Class 24 加强练习 3 (Recursion III)

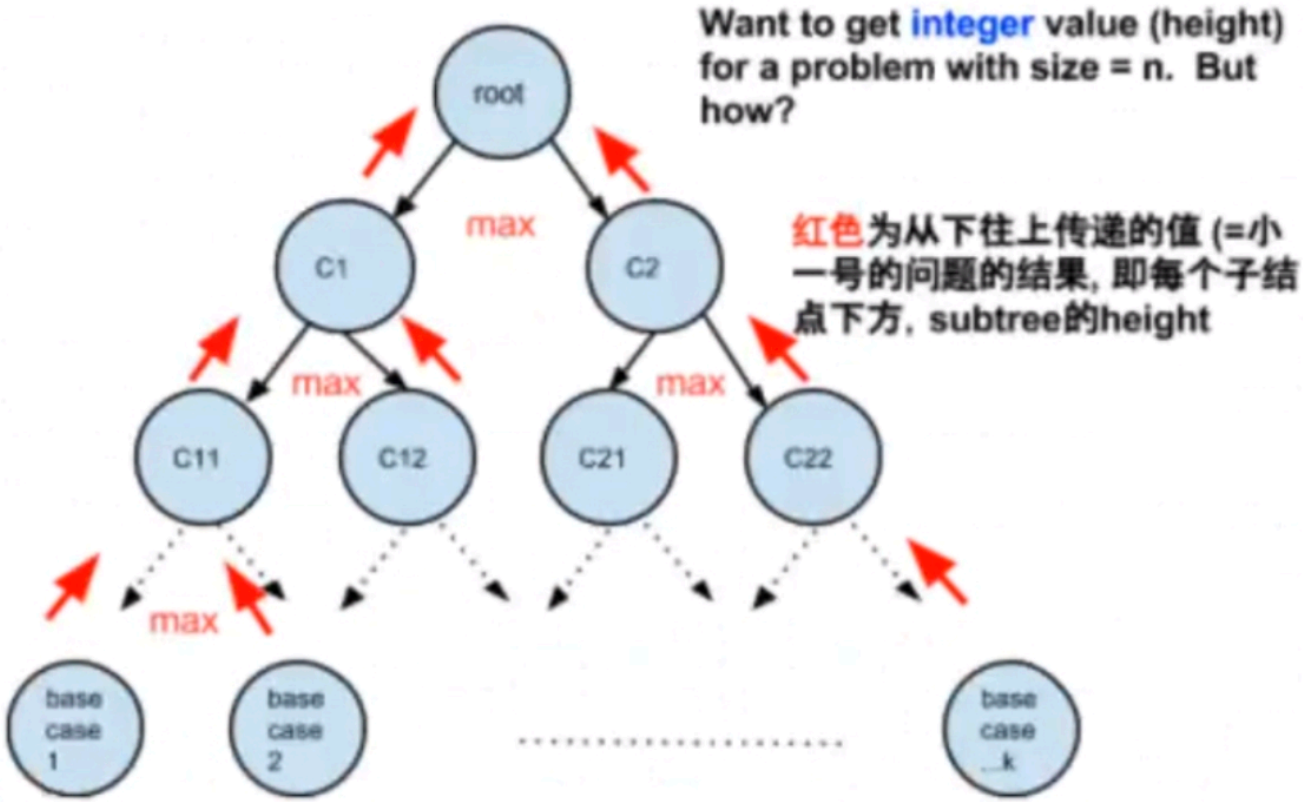
Q1. Tree + Recursion 第一类问题：

Use recursion to return values in a bottom-up way in binary tree

Q1.1 Determine whether a binary tree is a balanced binary tree **(O(nlogn) solution).**

What's the definition of "balanced"? It could be:

