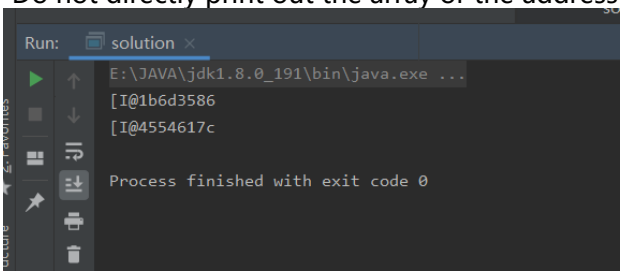
双指针

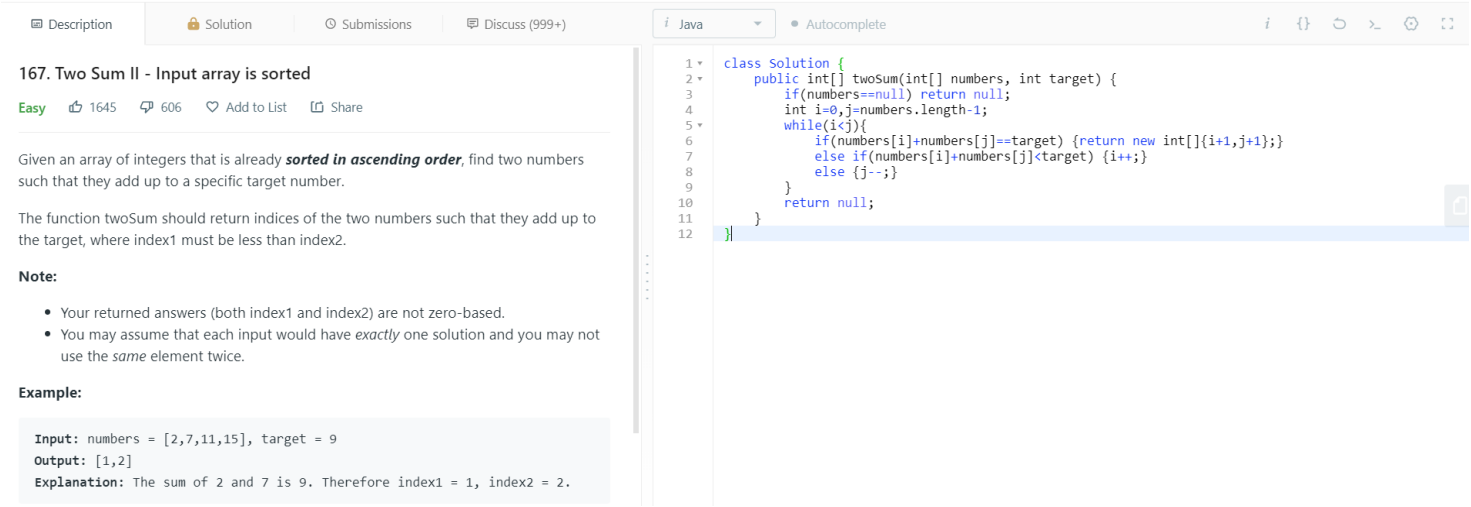
笔记本: LeetCode
创建时间: 2020/7/4 19:19 **更新时间:** 2020/7/7 23:24
作者: 粥粥

1. two sum (Input array is sorted)

   https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/description/
   One point from the start and one point form the end to traverse the array.
   Compare sum to target. Note the boundary conditions.
   Do not directly print out the array or the address would be got.

