Highest Occurring Element in an Array

Given an array **nums** of n integers, find the most frequent element in it i.e., the element that occurs the maximum number of times. If there are multiple elements that appear a maximum number of times, find the smallest of them.

Constraints:

- 1 <= n <= 10^5
- 1 <= nums[i] <= 10^4

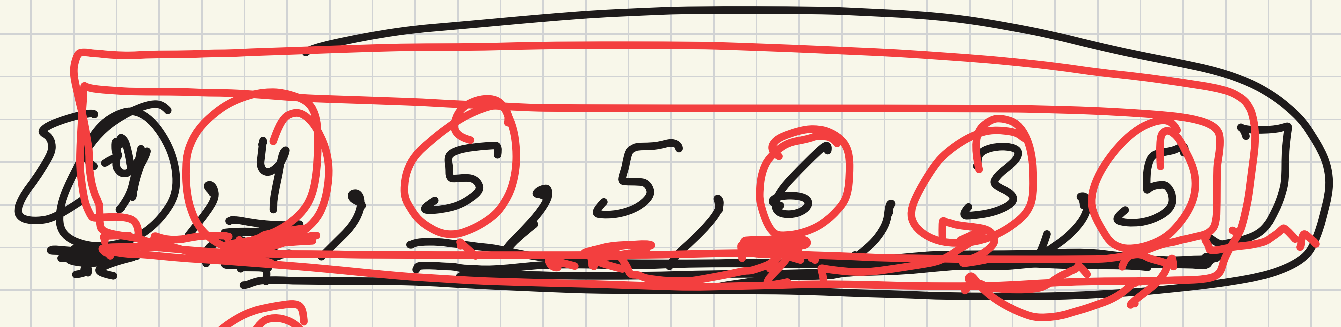
[1, 2, 2, 3, 3, 3]

1 → 1 time
2 → 2 times
3 → 3 times
→ 3

[4, 4, 5, 5, 6]

4 → 2 times
5 → 2 times
→ 4

Brute Force →



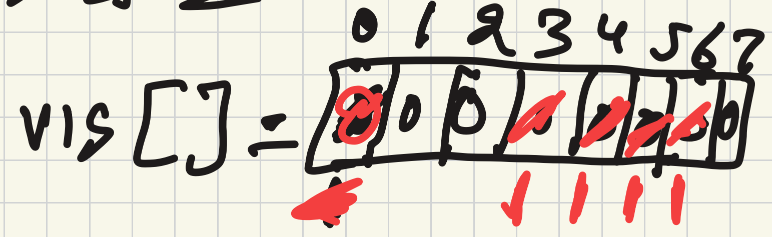
maxCount = ~~0~~ ~~2~~ 3
el = ~~-1~~ ~~4~~ 5

→ 5

count = ~~0~~ ~~1~~ 2

$(10^4 + 1)$

(max+1)



```
int func(arr)
    vis[] = [10^4+1]
    for(0 -> n-1)
        cnt=0
        if( vis[arr[i]] == 0)
            vis[arr[i]] = 1
            for(j -> 0 -> n-1)
                if (arr[i] == arr[j])
                    cnt++
            }
        }
        if ( cnt > maxCount)
            maxCount = cnt
            el = arr[i]
        else if (cnt == maxCount)
            el = Math.min(el, arr[i])
    }
}
```

TC -> $\boxed{O(N^2)}$

SC -> $O(10^4)$

$\boxed{[1,2,3,4,5]}$

Optimised Solution →

arr[] = [4, 4, 5, 5, 6, 2, 3, ~~5~~]

hash[] =

0	1	2	3	4	5	6
0	0	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~
		1	1	~~1~~	~~1~~	1
				2	~~2~~	
					3	

TC → O(N) + O(N) + O(N)

TC → 3O(N)

SC → O(max of array)

```
for (0 → n − 1) {
    hash[arr[i]]++
3
for (0 → n − 1) {
    if (hash[i] > maxCount) {
        maxCount = hash[i]
        el = i
3       3
```