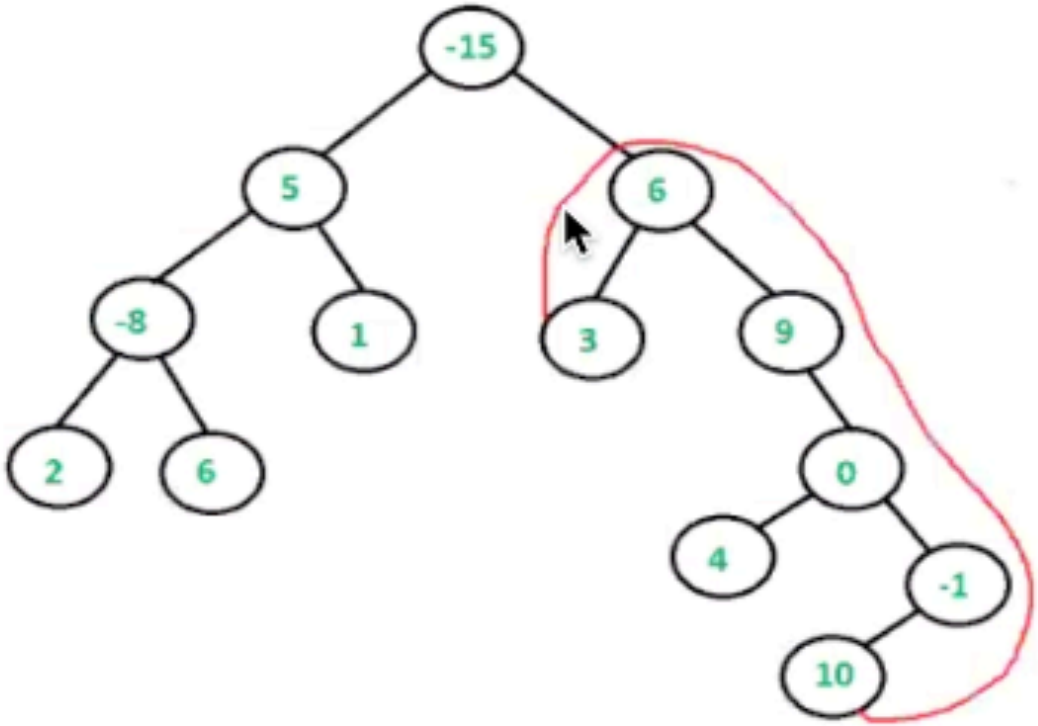
- the tree has a minimum possible overall height
- no leaf is too further away, i.e. 0 or 1, from root than any other leaf
- **left and right sub-trees have similar height, i.e. difference is 0 or 1 (balanced height)**



Q1.3 Midterm 2 question 2 (重复强调，简要复习)

Given a binary tree in which each node element contains a number. Find the maximum possible sum **from one leaf node to another**.

The maximum sum path may or may not go through root. For example, in the following binary tree, the maximum sum is 27(3 + 6 + 9 + 0 – 1 + 10). Expected time complexity is O(n).



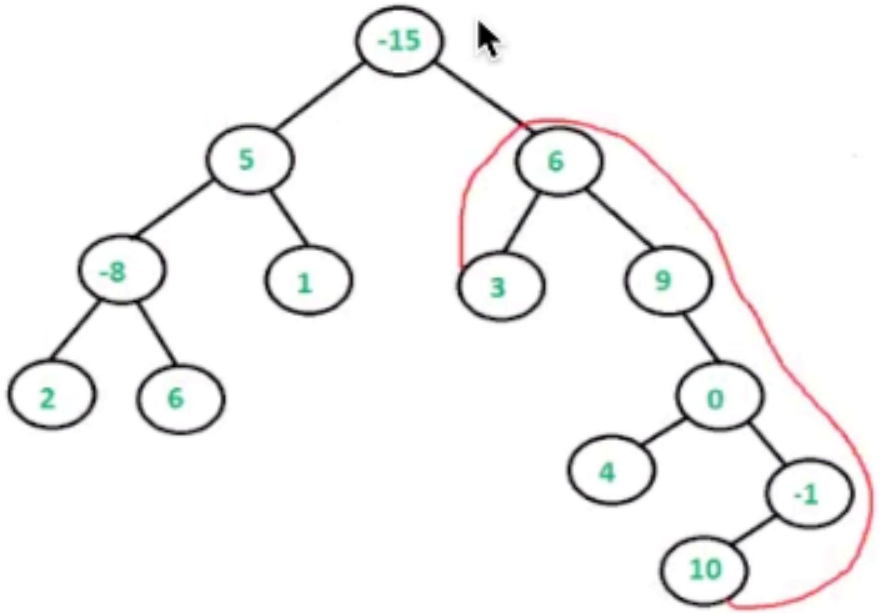
Way of thinking (Tricks)