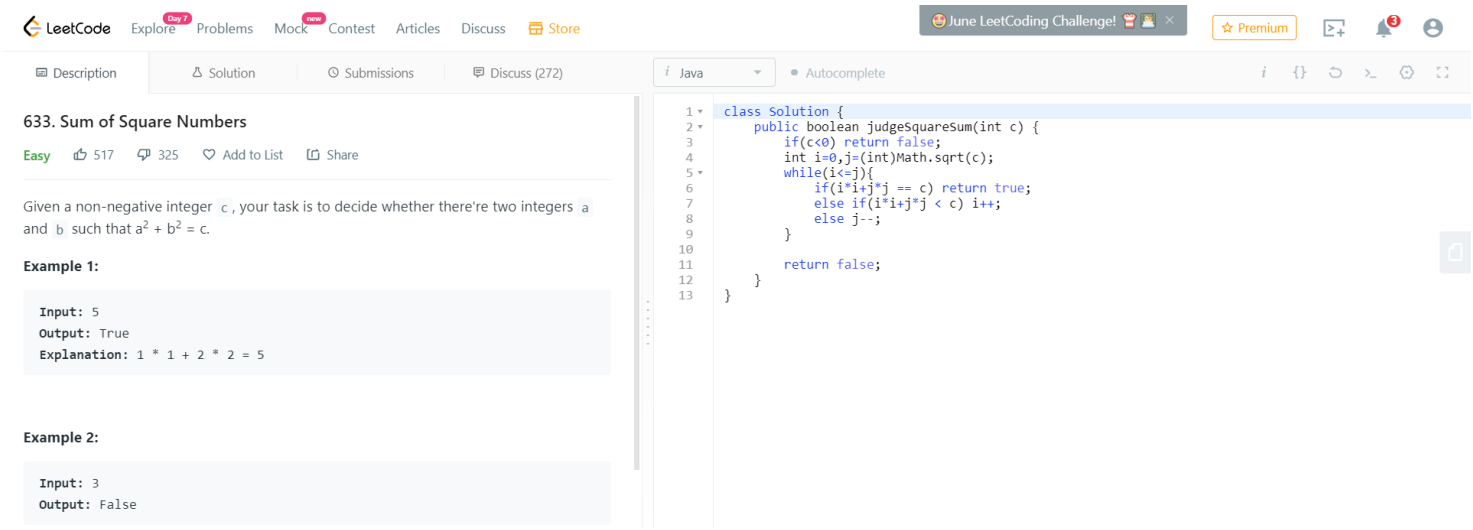


   TC: O(n)



