21. Merge Two Sorted Lists

In this question, we have two sorted list, and we want to merge them to one long list.

The first step is checking the null pointer

What I would do is using two pointers to loop through the two sorted lists separately, and then

When both pointers are not null, we can add them

20. Valid Parenthese

use stack to store the character

小括号 open bracket

方括号 square bracket

花括号 curly bracket

53. Maximum Subarray

we need to find the maximum subarray, we can use a

because we want to find a optimal value, this value is must be the

**811. Subdomain Visit Count**

In this question, the dot operator is the separation

202. Happy Number

14. Longest Common Prefix

Because we want to find the common prefix, so we must loop through all strings to check and the longest one must appear in every string

**387. First Unique Character in a String**

because we need to find the first

non-repeating should loop through all element

store the index for later use

**134. Gas Station**

**Because there is a circular route, we will use a while loop to traverse all station**

**392. Is Subsequence**

**Order is important**

**Use a queue to store the subsequence order**

**缩小范围英语怎么说?**

**213. House Robber II**

**We can reuse the function we wrote for the House Robber I**

**In the former question, whether to rob num[lo] is determined by ourself, but now it is constrained by whether num[hi] is robbed**

**69. Sqrt(x)**

**使用二分查找来不断缩小取值范围，逼近最终结果**

**915. Partition Array into Disjoint Intervals**

**The key step is to find the partition location to separate the array into two parts**

**使用两个指针分别指向最左边 和 最右边的元素 比较大小 left 这边一定小于right这边 否则不存在答案**

**nums[left] < nums[right]**

**因为要求的是Left subarray最小 所以需要将right指针左移，继续比较**

**终止条件为 nums[left] > nums[right] 或 left > right 指针越界**

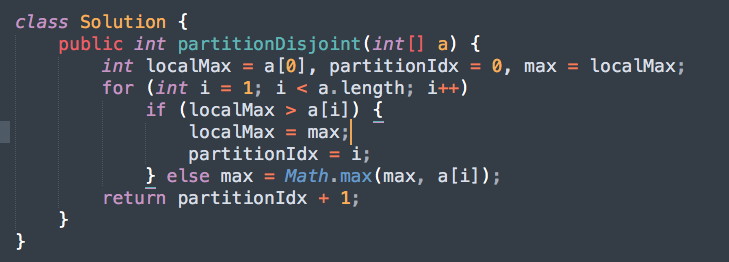
**这时需要考虑将左指针右移，因为我们需要将**

**其实不就是直接将最右边的指针不断左移 直到找到第一个比nums[0]小的元素？ 可以存在相等的情况**

**应该是找到最左边的第一个分割点 we have to incorporate a[i] to form the left subarray.**

**while (nums[left] < nums[right])**

**from the left side and from the right side, we compare the value of the pointers point**

****

**解释:**

suppose the original left subarray is from 0 to partitionIdx, the max value of that is localMax. If it is a valid partition, every value from partitionIdx + 1 to end should be >= localMax. But if we find a value in the right part, a[i], is smaller than localMax, which means the partition is not correct and we have to incorporate a[i] to form the left subarray. So the partitionIdx is set to be i, and we have to recalculate the max value of the new left subarray. (recorded in max)