

LintCode领扣题解 (/problem) / 最大子树 · Maximum Subtree

最大子树 · Maximum Subtree

中文

亚马逊 (/problem/?tags=amazon)

Depth-first Search (/problem/?tags=depth-first-search)

二叉树 (/problem/?tags=binary-tree)

描述

给你一棵二叉树，找二叉树中的一棵子树，他的所有节点之和最大。
返回这棵子树的根节点。

📌 LintCode 会把你返回的节点作为最优子树来打印。 数据保证有且仅有唯一的解。

样例

样例 1:

输入：
{1,-5,2,0,3,-4,-5}
输出:3
说明
这棵树如下所示：
1
/ \
-5 2
/ \ / \
0 3 -4 -5
以3为根的子树（只有3一个节点）的和是最大的，所以返回3。

样例 2:

输入：
{1}
输出:1
说明：
这棵树如下所示：
1
这棵树只有整体这一个子树，所以返回1。

在线评测地址: <https://www.lintcode.com/problem/maximum-subtree/> (<https://www.lintcode.com/problem/maximum-subtree/>)

收起题目描述 ^

语言类型


ALL (9)

python (4)

cpp (3)

java (2)

上传题解

 DDBear
更新于 11/21/2020, 5:20:01 AM

解题思路

二叉树是一个递归结构，可以根据两个子树的信息来维护当前子树的信息，对于某个节点来说，以该节点为根的子树和等于当前节点的权值加上左子树和右子树的权值。这可以通过自底向上的后序遍历来实现。

算法：二叉树的遍历

维护子树和：

1. 当根为 `nullptr` 时，返回 `0`，即为递归出口。
2. 获取左子树的权值和。
3. 获取右子树的权值和。
4. 更新当前的权值和为左子树加右子树加当前节点的值并返回。

复杂度分析

设树的节点数为 n 。

- 时间复杂度 $O(n)$
 - 每个节点会被访问 1 次，时间复杂度为 $O(n)$ 。
- 空间复杂度 $O(n)$
 - 在树上递归的遍历，最多会在栈上占用 $O(n)$ 的空间。

源代码

[java](#)[c++](#)[python](#)

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
"""
Definition of TreeNode:
class TreeNode:
    def __init__(self, val):
        self.val = val
        self.left, self.right = None, None
"""
import sys
class Solution:
    """
    @param root: the root of binary tree
    @return: the maximum weight node
    """
    def findSubtree(self, root):
        sum, max_sum, max_subtree = self.get_sum(root)
        return max_subtree

# 分别返回当前子树和, 子树中最小的和, 目标节点
def get_sum(self, root):
    # 递归出口
    if root is None:
        return -sys.maxsize, -sys.maxsize, None
    # 获得左右子树的和
    left_sum, left_max_sum, left_max_subtree = self.get_sum(root.left)
    right_sum, right_max_sum, right_max_subtree = self.get_sum(root.right)
    max_sum = sys.maxsize
    # 计算当前子树和, 并更新答案
    sum = root.val
    if root.left is not None:
        sum += left_sum
    if root.right is not None:
        sum += right_sum
    max_sum = max(left_max_sum, right_max_sum, sum)

    if sum == max_sum:
        return sum, max_sum, root
    if left_max_sum == max_sum:
        return sum, max_sum, left_max_subtree
    return sum, max_sum, right_max_subtree
```

👍 获赞 3

💬 2 条评论

你的口袋题库

2000+ 算法真题、国内外名企题库免费开放



九章算法APP

令狐冲

更新于 10/17/2020, 2:37:35 AM

以每个点为根节点root (即每一个子树), 找到左子树的节点x和与右子树的节点和y。如果 $root.val + x.val + y.val >$ 当前已知最大值, 就更新答案。

👍 获赞 1 💬 1 条评论

Page 4 of 10

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    # @param {TreeNode} root the root of binary tree
    # @return {int} the maximum weight node
    import sys
    maximum_weight = 0
    result = None

    def findSubtree(self, root):
        # Write your code here
        self.helper(root)

        return self.result

    def helper(self, root):
        if root is None:
            return 0

        left_weight = self.helper(root.left)
        right_weight = self.helper(root.right)

        if left_weight + right_weight + root.val >= self.maximum_weight or self.result is None:
            self.maximum_weight = left_weight + right_weight + root.val
            self.result = root

        return left_weight + right_weight + root.val
```

邀请有礼
vitation/sh:



👍 获赞 1 💬 5 条评论



令狐冲

更新于 6/9/2020, 7:04:30 AM

以每个点为根节点root（即每一个子树），找到左子树的节点x和与右子树的节点和y。如果 $root.val + x.val + y.val >$ 当前已知最大值，就更新答案。

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    /**
     * @param root the root of binary tree
     * @return the maximum weight node
     */
    TreeNode* findSubtree(TreeNode* root) {
        // Write your code here
        helper(root);
        return result;
    }

    int helper(TreeNode* root) {
        if (root == NULL) {
            return 0;
        }
        int left_weight = helper(root->left);
        int right_weight = helper(root->right);

        if (result == nullptr || left_weight + right_weight + root->val > maximum_weight) {
            maximum_weight = left_weight + right_weight + root->val;
            result = root;
        }
        return left_weight + right_weight + root->val;
    }

private:
    TreeNode* result = nullptr;
    int maximum_weight = INT_MIN;
};
```

👍 获赞 0

💬 添加评论



九章用户OPC79M

更新于 6/9/2020, 7:04:22 AM

Record max sum in each traversal, update result if new max found.

