算法讨论班第二十三期——郭清沛

2016年2月26日

1. Find Minimum in Rotated Sorted Array

Suppose a sorted array is rotated at some pivot unknown to you beforehand.

(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2).

Find the minimum element.

You may assume no duplicate exists in the array.

思路：

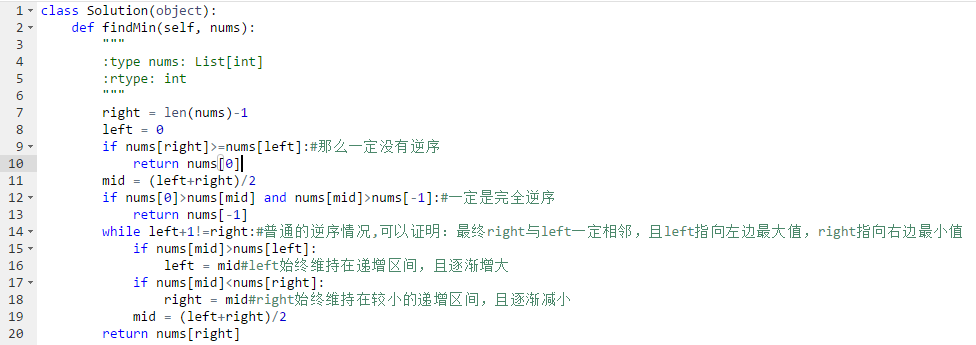
逆序之后的数组一共只有3种情况，分别为：

1. 没有逆序：单调递增的数组
2. 从中间逆序：这是我们需要考虑的一般情况
3. 完全逆序：单调递减的数组

对于一般情况我们单独讨论:

显然题目让我们找最小，直接遍历一遍，一定可以在O(n)的时间内解决，但这样就没有利用到已知条件：这个数组是部分有序的，因此猜想一定有一种O(logn)的算法——想到二分查找。Left,mid,right ,其中 mid = (left+right) /2

Left一直在左边的递增区间并逐渐向最大值方向移动，right维持在右边的递增区间并逐渐向最小值方向移动，那么可以证明最终 right一定等于 left+1,left指向最大值，right指向最小值，返回right



拓展：

回顾我们曾经做过的题：

33. Search in Rotated Sorted Array：

Suppose a sorted array is rotated at some pivot unknown to you beforehand.

(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2).

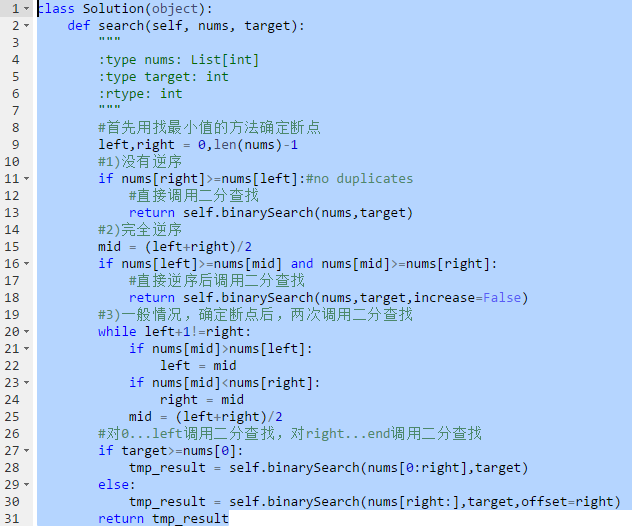
You are given a target value to search. If found in the array return its index, otherwise return -1.

You may assume no duplicate exists in the array.