3

maxCount = ~~0~~ ~~1~~ 2

el = ~~-1~~ ~~2~~ 4

5

[4, 4, 5, 5, 6]

0	1	2	3	4	5	~~6~~
0	0	0	0	2	2	1

maxCount = 2

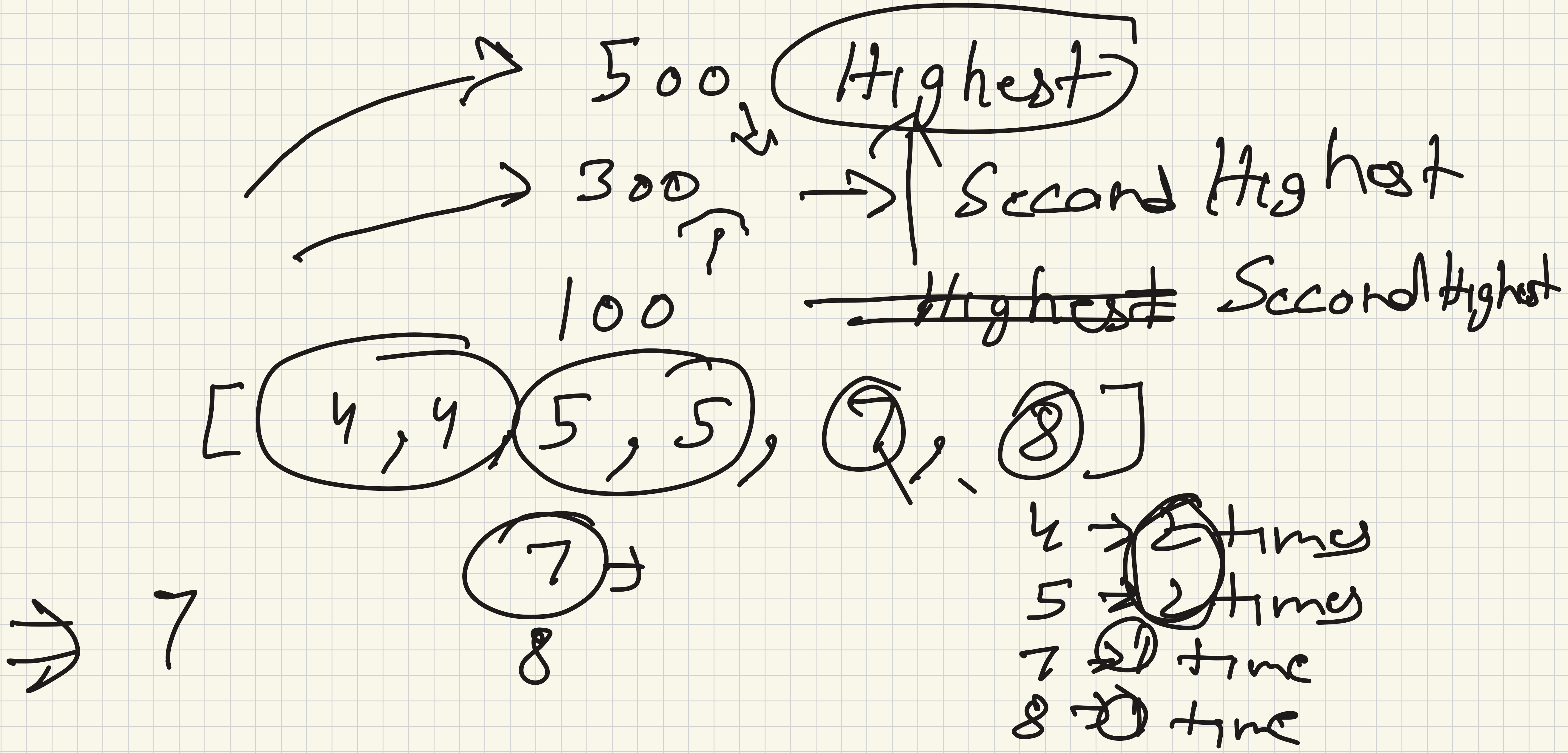
el = 4 → 4

Second Highest Occurring Element in an Array

Given an array of n integers, find the second most frequent element in it.
If there are multiple elements that appear second most frequent times, find the smallest of them.
If second most frequent element does not exist return -1.