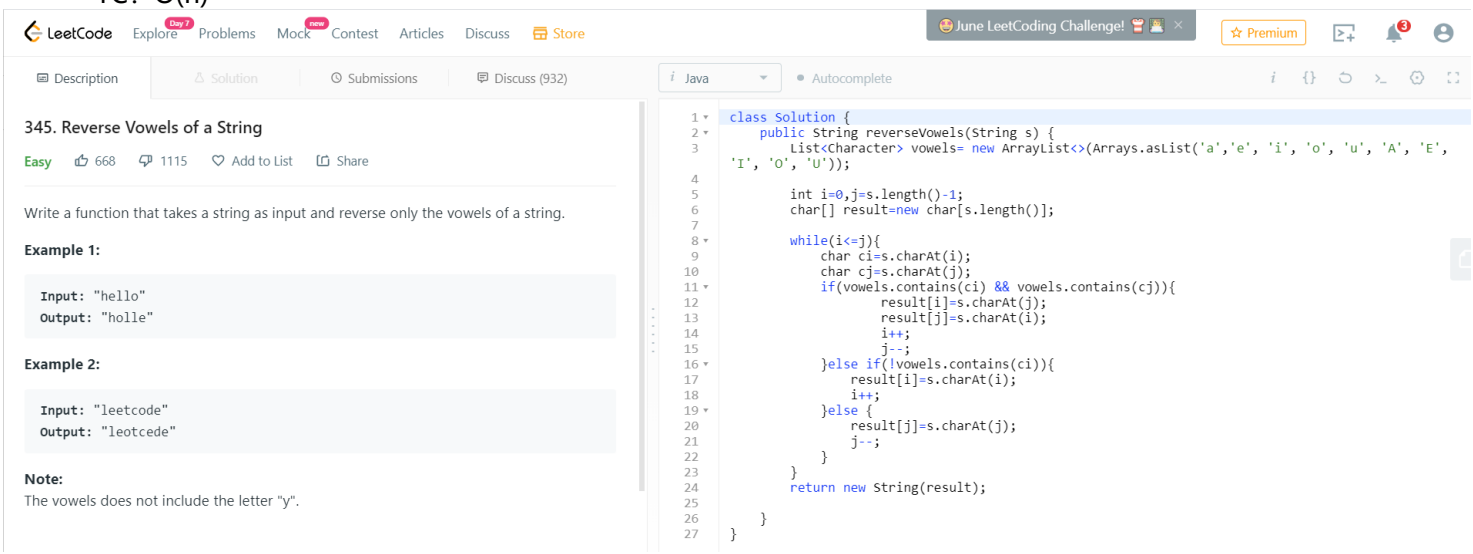
2. sum of square numbers

   https://leetcode.com/problems/sum-of-square-numbers/
   See this as finding numbers in a sorted array from 0 to target. It turns to be same question as the first one.
   Optimize the length of the array. That is an array from 0 to sqrt(c).
   TC: O(n)



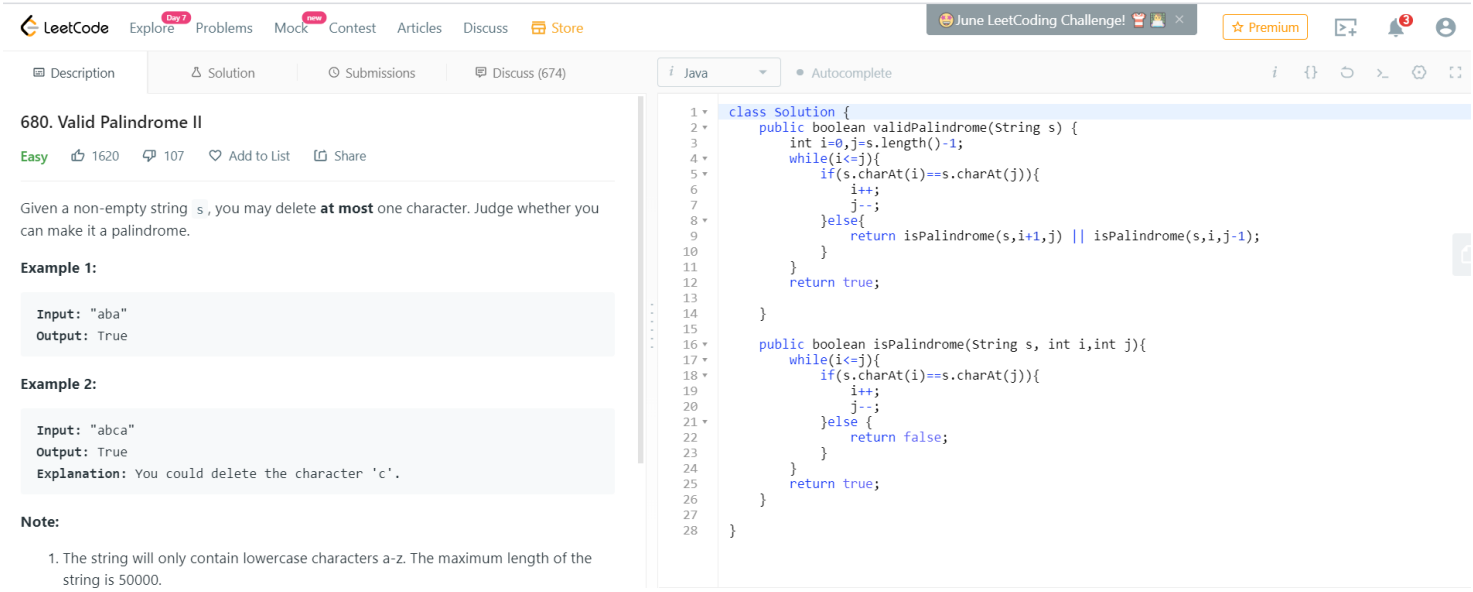
3. reverse vowels of a string

https://leetcode.com/problems/reverse-vowels-of-a-string/
List is an interface and can not be instantiation directly.
I can not exchange characters inside a string. So I set a new char[] to store the result.
TC：O(n)



4. valid palindroma

https://leetcode.com/problems/valid-palindrome-ii/description/
First have a function to determine that s is a palinfroma or not
one point from start one point from the end, if he character is the same, move two points.
if different,move one pointer and use the function to see if substring a palindroma or not



680. Valid Palindrome II

Easy 1620 107 Add to List Share

Given a non-empty string `s`, you may delete **at most** one character. Judge whether you can make it a palindrome.

