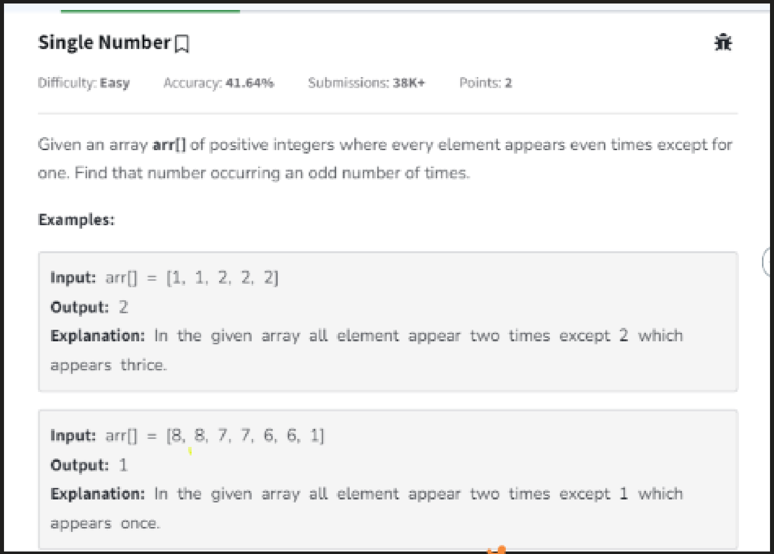


**Single Number**

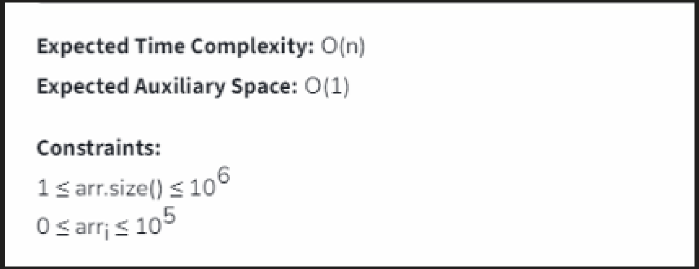
Difficulty: **Easy** Accuracy: **41.64%** Submissions: **38K+** Points: **2**

Given an array **arr[]** of positive integers where every element appears even times except for one. Find that number occurring an odd number of times.

**Examples:**

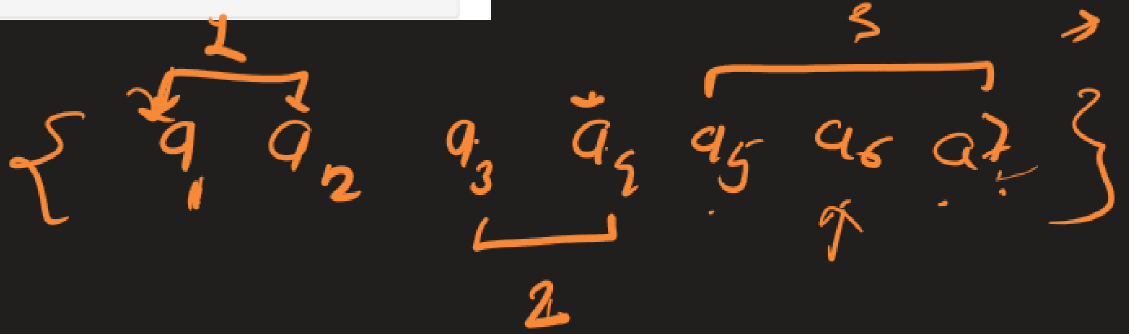
**Input:** arr[] = [1, 1, 2, 2, 2]
**Output:** 2
**Explanation:** In the given array all element appear two times except 2 which appears thrice.

**Input:** arr[] = [8, 8, 7, 7, 6, 6, 1]
**Output:** 1
**Explanation:** In the given array all element appear two times except 1 which appears once.