```
/**
 * 本参考程序由九章算法用户提供。版权所有，转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作，授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括：九章算法班 2020升级版，算法强化班，算法基础班，北美算法面试高频题班，Java 高级工程师 P6+ 小班课，面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括：系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括：动态规划专题班，Big Data - Spark 项目实战，Django 开发项目课
 * - 更多详情请见官方网站：http://www.jiuzhang.com/?utm_source=code
 */
class Solution:
    def findSubtree(self, root):
        self.max_sum = -sys.maxint
        self.ans = root
        self.dfs(root)
        return self.ans

    def dfs(self, node):
        if not node:
            return 0

        left = self.dfs(node.left)
        right = self.dfs(node.right)
        s = node.val + left + right

        if s > self.max_sum:
            self.max_sum = s
            self.ans = node

        return s
```

👍 获赞 0 💬 添加评论



JeremyXu

更新于 6/9/2020, 7:04:13 AM

补充一个不需要定义全局变量的做法，以防面试官要求pure function

```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ /
Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm\_source=code
 */
"""
Definition of TreeNode:
class TreeNode:
    def __init__(self, val):
        self.val = val
        self.left, self.right = None, None
"""

class Solution:
    """
    @param root: the root of binary tree
    @return: the maximum weight node
    """
    def findSubtree(self, root):
        # write your code here
        return self.findSubtreeHelper(root)[2]

    def findSubtreeHelper(self, root):

        if root is None:

            return 0, 0, None

        left_max_sum, left_sum, left_root = self.findSubtreeHelper(root.left)
        right_max_sum, right_sum, right_root = self.findSubtreeHelper(root.right)

        curt_sum = left_sum + right_sum + root.val
        max_sum = max(left_max_sum, right_max_sum, curt_sum)

        if max_sum == left_max_sum:

            max_root = left_root

        elif max_sum == right_max_sum:

            max_root = right_root

        else:

            max_root = root

        return max_sum, curt_sum, max_root
```

👍 获赞 0

💬 1 条评论



社会我喵哥

更新于 6/9/2020, 7:04:06 AM

不用全局变量也可以, 就得把TreeNode 和 ans 作为参数加到dfs 方法里去了。


```
/**
 * 本参考程序由九章算法用户提供。版权所有, 转发请注明出处。
 * - 九章算法致力于帮助更多中国人找到好的工作, 授课老师均来自硅谷和国内的一线大公司在职工程师。
 * - 现有的求职课程包括: 九章算法班 2020升级版, 算法强化班, 算法基础班, 北美算法面试高频题班, Java 高级工程师 P6+ 小班课, 面试软技能指导 - BQ / Resume / Project 2020版
 * - Design类课程包括: 系统设计 System Design, 面向对象设计 OOD
 * - 专题及项目类课程包括: 动态规划专题班, Big Data - Spark 项目实战, Django 开发项目课
 * - 更多详情请见官方网站: http://www.jiuzhang.com/?utm_source=code
 */
class Solution {
public:
    /**
     * @param root: the root of binary tree
     * @return: the maximum weight node
     */
    int maxSum = INT_MIN;
    TreeNode* ans = nullptr;

    TreeNode* findSubtree(TreeNode* root) {
        dfs(root);
        return ans;
    }
    int dfs(TreeNode* root) {
        if (root == nullptr) return 0;
        int left = dfs(root->left);
        int right = dfs(root->right);
        int sum = left + right + root->val;
        if (ans == nullptr || sum > maxSum) {
            maxSum = sum;
            ans = root;
        }
        return sum;
    }
};
```

👍 获赞 0

💬 添加评论

进阶课程

视频+互动

直播+互动

直播+互动

互动课

九章算法班 2021 版

8周时间精通 57 个核心高频考点, 9 招击破 FLAG、BATJ 算法面试。22...

系统架构设计 System Design 2021 版

成为百万架构师必上。30 课时带你快速掌握 18 大系统架构设计知识点与面...

九章算法面试高频题冲刺班

每期更新 15% 题目, 考前押题, 一举拿下 FLAG & BATJ Offer

面向对象设计 OOD

应届生及亚马逊面试必考, IT 求职必备基础