Example 1:

```
Input: "aba"
Output: True
```

Example 2:

```
Input: "abca"
Output: True
Explanation: You could delete the character 'c'.
```

Note:

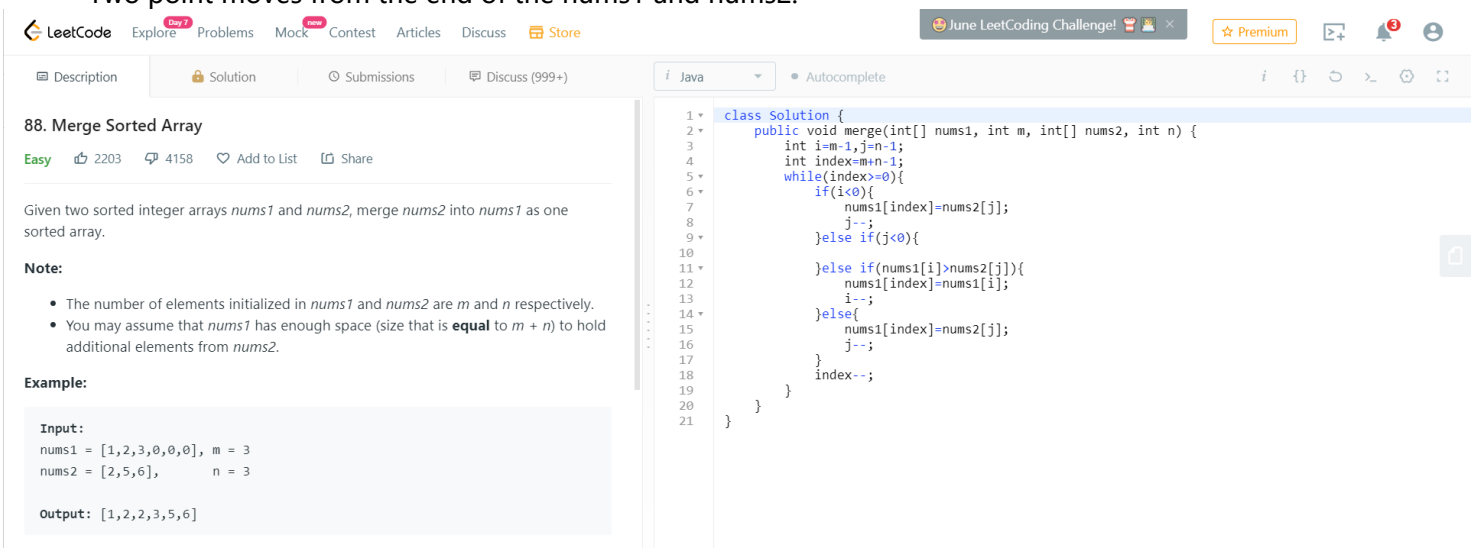
1. The string will only contain lowercase characters a-z. The maximum length of the string is 50000.

```java
class Solution {
    public boolean validPalindrome(String s) {
        int i=0,j=s.length()-1;
        while(i<=j){
            if(s.charAt(i)==s.charAt(j)){
                i++;
                j--;
            }else{
                return isPalindrome(s,i+1,j) || isPalindrome(s,i,j-1);
            }
        }
        return true;

    }

    public boolean isPalindrome(String s, int i,int j){
        while(i<=j){
            if(s.charAt(i)==s.charAt(j)){
                i++;
                j--;
            }else {
                return false;
            }
        }
        return true;
    }
}
```

5. merge sorted array

https://leetcode.com/problems/merge-sorted-array/

Two point moves from the end of the nums1 and nums2.



88. Merge Sorted Array

Easy 2203 4158 Add to List Share

Given two sorted integer arrays *nums1* and *nums2*, merge *nums2* into *nums1* as one sorted array.

