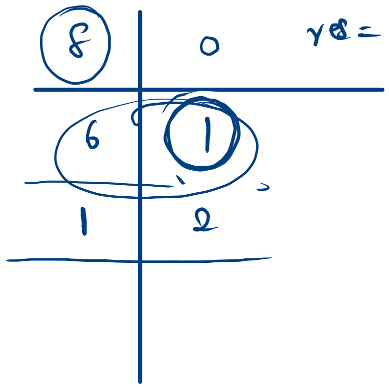
$n = 6$

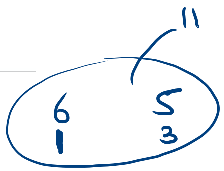
$tar = 11$

$key = 5$

$res = tar - key$

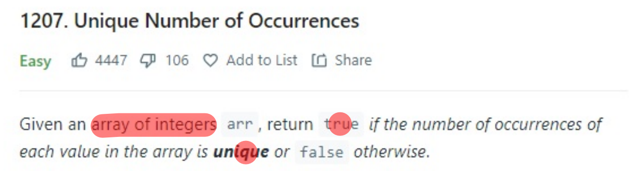
$= 11 - 5 = \boxed{6}$





key → ele

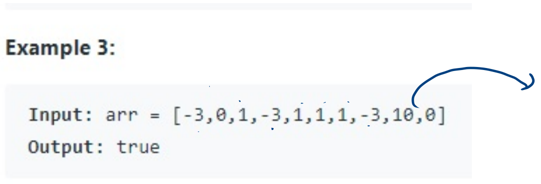
value → idx of ele.



1207. Unique Number of Occurrences

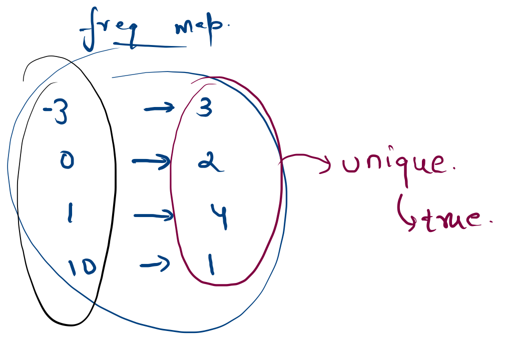
Easy 4447 106 Add to List Share

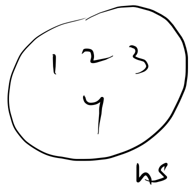
Given an array of integers `arr`, return `true` *if the number of occurrences of each value in the array is* ***unique*** *or* `false` *otherwise.*



Example 3:

```
Input: arr = [-3,0,1,-3,1,1,1,-3,10,0]
Output: true
```

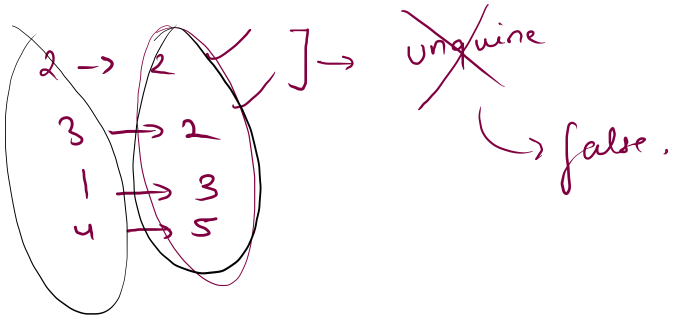


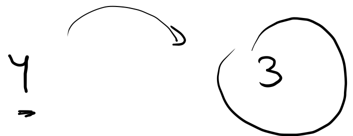


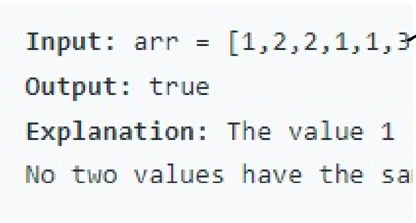
2 3 1 2 3 1 1 4 4 4 4 4

✓ 1. freq map.

✓ 2. values are unique.