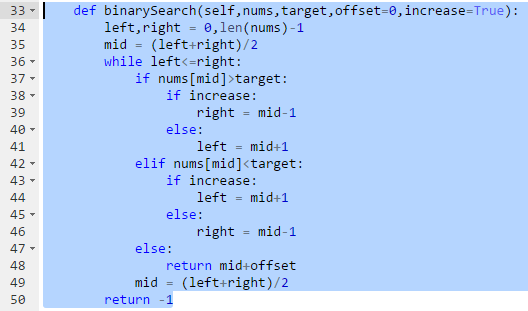
参考以上的思路：

1. 利用上一题的结论，找到最小值或者称之为”拐点”
2. 在“拐点”两侧分别进行二分查找

程序主体：



调用一般性的二分查找:



2. Sum Root to Leaf Numbers

Given a binary tree containing digits from 0-9 only, each root-to-leaf path could represent a number.

An example is the root-to-leaf path 1->2->3 which represents the number 123.

Find the total sum of all root-to-leaf numbers.

For example,

1

/ \

2 3

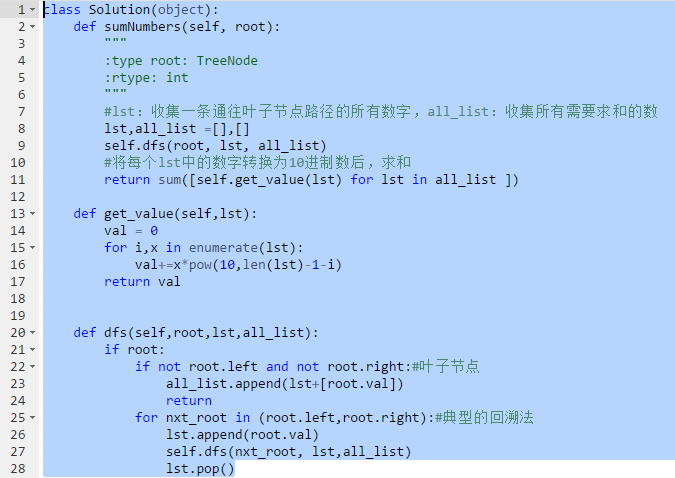
The root-to-leaf path 1->2 represents the number 12.

The root-to-leaf path 1->3 represents the number 13.

Return the sum = 12 + 13 = 25.

解题思路：

思路很直观：dfs遍历整棵树，到树的叶子节点(左右子树都为空)时，将这个数保存起来。



3. Palindrome Partitioning

Given a string s, partition s such that every substring of the partition is a palindrome.

Return all possible palindrome partitioning of s.

For example, given s = "aab",

Return

[

["aa","b"],

["a","a","b"]

]

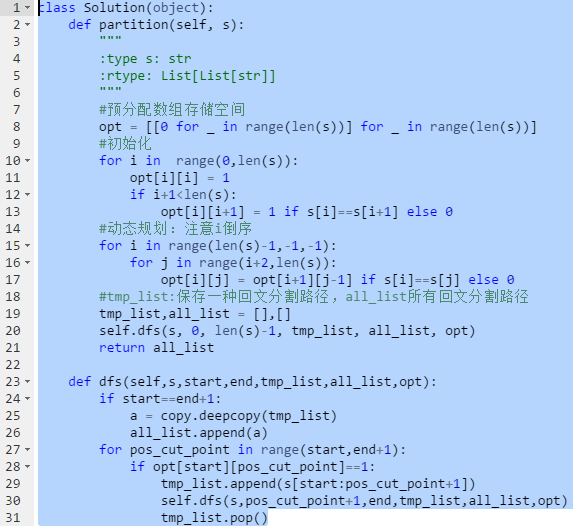
思路：

经典问题：回文串问题

动态规划：

OPT[i][j] = OPT[i+1][j-1] && A[i]==A[j]

初始化:OPT[i][i] =1 ,OPT[i][i+1]=(A[i]==A[i+1])



4. Flatten Binary Tree to Linked List

Given a binary tree, flatten it to a linked list in-place.