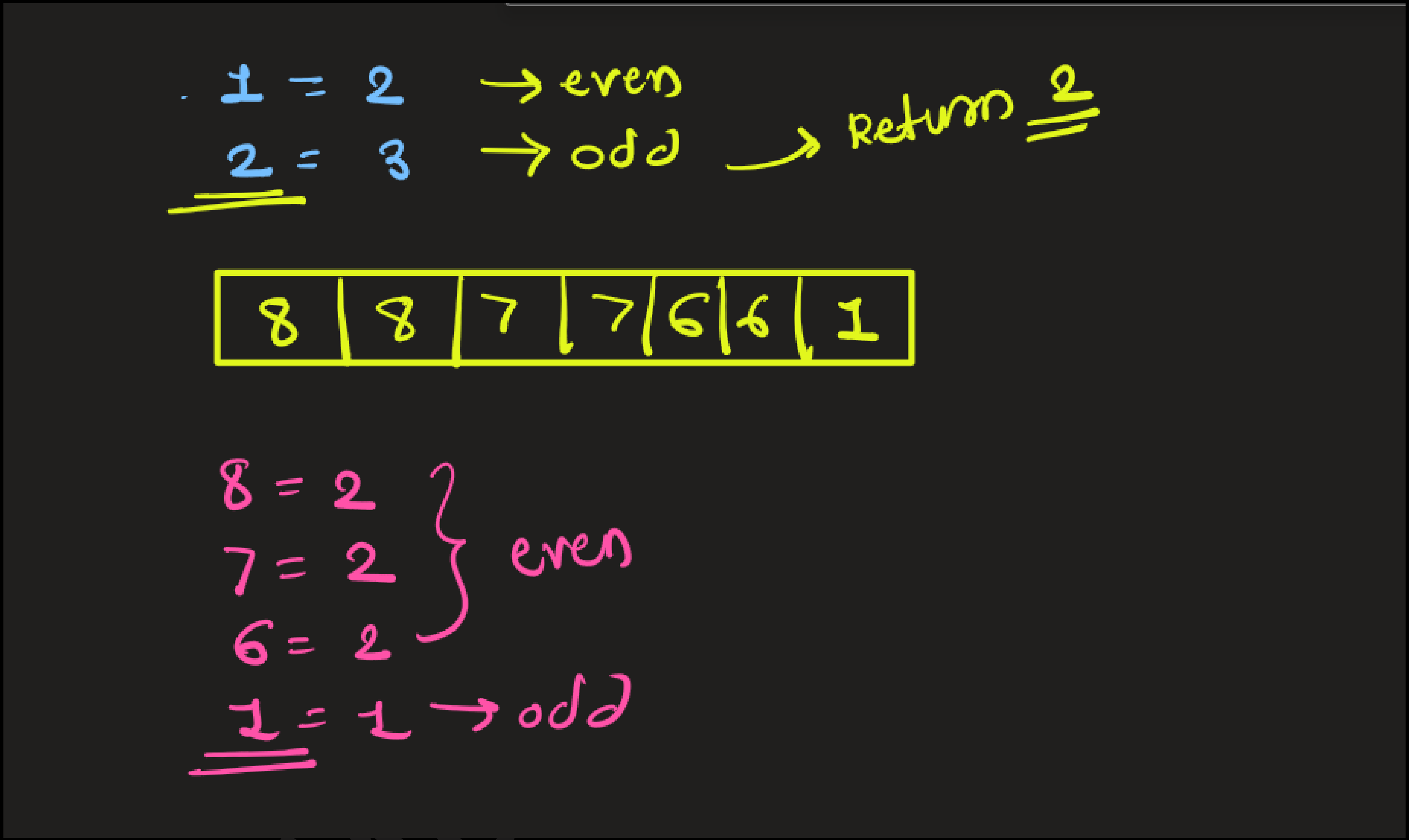
**Expected Time Complexity:** O(n)
**Expected Auxiliary Space:** O(1)

**Constraints:**
$1 \le arr.size() \le 10^6$
$0 \le arr_i \le 10^5$



|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 1 | 2 | 2 | 2 |

=



1 = 2 → even

2 = 3 → odd → Return 2

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 8 | 8 | 7 | 7 | 6 | 6 | 1 |