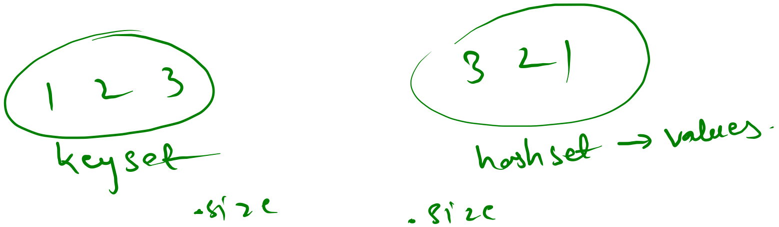




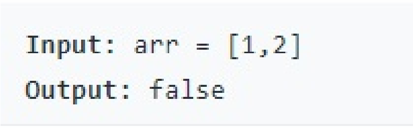
```
Input: arr = [1,2,2,1,1,3
Output: true
Explanation: The value 1
No two values have the sa
```

$\rightarrow$ 1 2 2 1 1 3

1 → 3
2 → 2
3 → 1



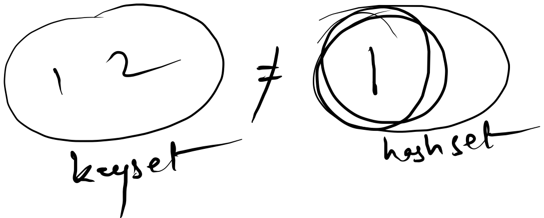
Example 2:



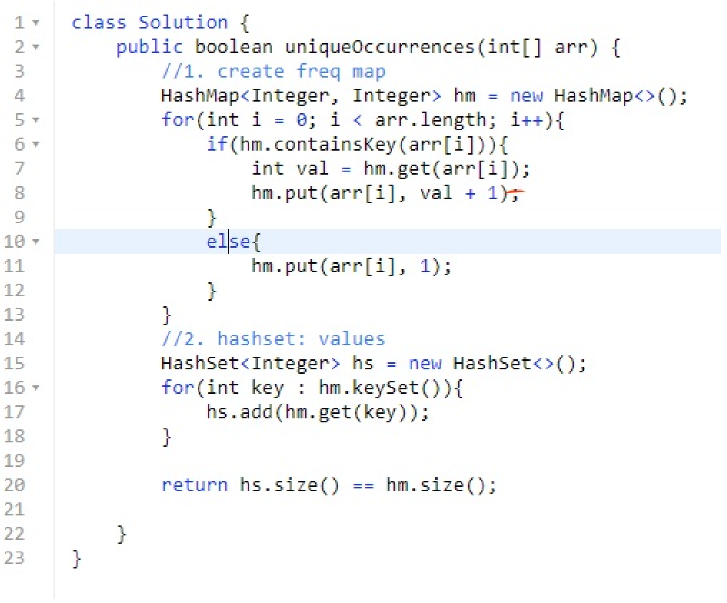
```
Input: arr = [1,2]
Output: false
```

eg. 2   1 2

1 → 1
2 → 1



$\boxed{\text{unique}}$



```java
class Solution {
    public boolean uniqueOccurrences(int[] arr) {
        //1. create freq map
        HashMap<Integer, Integer> hm = new HashMap<>();
        for(int i = 0; i < arr.length; i++){
            if(hm.containsKey(arr[i])){
                int val = hm.get(arr[i]);
                hm.put(arr[i], val + 1);
            }
            else{
                hm.put(arr[i], 1);
            }
        }
        //2. hashset: values
        HashSet<Integer> hs = new HashSet<>();
        for(int key : hm.keySet()){
            hs.add(hm.get(key));
        }

        return hs.size() == hm.size();

    }
}
```

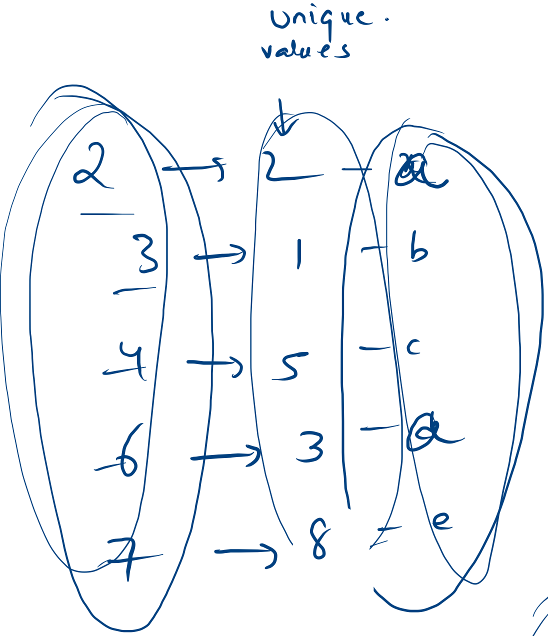
$\{1, 2, 3\}$

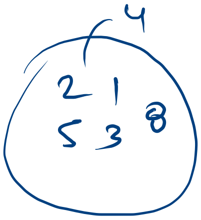
$\cancel{K}$ $\cancel{K}$ K

3

$K \longrightarrow V$

values.