For example,

Given

1

/ \

2 5

/ \ \

3 4 6

The flattened tree should look like:

1

\

2

\

3

\

4

\

5

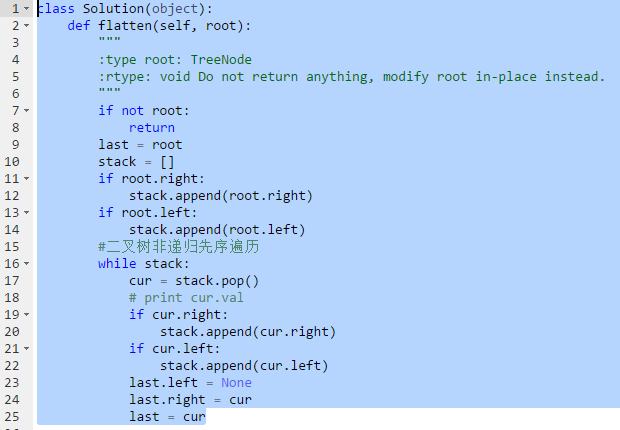
\

6

思路：

基本题：考察二叉树的先序遍历

观察发现flattened tree是按先序遍历节点的顺序首尾相接，因此先序遍历时保存上一个节点，当遍历到当前节点时，将上一个节点的指针指向当前节点即可。



5.Convert Sorted List to Binary Search Tree

Given a singly linked list where elements are sorted in ascending order, convert it to a height balanced BST.

思路：

回顾数据结构的基础知识：

BST的概念：

1. 是二叉树
2. 1.若左子树不空,则左子树上所有结点的值均小于它的根结点的值；

2.若右子树不空,则右子树上所有结点的值均大于它的根结点的值；

3.左、右子树也分别为二叉排序树

性质：中序遍历是一个递增的序列

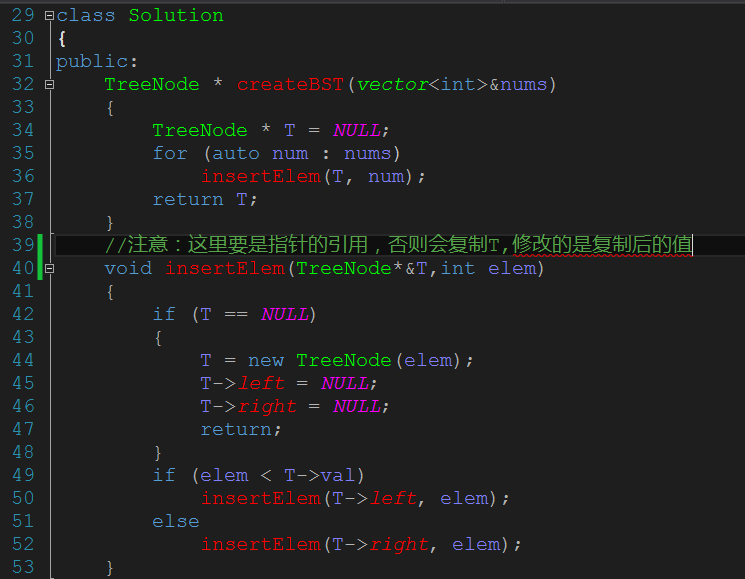
平衡树的概念：

1. 左右子树的深度差绝对值小于等于1
2. 每棵子树都是平衡树

为什么需要引入平衡树？

* 树的查找所花的时间与树的高度h 成比例, 并不与树的容量 n 成比例
* 如果可以让树维持矮矮胖胖的好身材, 也就是让h维持在O(lg n)左右, 完成上述工作就很省时间。能够一直维持好身材, 不因新增删除而长歪的搜寻树, 叫做balanced search tree（平衡树）。

1. 首先回顾本科数据结构的练习题：给一个数组(无重复的数)，如何建一个BST



思考为什么这里用C++实现？？而不用python实现?

1. 具体这题应该怎么做呢？

对于一个已经排序的数组，如果我们希望构成的二叉排序树是平衡的，那么就要让其左右子树的节点数量尽可能接近，因此取每次取中值作为平衡树的根节点。

具体算法如下：