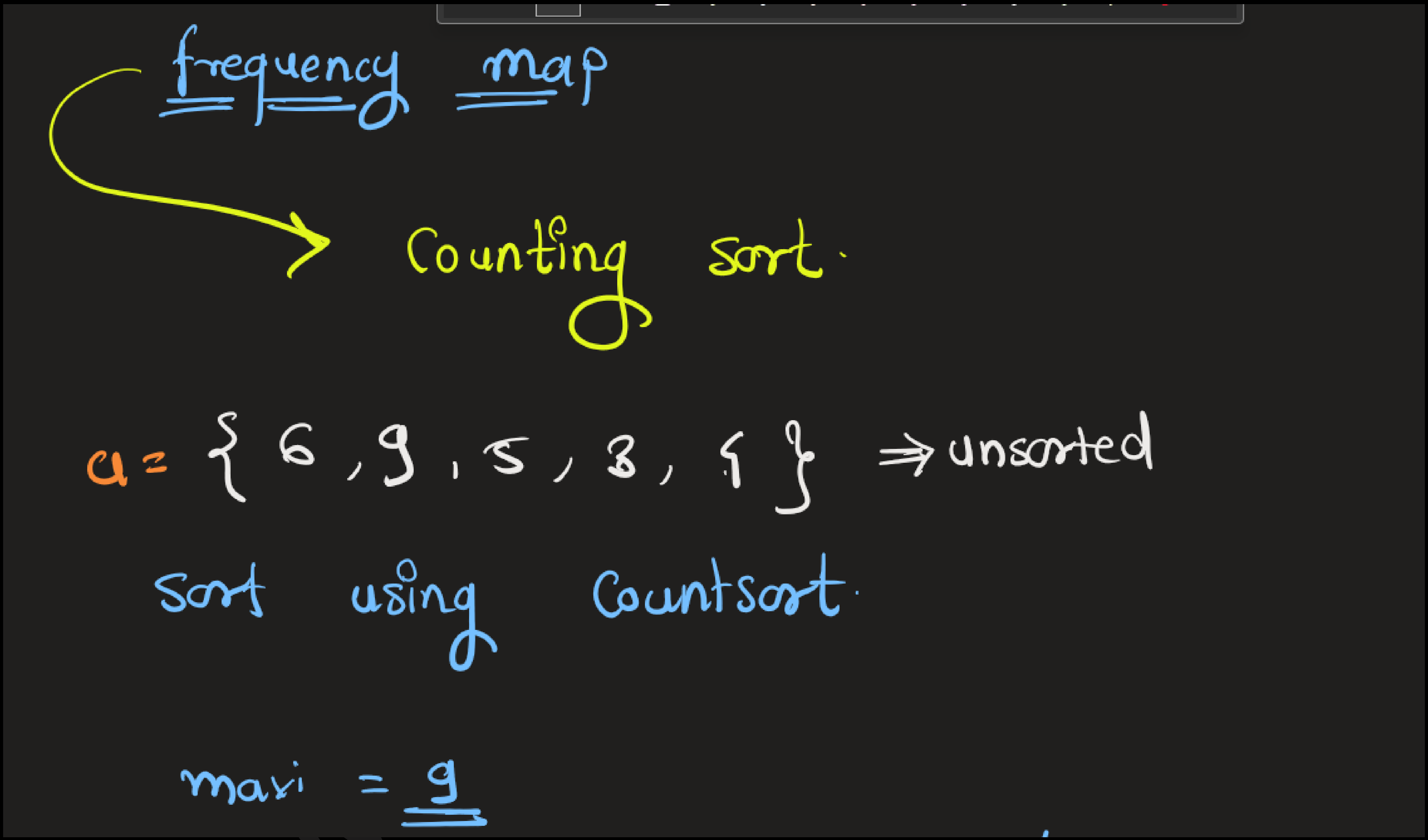
8 = 2

7 = 2

6 = 2

} even

1 = 1 → odd



frequency map

Counting sort.

