

2870. Minimum Number of Operations to Make Array Empty

Medium

1.2K

59

Companies

You are given a **0-indexed** array **nums** consisting of positive integers.

There are two types of operations that you can apply on the array **any** number of times:

- Choose **two** elements with **equal** values and **delete** them from the array.
- Choose **three** elements with **equal** values and **delete** them from the array.

Return the **minimum** number of operations required to make the array empty, or **-1** if it is not possible.

Example 1:

Input: `nums = [2,3,3,2,2,4,2,3,4]`

Output: 4

Explanation: We can apply the following operations to make the array empty:

- Apply the first operation on the elements at indices 0 and 3. The resulting array is `nums = [3,3,2,4,2,3,4]`.
- Apply the first operation on the elements at indices 2 and 4. The resulting array is `nums = [3,3,4,3,4]`.
- Apply the second operation on the elements at indices 0, 1, and 3. The resulting array is `nums = [4,4]`.
- Apply the first operation on the elements at indices 0 and 1. The resulting array is `nums = []`.

It can be shown that we cannot make the array empty in less than 4 operations.

Example 2:

Input: `nums = [2,1,2,2,3,3]`

Output: -1

Explanation: It is impossible to empty the array.

Constraints:

- $2 \leq \text{nums.length} \leq 10^5$
- $1 \leq \text{nums}[i] \leq 10^5$

```

function minOperations($nums) {
    $freq = array_count_values($nums);
    $operations = 0;

    foreach ($freq as $count) {
        if($count % 3 == 0){
            $operations += (int) ($count / 3);
        }elseif($count % 2 == 0){
            $operations += (int) ($count / 2);
        }else{
            return -1;
        }
    }

    return $operations;
}

```

300. Longest Increasing Subsequence

Given an integer array `nums`, return *the length of the longest **strictly increasing***

subsequence

.

.

Example 1:

Input: `nums = [10,9,2,5,3,7,101,18]`

Output: `4`

Explanation: The longest increasing subsequence is `[2,3,7,101]`, therefore the length is 4.

Example 2:

Input: `nums = [0,1,0,3,2,3]`

Output: `4`

Example 3:

Input: `nums = [7,7,7,7,7,7,7]`

Output: `1`

```
function lengthOfLIS($nums) {  
    if (empty($nums)) {  
        return 0;  
    }  
  
    $n = count($nums);  
    $dp = array_fill(0, $n, 1);  
  
    for ($i = 1; $i < $n; $i++) {  
        for ($j = 0; $j < $i; $j++) {  
            if ($nums[$i] > $nums[$j]) {  
                $dp[$i] = max($dp[$i], $dp[$j] + 1);  
            }  
        }  
    }  
  
    return max($dp);  
}
```