Constraints:

- $1 <= n <= 10^5$
- $1 <= nums[i] <= 10^4$



Brute Force →

[4, 4, 5, 5, 6, 7, 5]

vis = [0 | 1 | 1 | 1 | 1]

maxCount = ~~0~~ ~~2~~ 3

el = ~~-1~~ ~~4~~ 5

secondMaxCount = ~~0~~ ~~0~~ 2

secondEl = ~~-1~~ ~~1~~ 4

count = ~~0~~ ~~2~~ ~~3~~ ~~1~~ 1

→ 4

[1, 2, 3, 4, 5]

```
int func (arr)
    visited [] = [10^4]
    for (i -> 0 -> n-1)
        if (visited[arr[i]] == 0)
            cnt = 0
            for (j -> 0 -> n-1) {
                if (arr[i] == arr[j])
                    count++
            }
```

TC → $O(N^2)$

SC → $O(10^4)$

```
if (cnt > maxCount) {
        secondMaxCount = maxCount
        secondEl = el
        maxCount = cnt
        el = arr[i]
} else if (cnt == maxCount) {
        el = Math.min(el, arr[i])
} else if (cnt > secondMaxCount)
        secondMaxCount = cnt
        secondEl = arr[i]
else if (cnt == secondMaxCount)
        secondEl = Math.min(secondEl, arr[i])
}
return secondEl
```

Optimised →

[4, 4, 5, 5, 6, 7, 5]