$a = \{6, 9, 5, 3, 1\}$ ⇒ unsorted

sort using Countsort.

$maxi = \underline{9}$

int freq [] = new int [max+1]

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

⇒ freq

$$arr = \{\overset{0}{6}, \overset{1}{9}, 5, 3, 4\}$$

freq [arr[i]] += 1;

$$\underline{arr\ [0]}$$

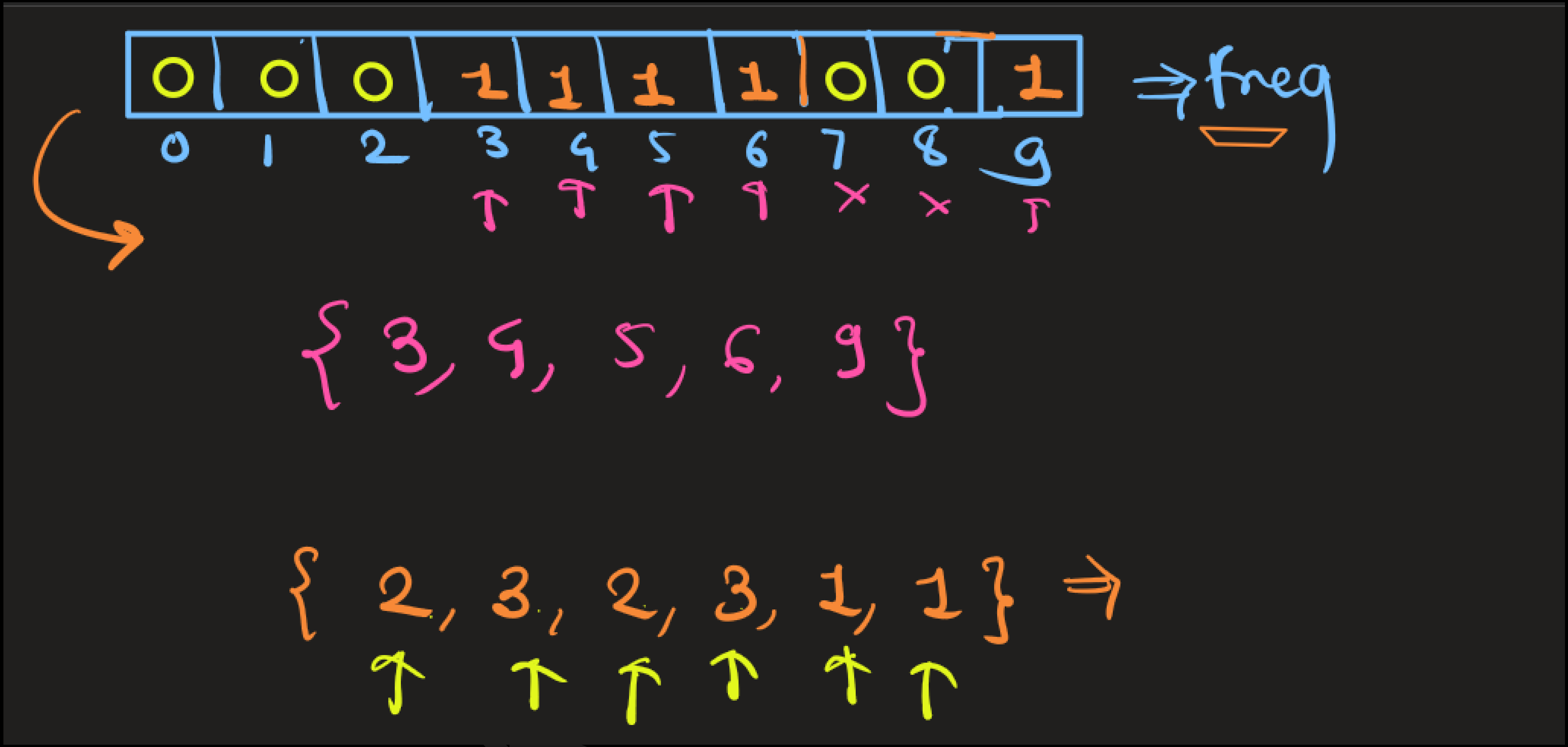
↓

freq [6] += 1

freq [3] += 1

freq [9] += 1

freq [5] += 1



|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|   |   |   | ↑ | ↑ | ↑ | ↑ | × | × | ↑ |

