3375. Minimum Operations to Make Array Values Equal to K

Easy | Topics | Companies | Hint

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

An integer `h` is called **valid** if all values in the array that are **strictly greater** than `h` are *identical*.

For example, if `nums = [10, 8, 10, 8]`, a **valid** integer is `h = 9` because all `nums[i] > 9` are equal to 10, but 5 is not a **valid** integer.

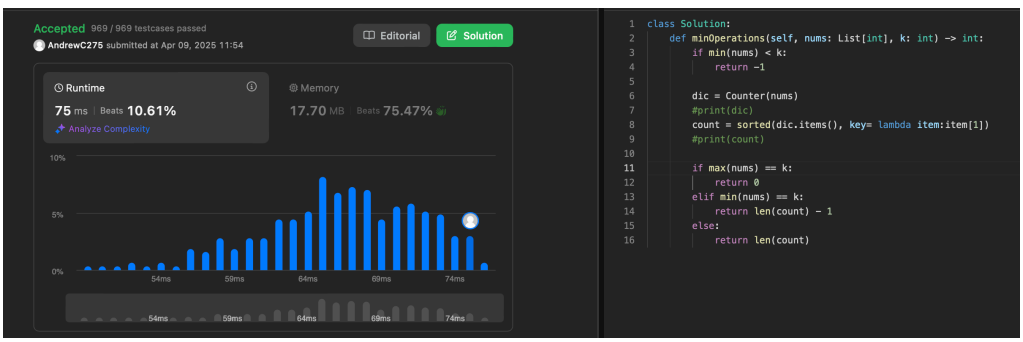
You are allowed to perform the following operation on `nums`:

- Select an integer `h` that is *valid* for the **current** values in `nums`.
- For each index `i` where `nums[i] > h`, set `nums[i]` to `h`.

Return the **minimum** number of operations required to make every element in `nums` **equal** to `k`. If it is impossible to make all elements equal to `k`, return -1.

Example 1:

> **Input:** `nums = [5,2,5,4,5], k = 2`
>
> **Output:** 2
>
> **Explanation:**
>
> The operations can be performed in order using valid integers 4 and then 2.

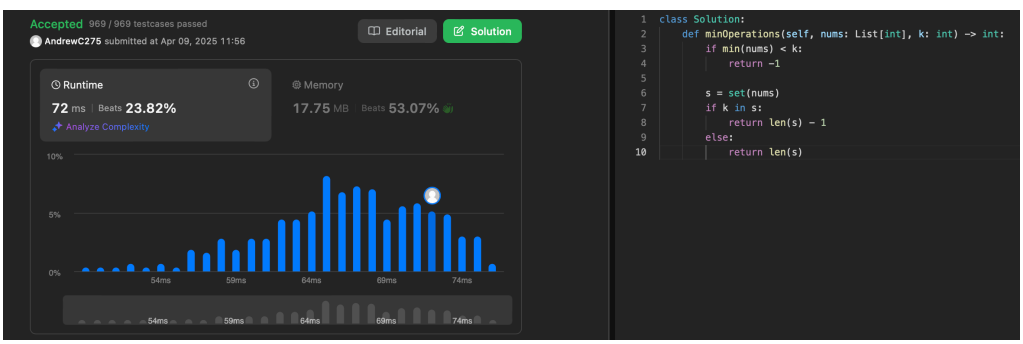



```python
class Solution:
    def minOperations(self, nums: List[int], k: int) -> int:
        if min(nums) < k:
            return -1

        dic = Counter(nums)
        #print(dic)
        count = sorted(dic.items(), key= lambda item:item[1])
        #print(count)

        if max(nums) == k:
            return 0
        elif min(nums) == k:
            return len(count) - 1
        else:
            return len(count)
```

# of each val doesn't matter




```python
class Solution:
    def minOperations(self, nums: List[int], k: int) -> int:
        if min(nums) < k:
            return -1

        s = set(nums)
        if k in s:
            return len(s) - 1
        else:
            return len(s)
```