max = MIN_VALUE

1. Find max in arr →

```
for (0 -> n-1)
    max = Math.max(max, arr[i])
```

3

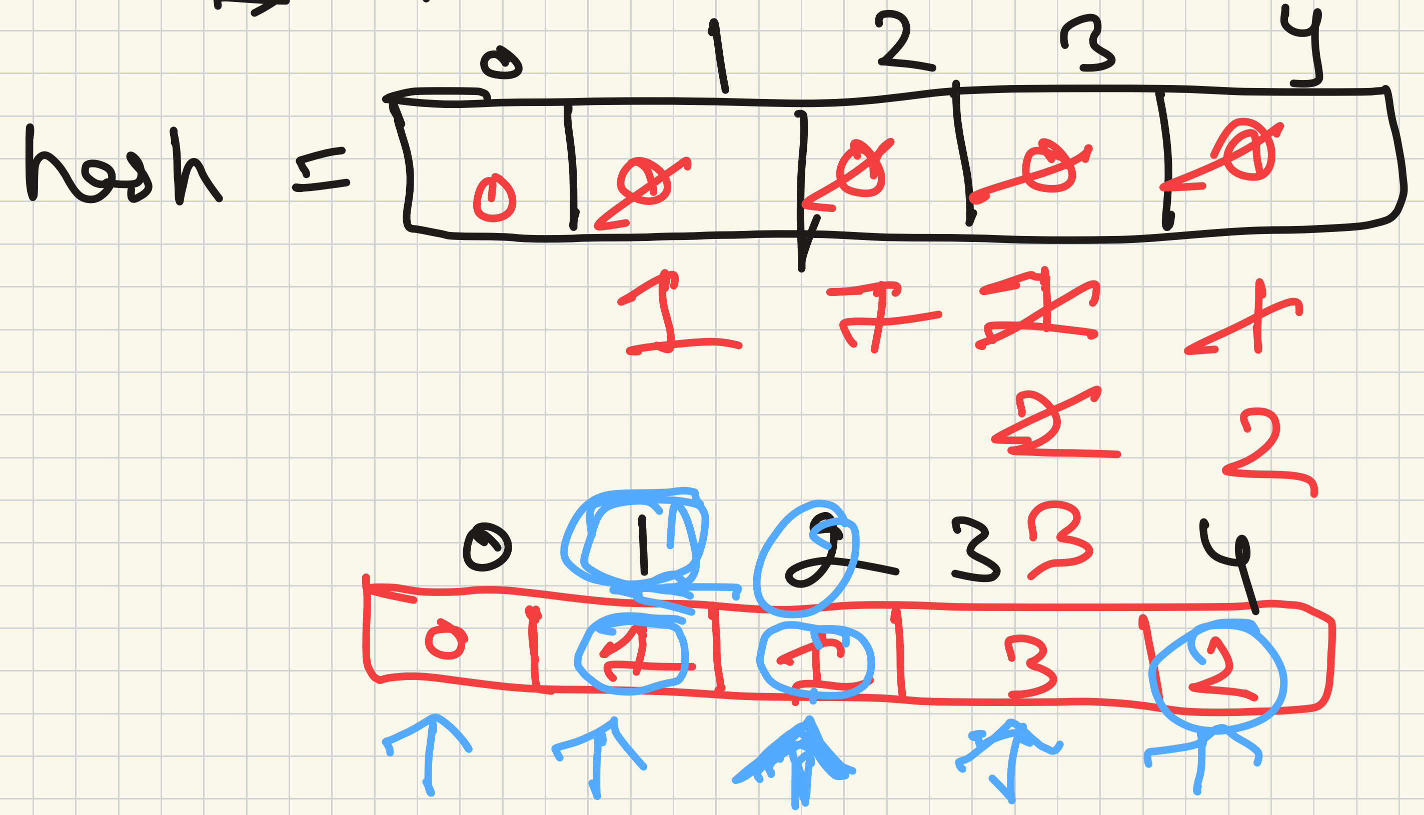
2. Create hash arr with size max +1

hash[] = [max+1]

3. Run loop over arr & fill hash.

[4, 4, 3, 3, 1, 2, 3]

⇒ 4



4. Loop over hash arr ⇒

maxCount = ~~0~~ ~~2~~ ~~1~~ 3 secondMaxCount = 0 2 ~~1~~ 2
el = ~~-1~~ ~~1~~ 3 secondEl = ~~-1~~ -1 ~~1~~ 4