⇒ freq

{3, 4, 5, 6, 9}

{2, 3, 2, 3, 1, 1} ⇒

int freq = new int[4]

| | | | | |
|---|---|---|---|---|
| 0 | 2 | 2 | 2 | 0 |
| 0 | 1 | 2 | 3 | 4 |

freq[arr[i]] ++

freq[arr[i]] = $\overset{1}{\overline{\text{freq[arr[i]]}}}$ +1

1 +1

| | | | | |
|---|---|---|---|---|
| 0 | 2 | 2 | 2 | 0 |
| 0 | 1 | 2 | 3 | 4 |

while(

$\{\underline{1}, \overset{1}{1}, 2, 2, 3, 3\}$

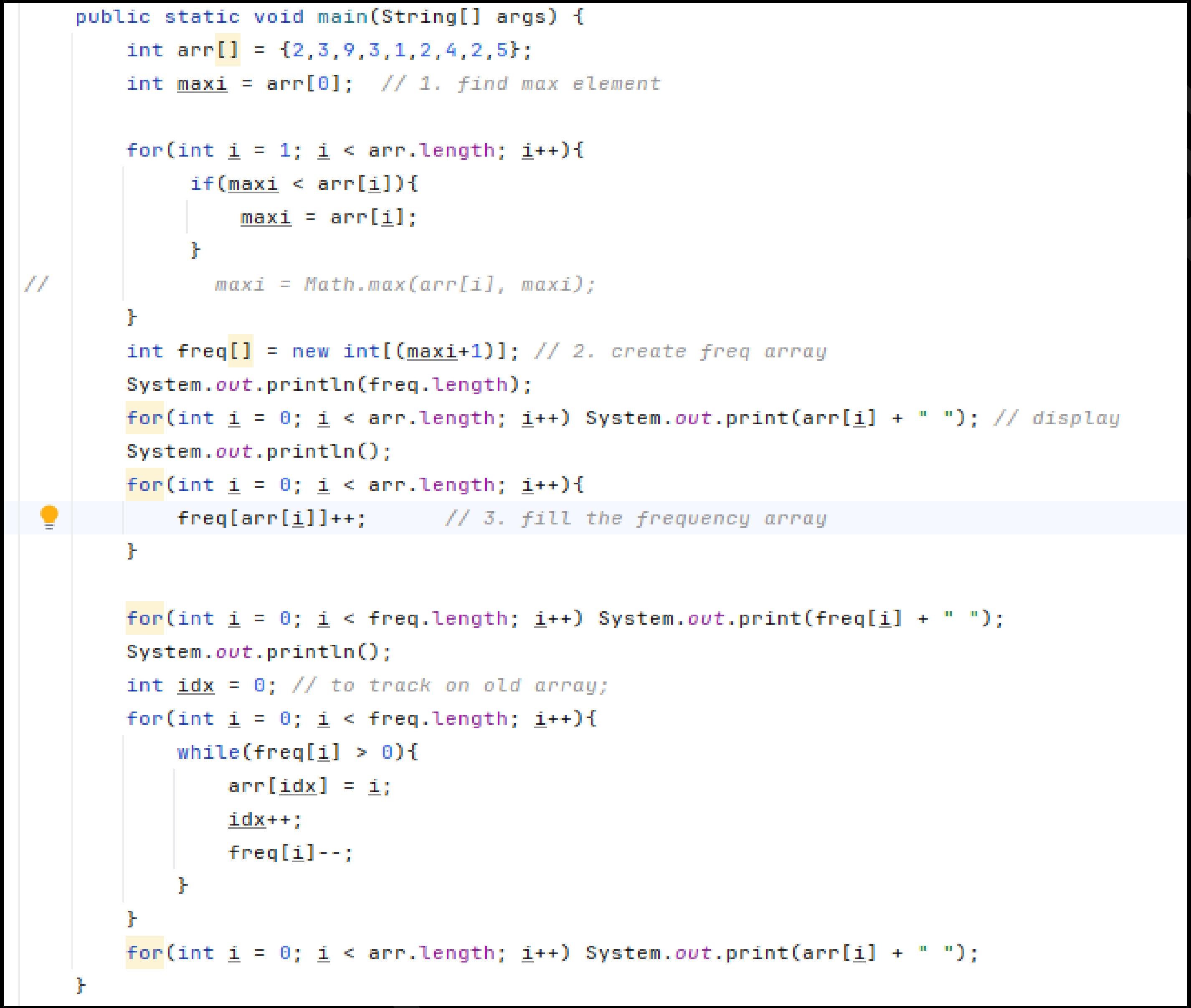
1) find max Element

```
int maxi = arr[0];

for (int i = 1; i <= arr.length; i++){
        maxi = Math.max(arr[i],
                        maxi);
}
```