1. **What do you expect from your lchild / rchild?**
   - Max single path in my left subtree **(1)**
   - Max single path in my right subtree **(2)**
2. What do you want to do in the current layer?
   - update global_max = left + right + root.value if feasible
3. **What do you want to report to your parent? (same as Q1 == Q3)**
   - it is usually the return type of the Recursion function

Q1.4 (人字形path 问题)

Laicode.com Class 20 (Maximum Path Sum Binary Tree II)

Get Maximum sum of the path cost from **any node to any node** (not necessarily from leaf to leaf)



Way of thinking (Tricks)

1. **What do you expect from your lchild / rchild?**

   Max single path in my left subtree **[ended at the left child node] , if this value is negative, we discard it**

   Max single path in my right subtree **[ended at the right child node] , if this value is negative, we discard it**

2. What do you want to do in the current layer?

   update global_max = left + right + root.value if feasible

3. **What do you want to report to your parent? (same as Q1 == Q3)**

   it is usually the return type of the Recursion function

Q2. Tree + Recursion 第二类问题： (Path Problem in binary tree)

Discussion:

Note that: Tree 相关问题，路径种类可以分为两大类

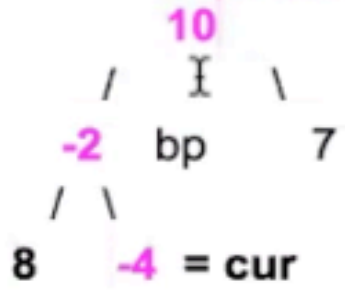
Class 1: 人字形path，这类题一般需要从下往上传integer value (E.g.,**Q1.1 - 1.4 above**)

Class 2：从root 往下 （**直上直下**）path

Key point: carry a **直上直下** path prefix (非人字形) while traversing the tree:

a. complete path from leaf to root

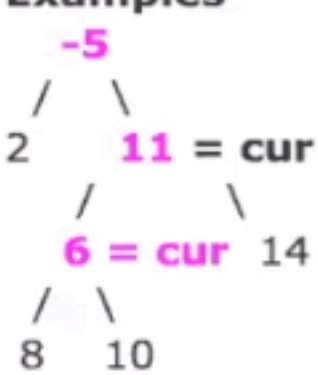
b. sub-section of complete path from leaf to root



Prefix_of_path = {10, -2, -4}

Q2.2 (laicode.com Class 20) Given a binary tree in which each node contains an integer number. Determine if there exists a path from any node to any node (the two nodes can be the same node and they can only be on the path from root to one of the leaf nodes), the sum of the numbers on the path **is equal to the given target number.**

Examples



If target = 17, There exists a path 11 + 6, the sum of the path is target,
If target = 100, There does not exist any paths sum of which is target.