⇒ 4

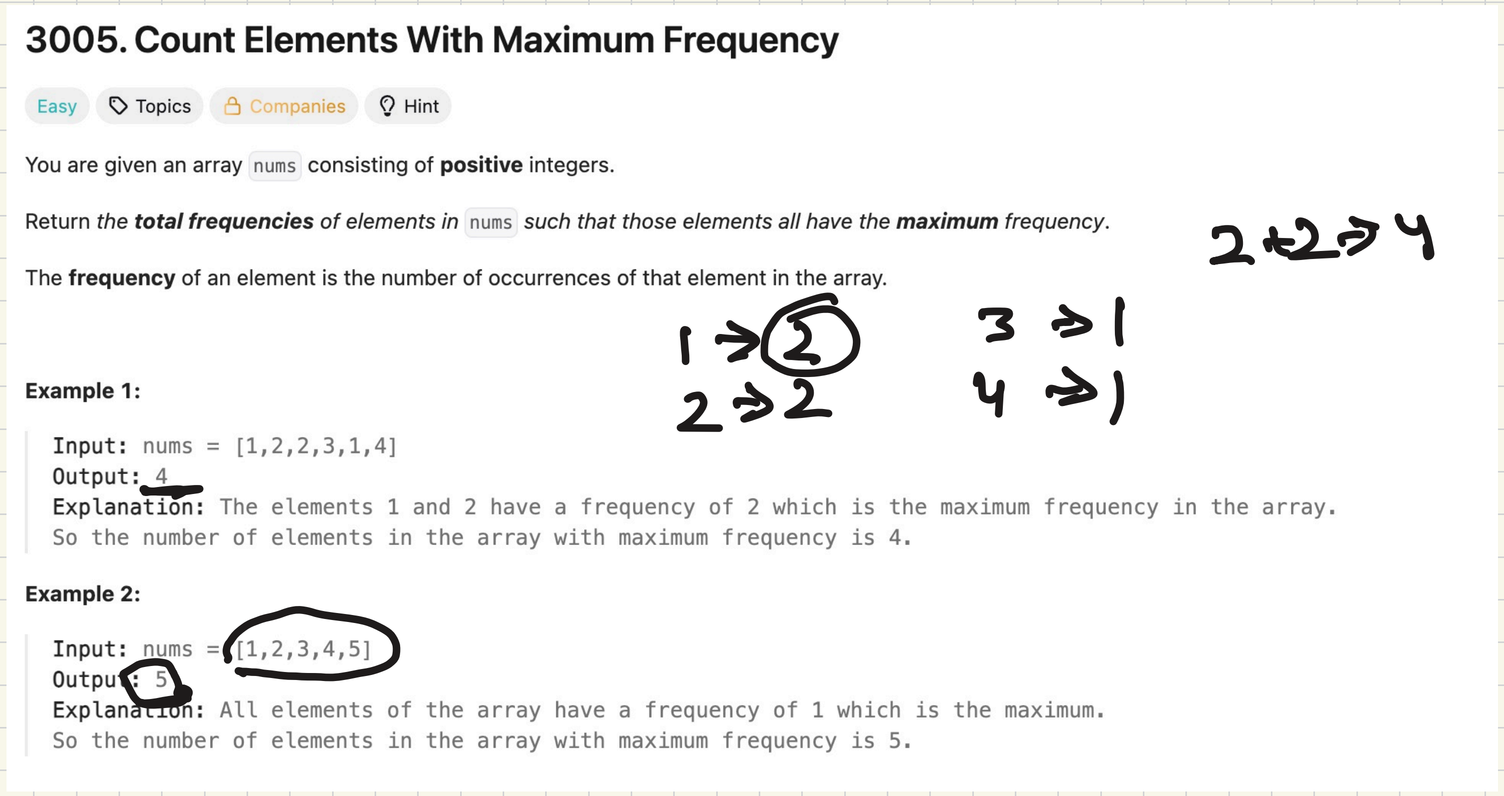
```
int func (arr)
    max = MIN_VALUE
    for (i → 0 → n-1)
            max = Math.max(max, arr[i])
    }
hash[] = [max+1]
for ( i → 0 → n - 1 )
        hash [arr [i]] ++
maxCount = 0
   el = -1
secondMaxCount = 0
 second El = -1
for ( i -> 0 -> n-1)
    if ( hash[i] > maxCount){
        secondMaxCount = maxCount
          second El = el
         maxCount = hash[i]
            el = i
    } else if ( hash[i] > secondMaxCount
          && hash[i] < maxCount)
          secondMaxCount = hash[i]
             Second El = i
    }
```

TC -> O(N) + O(N) + O(N)
-> 3 * O(N)
SC -> O(max element of array)

return secondEl

Today's Motivation

"You have the right to perform your prescribed duty, but you are not entitled to the fruits of your actions."



3005. Count Elements With Maximum Frequency

Easy · Topics · Companies · Hint

You are given an array `nums` consisting of **positive** integers.

Return *the **total frequencies** of elements in* `nums` *such that those elements all have the **maximum** frequency*.

The **frequency** of an element is the number of occurrences of that element in the array.

$1 \to 2$

$2 \to 2$

$3 \to 1$

$4 \to 1$

$2+2 \to 4$

Example 1:

```
Input: nums = [1,2,2,3,1,4]
Output: 4
Explanation: The elements 1 and 2 have a frequency of 2 which is the maximum frequency in the array.
So the number of elements in the array with maximum frequency is 4.
```

Example 2:

```
Input: nums = [1,2,3,4,5]
Output: 5
Explanation: All elements of the array have a frequency of 1 which is the maximum.
So the number of elements in the array with maximum frequency is 5.
```