Description | Editorial | Solutions (3.6K) | Submissions

228. Summary Ranges

Easy | 3.2K | 1.7K

Companies

You are given a **sorted unique** integer array `nums`.

A **range** `[a,b]` is the set of all integers from `a` to `b` (inclusive).

Return *the **smallest sorted** list of ranges that **cover all the numbers in the array exactly***. That is, each element of `nums` is covered by exactly one of the ranges, and there is no integer `x` such that `x` is in one of the ranges but not in `nums`.

Each range `[a,b]` in the list should be output as:

- `"a->b"` if `a != b`
- `"a"` if `a == b`

Example 1:

```
Input: nums = [0,1,2,4,5,7]
Output: ["0->2","4->5","7"]
Explanation: The ranges are:
[0,2] --> "0->2"
[4,5] --> "4->5"
[7,7] --> "7"
```

Example 2:

```
Input: nums = [0,2,3,4,6,8,9]
Output: ["0","2->4","6","8->9"]
Explanation: The ranges are:
[0,0] --> "0"
[2,4] --> "2->4"
[6,6] --> "6"
```

Java | Auto

```java
class Solution {
    public List<String> summaryRanges(int[] nums) {
      List<String> list  = new ArrayList();
      int n = nums.length;
      for(int i=0; i<n; i++) {
          int start = nums[i];
          while(i + 1 < n && nums[i + 1] == nums[i]+1)
          i++;
          if(start != nums[i])
            list.add("" + start + "->" + nums[i]);
            else
                list.add("" + start);
      }
      return list;
    }
}
```

Testcase | Result

Accepted Runtime: 10 ms

Case 1 | Case 2

Input

nums =
[0,1,2,4,5,7]

Output

["0->2","4->5","7"]

Expected

["0->2","4->5","7"]

Console | Run | Submit

Description | Editorial | Solutions (10K) | Submissions

125. Valid Palindrome

Easy 7K 7.2K

Companies

A phrase is a **palindrome** if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers.

Given a string `s`, return `true` *if it is a* ***palindrome****, or* `false` *otherwise.*

Example 1:

```
Input: s = "A man, a plan, a canal: Panama"
Output: true
Explanation: "amanaplanacanalpanama" is a palindrome.
```

Example 2:

```
Input: s = "race a car"
Output: false
Explanation: "raceacar" is not a palindrome.
```

Example 3:

```
Input: s = " "
Output: true
Explanation: s is an empty string "" after removing non-alphanumeric characters.
Since an empty string reads the same forward and backward, it is a palindrome.
```

Java | Auto

```java
class Solution {
    public boolean isPalindrome(String s) {
        if (s.isEmpty()) {
            return true;
        }
        int start = 0;
        int last = s.length() - 1;
        while(start <= last) {
            char currFirst = s.charAt(start);
            char currLast = s.charAt(last);
            if (!Character.isLetterOrDigit(currFirst )) {
                start++;
            } else if(!Character.isLetterOrDigit(currLast)) {
                last--;
            } else {
                if (Character.toLowerCase(currFirst) != Character.toLowerCase(currLast)) {
                    return false;
                }
```

Testcase | Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

s =
"A man, a plan, a canal: Panama"

Output

true

Expected

true

Console | Run | Submit