26. Remove Duplicates from Sorted Array

March 5, 2016 array (/articles/?tag=array)

Question | Editorial Solution

Question

> Given a sorted array, remove the duplicates in place such that each element appear only *once* and return the new length.
>
> Do not allocate extra space for another array, you must do this in place with constant memory.
>
> For example,
> Given input array *nums* = `[1,1,2]`,
>
> Your function should return length = `2`, with the first two elements of *nums* being `1` and `2` respectively. It doesn't matter what you leave beyond the new length.

Quick Navigation

Solution

Approach #1 (Two Pointers) [Accepted]

Algorithm

Since the array is already sorted, we can keep two pointers i and j, where i is the slow-runner while j is the fast-runner. As long as $nums[i] = nums[j]$, we increment j to skip the duplicate.

When we encounter $nums[j] \neq nums[i]$, the duplicate run has ended so we must copy its value to $nums[i+1]$. i is then incremented and we repeat the same process again until j reaches the end of array.

```
public int removeDuplicates(int[] nums) {
    if (nums.length == 0) return 0;
    int i = 0;
    for (int j = 1; j < nums.length; j++) {
        if (nums[j] != nums[i]) {
            i++;
            nums[i] = nums[j];
        }
    }
    return i + 1;
}
```

Complexity analysis

- Time complextiy : $O(n)$. Assume that n is the length of array. Each of i and j traverses at most n steps.
- Space complexity : $O(1)$.

(/ratings/107/4/?return=/articles/remove-duplicates-sorted-array/) (/ratings/107/4/?return=/articles/remove-duplicates-sorted-array/) (/ratings/107/4/?return=/articles/remove-

Average Rating: 4.30 (125 votes)

Previous

Next > (/articles/valid-anagram/)

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

Join the conversation

Signed in as **lixiaozheng.good**.

Post a Reply

sophyg commented 2 months ago

nice solution

iscuss.leetcode.com/user/sophyg)

ash.ps312 commented 2 months ago

Doesn't list(set(array)) (in python) work as well? or does it create a new list?

iscuss.leetcode.com/user/ash-

GregBoy34 commented 2 months ago

I agree that remove is misleading here. Especially in an online environment where you dont have an interviewer to bounce questions off. However, learning still took place here.

iscuss.leetcode.com/user/gregboy34)

irtazasafi commented 3 months ago

Pathetically worded problem. Wasted a lot of time for me.

iscuss.leetcode.com/user/irtazasafi)

GregBoy34 commented 3 months ago

Thanks @cjgali1 (https://discuss.leetcode.com/uid/142609) for your feedback. @StefanPochmann (https://discuss.leetcode.com/uid/591) I think grouping "sloppy/noobs" together, regardless of the possibility that a person could be a "noob", "sloppy" or a "noob" and "sloppy" during a public discussion is not helpful to anyone working on this problem and could be left out. While correct (about a lot of things), the way you have responded here in a group setting could be improved. Unique returns a "past the end iterator", performing pointer arithmetic on a "placeholder" reference should have an undesired result. I have just started looking at c++ again here recently so I not 100% sure on that. Can you explain if I am incorrect in my observations of your solution in a professional manner?

iscuss.leetcode.com/user/gregboy34)

cjgali1 commented 3 months ago

@StefanPochmann (https://discuss.leetcode.com/uid/591) I agree that anyone who fails to read the problem has no business complaining, but simply downvoting those who think the description is poor is just as much being a noob as failing to read the question. Is this an environment for learning or is this 4chan? If someone thinks there's an issue, maybe ask them why.

I don't think anyone said "this function is unreasonable," the point is that the description of the problem is unclear to some people. Why not ask them what they think is unclear, rather than downvoting? I think calling it "removal" is intentionally deceptive, considering the documentation for unique (while still using the word "removal") specifically states that nothing is being removed (as I mentioned in my prior post), but rather, that elements are being swapped in-place.

Again, the answer can simply be "hey, I think you're misinterpreting this for X reason," but to downvote without explanation is not conducive to anyone here learning the algorithm in question.

iscuss.leetcode.com/user/cjgali1)

StefanPochmann commented 3 months ago

@cjgali1 (https://discuss.leetcode.com/uid/142609) The problem is alright, people are just being sloppy/noobs. The problem says "Your function should return length = 2, **with the first two elements of nums being 1 and 2 respectively**". Not the problem's fault if people ignore that and pretend it isn't there. And several people clearly **tried** to change the array by using something like `nums = ...`. Not the problem's fault if they don't know their language and don't realize that that doesn't do what they think it does.

iscuss.leetcode.com/user/stefanpochmann)

It's also not an unreasonable function, it's almost the same as one of the C++ standard, namely unique (http://en.cppreference.com/w/cpp/algorithm/unique), which also is for removing duplicates and also returns the new end point. It can even be used to trivially solve this problem:

```
class Solution {
public:
    int removeDuplicates(vector<int>& nums) {
        return unique(nums.begin(), nums.end()) - nums.begin();
    }
};
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

cjgali1 commented 3 months ago

@9431 (https://discuss.leetcode.com/uid/164197) Unfortunately that solution won't work because you do not alter the array in place, as the question specifies. In theory, saying something like "nums = set(nums)" would work, but in python, sets are implemented using dictionaries, so this violates the O(1) memory requirement, and will also not pass the judge.

All of that said, it's pretty pathetic that whoever made this problem decided to go through these comments and downvote anyone who noted the obviously poor description of the problem, rather than providing an explanation for the problem's description / reasoning for why it is not unclear or misleading, as many people have noted.

Nothing is being "removed" from the array at any point, the duplicate elements are just being swapped with their following element until the initial array [1,2,2,2,3,3,3,4,4,4] is something akin to [1,2,3,4,3,3,3,4,4,4]. So nothing is ever necessarily removed, certain array elements are simply overwritten by the values of other array elements. Hence, the wording of the original question IS misleading and unclear, and the question could do with being renamed for the sake of clarity.

iscuss.leetcode.com/user/cjgali1)

GregBoy34 commented 3 months ago

I am still a bit confused with this problem.... @9431 (https://discuss.leetcode.com/uid/164197), I believe the problem is meant to show a strategy for iterating through a list/tree in an array, without necessarily removing the duplicates, just assigning the elements following the given pattern. The sample states that the first step should appear correct and it does not matter what follows as long as you return the correct sum of the list as if duplicates were removed.. Does that seem accurate?

iscuss.leetcode.com/user/gregboy34)

saravanam commented 3 months ago

- (void)removeDuplicatesFromSortedArray:(NSMutableArray *)iArray {
  //1,1,2,3,4,4,5,6,7,7,8,11
  if (iArray.count == 0) {
  return;
  }
  NSUInteger length = iArray.count - 1;

  for (int j = 0; j < length; j++) {
  if ([iArray[j] isEqual:iArray[j+1]]) {
  [iArray removeObjectAtIndex:j];
  length = length - 1;
  }
  }
  NSLog(@"array after removing duplicate elements..%@", iArray);
  }

O(n) and O(1)

iscuss.leetcode.com/user/saravanam)

View original thread (https://discuss.leetcode.com/topic/26)

Load more comments...