448. Find All Numbers Disappeared in an Array

Add to List ▾

Question | Editorial Solution | My Submissions (/problems/find-all-numbers-disappeared-in-an-array/submissions/)

Total Accepted: **20926** Total Submissions: **38197** Difficulty: **Easy** Contributors: **yuhaowang001** (/yuhaowang001/)

Given an array of integers where 1 ≤ a[i] ≤ *n* (*n* = size of array), some elements appear twice and others appear once.

Find all the elements of [1, *n*] inclusive that do not appear in this array.

Could you do it without extra space and in O(*n*) runtime? You may assume the returned list does not count as extra space.

Example:

```
Input:
[4,3,2,7,8,2,3,1]

Output:
[5,6]
```

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/575) | Top Solutions | Pick One (/problems/random-one-question/)

C++ ▾

Notes

```
class Solution {
public:
    vector<int> findDisappearedNumbers(vector<int>& nums) {
        int n = nums.size();
        vector<int> res;
        for(int i = 0;i<n;i++){
            int m = abs(nums[i])-1;
            nums[m] = nums[m]<0?nums[m]:-nums[m];
        }
        for(int i = 0;i<n;i++){
            if(nums[i]>0) res.push_back(i+1);
        }
        return res;
    }
};
```

Custom Testcase □

Contribute Testcase ❓

Run Code | Submit Solution