Lintcode： Max Tree

给出一个没有重复的整数数组，在此数组上建立最大树的定义如下：

* 根是数组中最大的数
* 左子树和右子树元素分别是被父节点元素切分开的子数组中的最大值
* **样例**
* 给出数组 [2, 5, 6, 0, 3, 1]，构造的最大树如下：
* 6
* / \
* 5 3
* / / \
* 2 0 1

思路：單調棧。維持一個遞減棧。

當前值例如4，在前面第一個比當前大的數例如5,2肯定是3的右邊直接或者間接的孩子

而4和5中間的比4小的數

public class Solution {

13 /\*\*

14 \* @param A: Given an integer array with no duplicates.

15 \* @return: The root of max tree.

16 \*/

17 public TreeNode maxTree(int[] A) {

18 if (A.length==0) return null;

19

20 Stack<TreeNode> nodeStack = new Stack<TreeNode>();

21 nodeStack.push(new TreeNode(A[0]));

22 for (int i=1;i<A.length;i++)

23 if (A[i]<=nodeStack.peek().val){

24 TreeNode node = new TreeNode(A[i]);

25 nodeStack.push(node);

26 } else {

27 TreeNode n1 = nodeStack.pop();

28 while (!nodeStack.isEmpty() && nodeStack.peek().val < A[i]){

29 TreeNode n2 = nodeStack.pop();

30 n2.right = n1;

31 n1 = n2;

32 }

33 TreeNode node = new TreeNode(A[i]);

34 node.left = n1;

35 nodeStack.push(node);

36 }

37

38

39 TreeNode root = nodeStack.pop();

40 while (!nodeStack.isEmpty()){

41 nodeStack.peek().right = root;

42 root = nodeStack.pop();

43 }

44

45 return root;

46

47

48

49

50

51 }

52 }