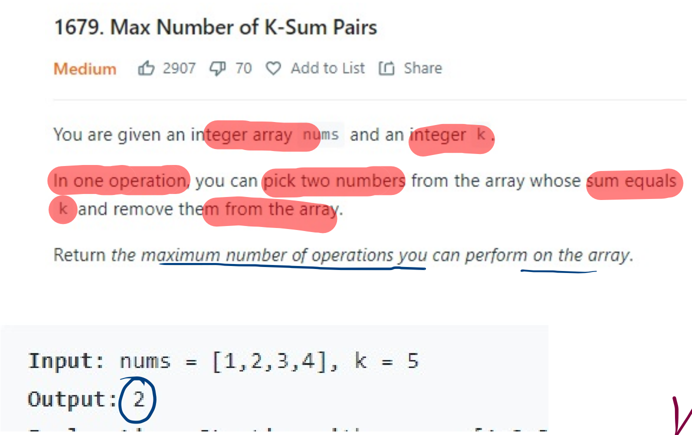
count of key = count of value

hm.size()

5 == ~~5~~ 4

== ==

key ~~key~~ value.



1679. Max Number of K-Sum Pairs

Medium 2907 70 Add to List Share

You are given an integer array `nums` and an integer `k`.

In one operation, you can pick two numbers from the array whose sum equals `k` and remove them from the array.

Return *the maximum number of operations you can perform on the array*.

```
Input: nums = [1,2,3,4], k = 5
Output: 2
```

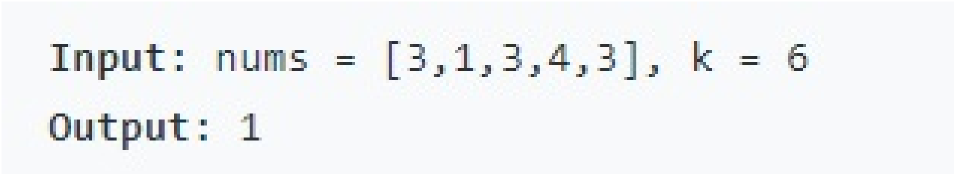
count = $\cancel{0}$ $\cancel{1}$ 2

k = 5

$\cancel{1}$ $\cancel{2}$ $\cancel{3}$ $\cancel{4}$

2 + 3

1 + 4



```
Input: nums = [3,1,3,4,3], k = 6
Output: 1
```

$k = 6$

step = 1

$\cancel{3}$ 1 $\cancel{3}$ 4 3

1 step $\longrightarrow$ 2 number

k = 6.

$\cancel{\boxed{3}}$ 1 $\cancel{4}$ $\downarrow \cancel{\boxed{3}}$ $\cancel{2}$ $\cancel{3}$ $\cancel{3}$ 3

i

count = ~~1~~ ~~2~~ $\boxed{3}$

freq map.

3	1
1	1

rest = 6 - 3 = $\boxed{3}$

$K = 5$

$ans = 2$

$\cancel{3} \quad \cancel{3} \quad \cancel{3} \quad \cancel{2} \quad \cancel{2} \quad \cancel{2} \quad 2$