2) Create new Array. freq

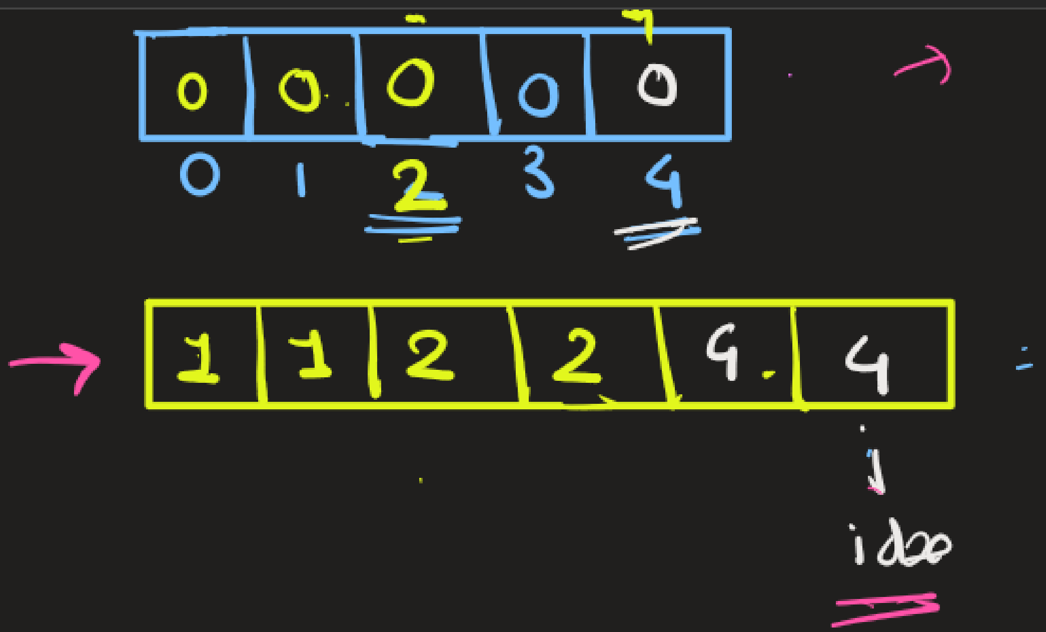
3) iterate on array arr & store
it's frequency in freq array



```java
    public static void main(String[] args) {
        int arr[] = {2,3,9,3,1,2,4,2,5};
        int maxi = arr[0];  // 1. find max element

        for(int i = 1; i < arr.length; i++){
            if(maxi < arr[i]){
                maxi = arr[i];
            }
//              maxi = Math.max(arr[i], maxi);
        }
        int freq[] = new int[(maxi+1)]; // 2. create freq array
        System.out.println(freq.length);
        for(int i = 0; i < arr.length; i++) System.out.print(arr[i] + " "); // display
        System.out.println();
        for(int i = 0; i < arr.length; i++){
            freq[arr[i]]++;      // 3. fill the frequency array
        }

        for(int i = 0; i < freq.length; i++) System.out.print(freq[i] + " ");
        System.out.println();
        int idx = 0; // to track on old array;
        for(int i = 0; i < freq.length; i++){
            while(freq[i] > 0){
                arr[idx] = i;
                idx++;
                freq[i]--;
            }
        }
        for(int i = 0; i < arr.length; i++) System.out.print(arr[i] + " ");
    }
```