Solution 1:

path_prefix = {-5 11 6}

← cur

6

11 + 6

(-5) + 11 + 6

Time = O(n ^ 2)

Pre-order to iterate the whole tree, and for each current node X, we do a for loop in {X .... root}

Solution 2:

Pre-order to iterate the whole tree, and for each current node X, we do a for loop in {X …. root}.

path_prefix = {-5 11 6}

11+6=17

root cur

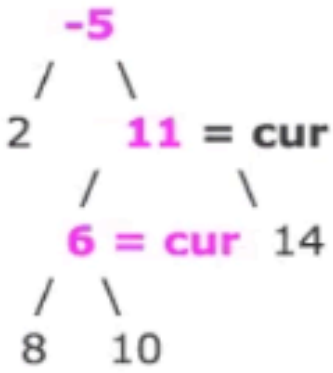
path_prefix = xxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxx

another_path_prefix target

another_path_prefix + target == path_prefix_to_current

another_path_prefix == path_prefix_to_current - target

12 17 = -5



path_prefix = {-5, 11, 6}

HashSet<path_prefix_sum> = {-5 6 12}

12 - target (17) = -5

We use a HashSet to store all path_prefix_sum.

Time = O(n)

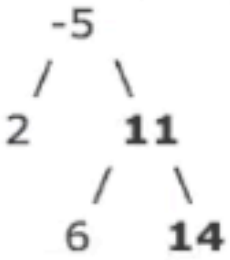
Q2.3 (laicode.com Class 20) **Maximum Path Sum Binary Tree III**

Given a binary tree in which each node contains an integer number. Find the maximum possible sum **from any node to any node (the two nodes can be the same node and they can only be on the path from root to one of the leaf nodes).**

Assumptions

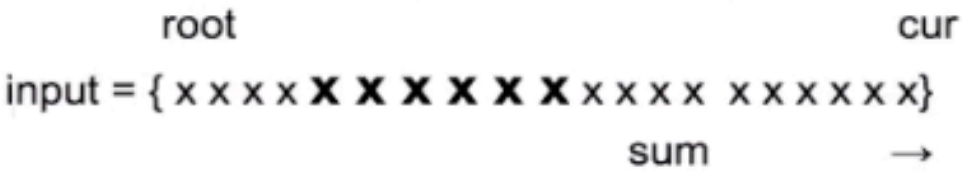
- The root of given binary tree is not null

Examples



The maximum path sum is 11 + 14 = 25

Solution 1 (DP max subarray sum) :



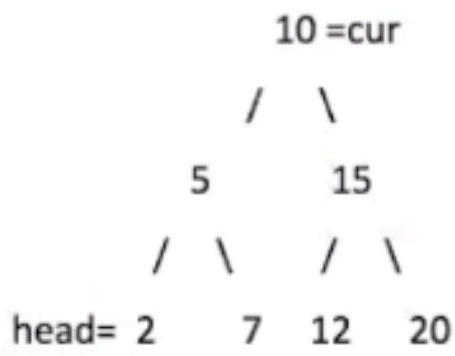

```
public void helper(TreeNode root, int[] max, int sum) {
        // base case
        if (root == null) {
                return;
        }

        if (sum < 0) {
                sum = root.value;
        } else {
                sum += root.value;
        }
        max[0] = Math.max(max[0], sum);  // this is actually a pre-order traversal. that's it!

        helper(root.left, max, sum);
        helper(root.right, max, sum);
}
```

Q3. Tree + Recursion 第三类问题： Tree Serialization Problem

Q3.1. Given a Binary Tree, convert it to a Doubly Linked List(DLL) in in-order sequence.

