Day3

December 19, 2023 11:28 AM

367. Valid Perfect Square

Given a positive integer num, return true *if* num *is a perfect square or* false *otherwise*.
A **perfect square** is an integer that is the square of an integer. In other words, it is the product of some integer with itself.
You must not use any built-in library function, such as sqrt.

Example 1:
Input: num = 16
Output: true
Explanation: We return true because 4 * 4 = 16 and 4 is an integer.
Example 2:
Input: num = 14
Output: false
Explanation: We return false because 3.742 * 3.742 = 14 and 3.742 is not an integer.

Constraints:

- $1 <= num <= 2^{31} - 1$

From <https://leetcode.com/problems/valid-perfect-square/>

Thought : Binary search

80. Remove Duplicates from Sorted Array II

Medium
6.3K
1.2K
Companies
Given an integer array nums sorted in **non-decreasing order**, remove some duplicates **in-place** such that each unique element appears **at most twice**. The **relative order** of the elements should be kept the **same**.
Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the **first part** of the array nums. More formally, if there are k elements after removing the duplicates, then the first k elements of nums should hold the final result. It does not matter what you leave beyond the first k elements.
Return k *after placing the final result in the first* k *slots of* nums.
Do **not** allocate extra space for another array. You must do this by **modifying the input array in-place** with O(1) extra memory.
Custom Judge:
The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int[] expectedNums = [...]; // The expected answer with correct length
int k = removeDuplicates(nums); // Calls your implementation
assert k == expectedNums.length;
for (int i = 0; i < k; i++) {
    assert nums[i] == expectedNums[i];
}
```

If all assertions pass, then your solution will be **accepted**.

Example 1:
Input: nums = [1,1,1,2,2,3]
Output: 5, nums = [1,1,2,2,3,_]
Explanation: Your function should return k = 5, with the first five elements of nums being 1, 1, 2, 2 and 3 respectively.
It does not matter what you leave beyond the returned k (hence they are underscores).
Example 2:
Input: nums = [0,0,1,1,1,1,2,3,3]
Output: 7, nums = [0,0,1,1,2,3,3,_,_]
Explanation: Your function should return k = 7, with the first seven elements of nums being 0, 0, 1, 1, 2, 3 and 3 respectively.
It does not matter what you leave beyond the returned k (hence they are underscores).

From <https://leetcode.com/problems/remove-duplicates-from-sorted-array-ii/description/>

26. Remove Duplicates from Sorted Array

Easy
13.3K
17.6K
Companies
Given an integer array nums sorted in **non-decreasing order**, remove the duplicates **in-place** such that each unique element appears only **once**. The **relative order** of the elements should be kept the **same**. Then return *the number of unique elements in* nums.
Consider the number of unique elements of nums to be k, to get accepted, you need to do the following things:

- Change the array nums such that the first k elements of nums contain the unique elements in the order they were present in nums initially. The remaining elements of nums are not important as well as the size of nums.
- Return k.

Custom Judge:
The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int[] expectedNums = [...]; // The expected answer with correct length
int k = removeDuplicates(nums); // Calls your implementation
assert k == expectedNums.length;
for (int i = 0; i < k; i++) {
    assert nums[i] == expectedNums[i];
}
```

If all assertions pass, then your solution will be **accepted**.

Example 1:
Input: nums = [1,1,2]
Output: 2, nums = [1,2,_]
Explanation: Your function should return k = 2, with the first two elements of nums being 1 and 2 respectively.
It does not matter what you leave beyond the returned k (hence they are underscores).
Example 2:
Input: nums = [0,0,1,1,1,2,2,3,3,4]
Output: 5, nums = [0,1,2,3,4,_,_,_,_,_]
Explanation: Your function should return k = 5, with the first five elements of nums being 0, 1, 2, 3, and 4 respectively.
It does not matter what you leave beyond the returned k (hence they are underscores).

Constraints:

- $1 <= nums.length <= 3 * 10^4$
- -100 <= nums[i] <= 100
- nums is sorted in **non-decreasing** order.

From <https://leetcode.com/problems/remove-duplicates-from-sorted-array/>

283. Move Zeroes

Easy
15.8K
410
Companies
Given an integer array nums, move all 0's to the end of it while maintaining the relative order of the non-zero elements.
Note that you must do this in-place without making a copy of the array.

Example 1:
Input: nums = [0,1,0,3,12]
Output: [1,3,12,0,0]
Example 2:
Input: nums = [0]
Output: [0]

From <https://leetcode.com/problems/move-zeroes/>

844. Backspace String Compare

Easy
7.4K
340
Companies
Given two strings s and t, return true *if they are equal when both are typed into empty text editors*. '#' means a backspace character.
Note that after backspacing an empty text, the text will continue empty.

Example 1:
Input: s = "ab#c", t = "ad#c"
Output: true
Explanation: Both s and t become "ac".
Example 2:
Input: s = "ab##", t = "c#d#"
Output: true
Explanation: Both s and t become "".
Example 3:
Input: s = "a#c", t = "b"
Output: false
Explanation: s becomes "c" while t becomes "b".

From <https://leetcode.com/problems/backspace-string-compare/>