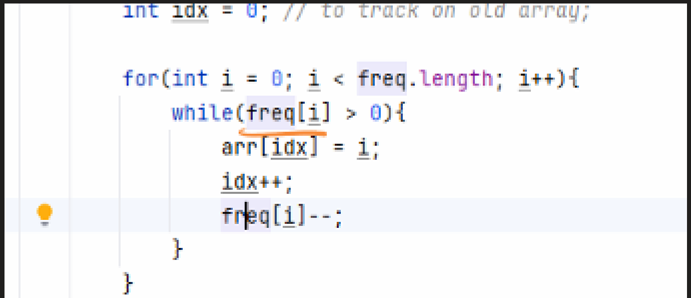
| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |

| 1 | 1 | 2 | 2 | 4. | 4 |
|---|---|---|---|---|---|

{ 2, 3, 9, 3, 1, 2, 4, 2, 5 }

| | |
|---|---|
| 1 = 1 | 4 = 1 |
| 2 = 3 | 5 = 1 |
| 3 = 2 | |
| 9 = 1 | |

{2, 2, 2}



```
int idx = 0; // to track on old array;

for(int i = 0; i < freq.length; i++){
    while(freq[i] > 0){
        arr[idx] = i;
        idx++;
        freq[i]--;
    }
}
```