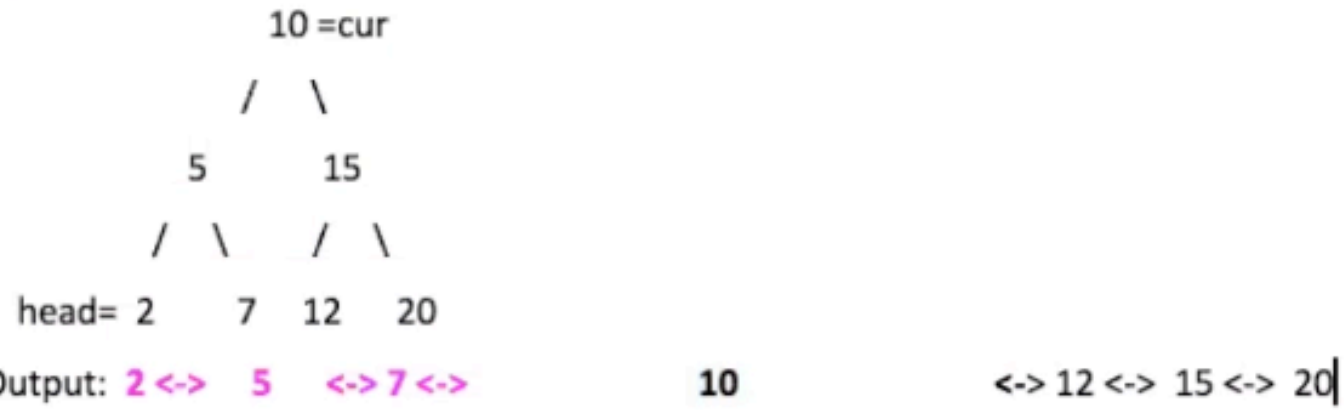


Output: 2 <-> 5 <-> 7 <-> 10 <-> 12 <-> 15 <-> 20

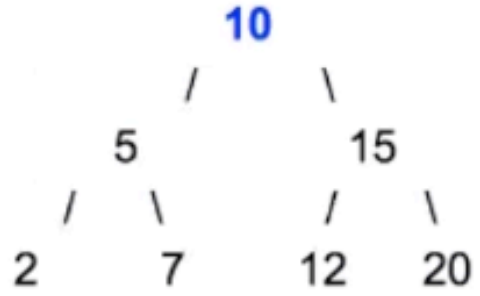
Discussion:

Reconstruct a tree by using xxx-order and **in-order** traversal sequences .

此类问题的要点就是把global 的问题一分为二(recursively)，每半边返回一个subtree的root node.



Q4.1 How to reconstruct a tree **(with no duplicate values)** with **pre-order** and **in-order** sequences of all nodes.



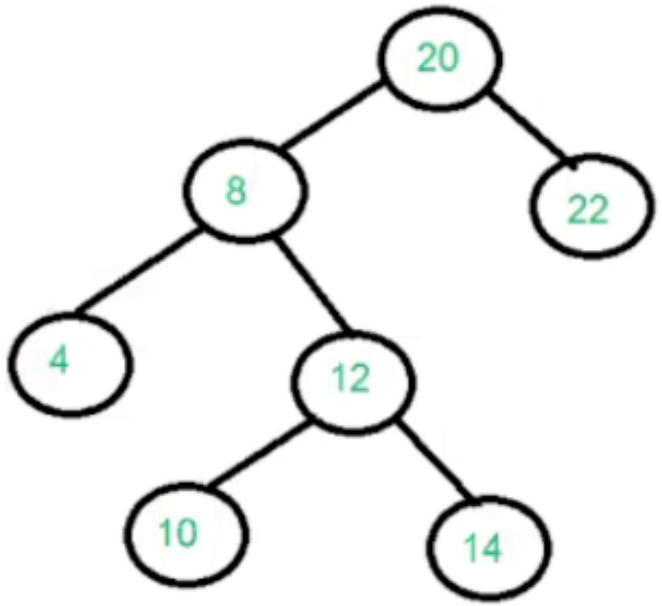
Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Preorder:	**10**	5	2	7	15	12	20
Inorder:	2	5	7	**10**	12	15	20

inLeft == 0 | idxMap[10] == 3 | inRight

idxMap[10] - inLeft = 3 - 0 = 3 == leftSize

Q4.3 Construct a tree from **Inorder** and **Level** order traversals of **binary tree**

Given the inorder and level-order traversal sequences of a Binary Tree **(you can assume all unique numbers in the tree)**, how to re-construct the Binary Tree.



Input: Two arrays that represent Inorder and level order traversals of a Binary Tree

index= 0 1 2 3 4 5 6

in-order[] = {4, 8, 10, 12, 14, 20, 22};

level-order[] = {20, 8, 22, 4, 12, 10, 14};