i

$rest = 5 - 2 = \boxed{3}$

2	1

eg. ~~3~~ ~~3~~ ~~2~~ ~~2~~ 2 2

K=5

ans = ~~0~~ ~~1~~ 2

i

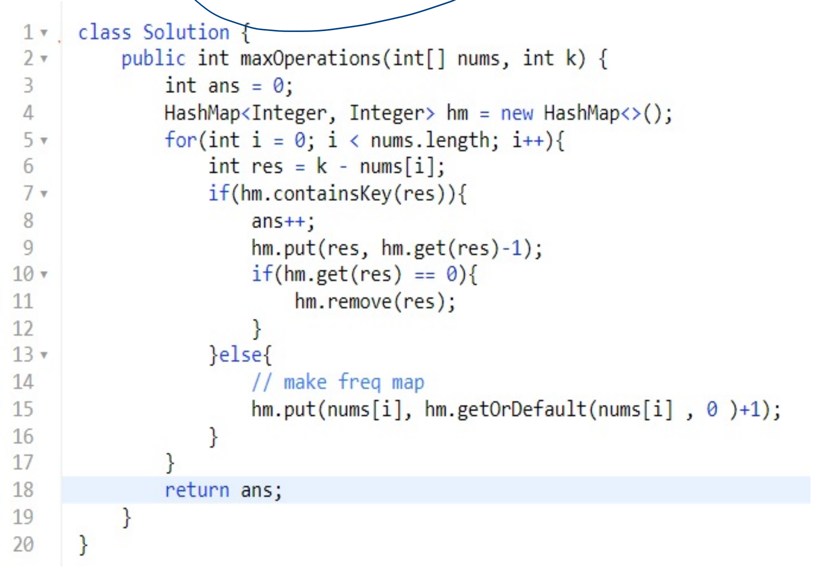
res = 5 - 2 = $\boxed{3}$

3, 1 -1

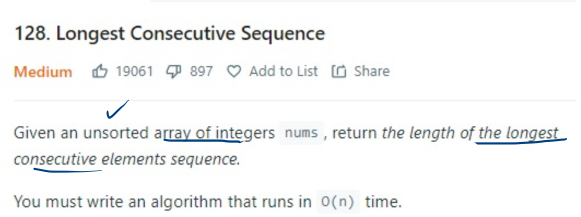
3, 0

0 == 0

2 ~~1~~ 2



```
class Solution {
    public int maxOperations(int[] nums, int k) {
        int ans = 0;
        HashMap<Integer, Integer> hm = new HashMap<>();
        for(int i = 0; i < nums.length; i++){
            int res = k - nums[i];
            if(hm.containsKey(res)){
                ans++;
                hm.put(res, hm.get(res)-1);
                if(hm.get(res) == 0){
                    hm.remove(res);
                }
            }else{
                // make freq map
                hm.put(nums[i], hm.getOrDefault(nums[i] , 0 )+1);
            }
        }
        return ans;
    }
}
```



128. Longest Consecutive Sequence

Medium 19061 897 Add to List Share

✓

Given an unsorted array of integers `nums`, return *the length of the longest consecutive elements sequence.*

You must write an algorithm that runs in `O(n)` time.

ans =

eg.

22	4	6	15	16	20	5	3	21	22	21
0	1	2	3	4	5	6	7	8	9	10

$\overset{0}{i}$

ans → $\boxed{4}$

filter duplicate.

14 15 16 17

17 - 14 - 1

= $\boxed{2}$

19 20 21 22 23

23 - 19 - 1

count = pre - ple - 1

23 - 19 - 1 = $\boxed{3.}$

2 3 4 5 6 7

pre - ple - 1

7 - 2 - 1 ✓

5 - 1 = $\boxed{4}$ ✓

Example 1:

```
Input: nums = [100,4,200,1,3,2]
Output: 4
Explanation: The longest consecutive elements sequence is [1, 2,
3, 4]. Therefore its length is 4.
```

Example 2:

```
Input: nums = [0,3,7,2,5,8,4,6,0,1]
Output: 9
```

0 3 7 2 5 8 4 6 0 1

0 $\boxed{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8}$

9

$\to O(n \log n)$

$\boxed{1\ 2\ 3\ 4}$ 100 200

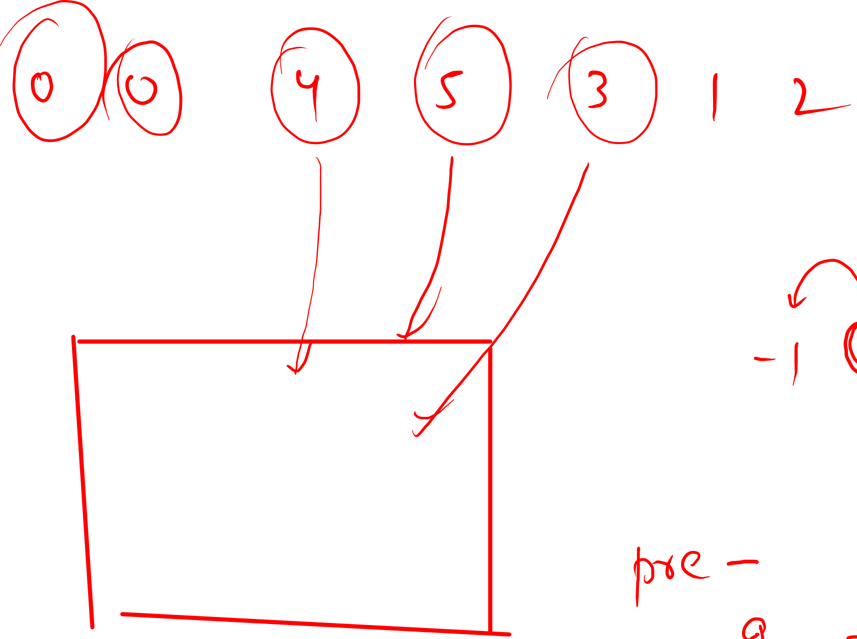
100 4 200 1 3 2

1 2 3 4 4

100 $\longrightarrow$ 1

200 $\longrightarrow$ 1

O(n)



-1
0
1
2
3
4
5
6
7
8
9

pre - ple - 1

$9 + 1 - 1 = \boxed{9}$