{2, 5, 5

( >0) {

arr[idx] = 2

idx ++

3-- -

}

{ 2, 5, 2, 5, 7, 7 }

selectn $O(n^2)$

Bubble

Insertn

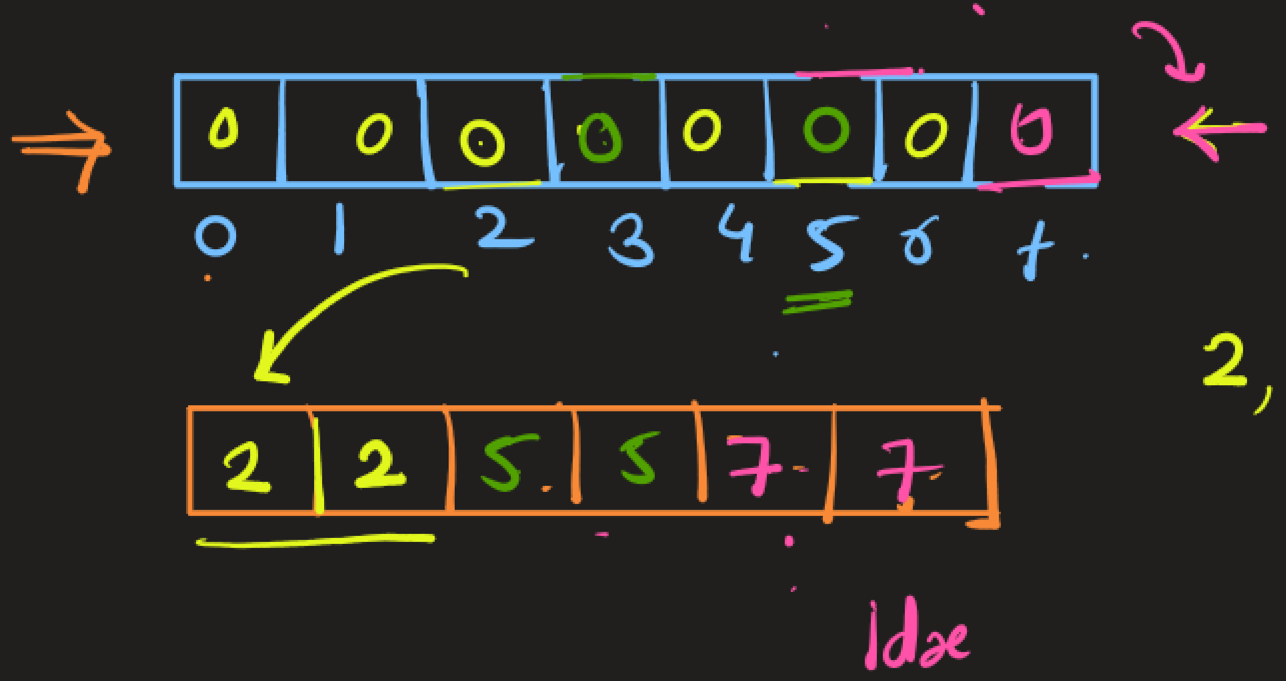
$O(n)$

freq[arr[i]] ++;

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 2 | 0 | 0 | 2 | 0 | 2 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

$freq[\underline{2}]{+}{+} = a{+}{+}, \quad a = a+1$

$\hookrightarrow arr[i] - arr[0] = \underline{\underline{2}}$

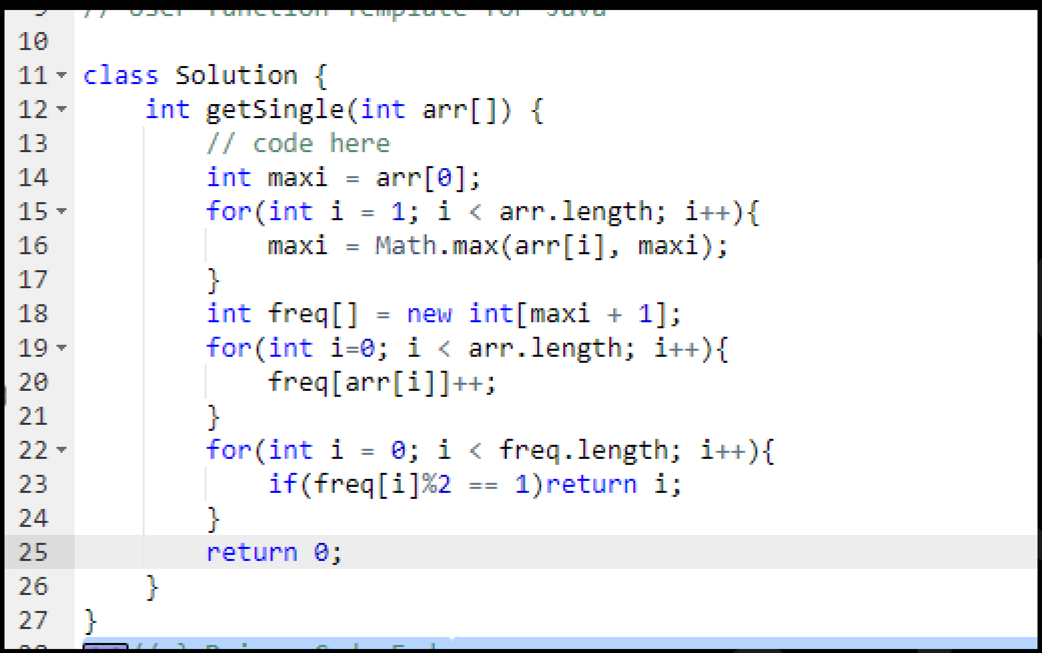


|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 2 | 2 | 5 | 5 | 7 | 7 |

2, 2, 5, 5, 7, 7

# Problem Solution :-



```java
// User function Template for Java

class Solution {
    int getSingle(int arr[]) {
        // code here
        int maxi = arr[0];
        for(int i = 1; i < arr.length; i++){
            maxi = Math.max(arr[i], maxi);
        }
        int freq[] = new int[maxi + 1];
        for(int i=0; i < arr.length; i++){
            freq[arr[i]]++;
        }
        for(int i = 0; i < freq.length; i++){
            if(freq[i]%2 == 1)return i;
        }
        return 0;
    }
}
```