Note:

- The number of elements initialized in *nums1* and *nums2* are *m* and *n* respectively.
- You may assume that *nums1* has enough space (size that is **equal** to *m* + *n*) to hold additional elements from *nums2*.

Example:

```
Input:
nums1 = [1,2,3,0,0,0], m = 3
nums2 = [2,5,6],       n = 3

Output: [1,2,2,3,5,6]
```

```java
class Solution {
    public void merge(int[] nums1, int m, int[] nums2, int n) {
        int i=m-1,j=n-1;
        int index=m+n-1;
        while(index>=0){
            if(i<0){
                nums1[index]=nums2[j];
                j--;
            }else if(j<0){

            }else if(nums1[i]>nums2[j]){
                nums1[index]=nums1[i];
                i--;
            }else{
                nums1[index]=nums2[j];
                j--;
            }
            index--;
        }
    }
}
```

6. linked list cycle

https://leetcode.com/problems/linked-list-cycle/

To find a ring, we can use a map to store node, and if the next node has been shown in the map before, we can see there is a ring.

But this cost space complexity.

So we use two pointer, one is slower and one is faster. One pointer move one step each time and the other one move two steps each time.

If there is a ring, two point will meet.

LeetCode Explore Problems Mock Contest Articles Discuss Store — June LeetCoding Challenge! — Premium

Description | Solution | Submissions | Discuss (999+)

141. Linked List Cycle

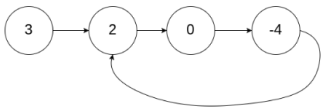
Easy 2861 478 Add to List Share

Given a linked list, determine if it has a cycle in it.

To represent a cycle in the given linked list, we use an integer `pos` which represents the position (0-indexed) in the linked list where tail connects to. If `pos` is `-1`, then there is no cycle in the linked list.

Example 1:

```
Input: head = [3,2,0,-4], pos = 1
Output: true
Explanation: There is a cycle in the linked list, where tail connects to the second node.
```



Java | Autocomplete

```java
/**
 * Definition for singly-linked list.
 * class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode(int x) {
 *         val = x;
 *         next = null;
 *     }
 * }
 */
public class Solution {
    public boolean hasCycle(ListNode head) {
        if(head == null) return false;
        ListNode l1=head, l2=head.next;
        while(l1!=null && l2 !=null && l2.next!=null){
            if(l1==l2){
                return true;
            }else{
                l1=l1.next;
                l2=l2.next.next;
            }
        }
        return false;
    }
}
```

Your previous code was restored from your local storage. Reset to default

7. Longest word in dictionary through deleting

https://leetcode.com/problems/longest-word-in-dictionary-through-deleting/

Use two points to determin if it is a substring

compareTo(): lexicographical order

TC:O(n)

LeetCode Explore Problems Mock Contest Articles Discuss Store — June LeetCoding Challenge! — Premium

Description | Solution | Submissions | Discuss (297)

524. Longest Word in Dictionary through Deleting

Medium 539 233 Add to List Share

Given a string and a string dictionary, find the longest string in the dictionary that can be formed by deleting some characters of the given string. If there are more than one possible results, return the longest word with the smallest lexicographical order. If there is no possible result, return the empty string.

Example 1:

```
Input:
s = "abpcplea", d = ["ale","apple","monkey","plea"]

Output:
"apple"
```

Example 2:

```
Input:
s = "abpcplea", d = ["a","b","c"]

Output:
"a"
```

Java | Autocomplete

```java
class Solution {
    public String findLongestWord(String s, List<String> d) {
        String longestString="";
        for(String target : d){
            int l1=longestString.length(),l2=target.length();
            if(isSubString(s,target)){
                if(l1<l2 || (l1==l2 && target.compareTo(longestString)<0)){
                    longestString=target;
                }
            }else
                continue;
        }
        return longestString;
    }

    public boolean isSubString(String s, String target){
        int i=0,j=0;
        while(i<s.length() && j<target.length()){
            if(s.charAt(i)==target.charAt(j)){
                i++;
                j++;
            }else{
                i++;
            }
        }
        return j==target.length();
    }
}
```

Your previous code was restored from your local storage. Reset to default