



数据结构与算法实验

实验五：二叉树

彭振辉

中山大学人工智能学院

2023年秋季学期



- **熟练掌握二叉树的逻辑结构和存储结构；**
- **熟练掌握二叉树的各种遍历算法。**

实验要求



- **二叉树结构和运算定义，算法的实现以库文件方式实现，不得在测试主程序中直接实现；**
- **程序有较好可读性，各运算和变量的命名直观易懂，符合关键工程要求；**
- **程序有适当的注释。**

实验任务1



- 给定一个二叉树，判断它是否是高度平衡的二叉树（即一个二叉树每个节点的左右两个子树的高度差的绝对值不超过1）。

```
struct TreeNode {  
    int val;  
    TreeNode *left;  
    TreeNode *right;  
    TreeNode() : val(0), left(nullptr), right(nullptr) {}  
    TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}  
    TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}  
};
```

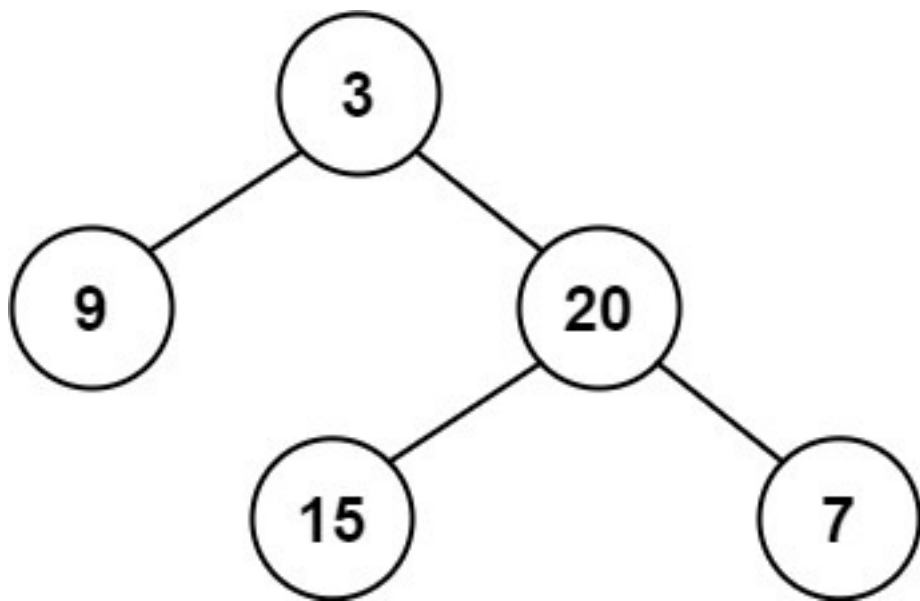
需要在报告中有以下两个示例的通过示例截图

输入：root = [3,9,20,null,null,15,7]	输入：root = [1,2,2,3,3,null,null,4,4]
输出：true	输出：false

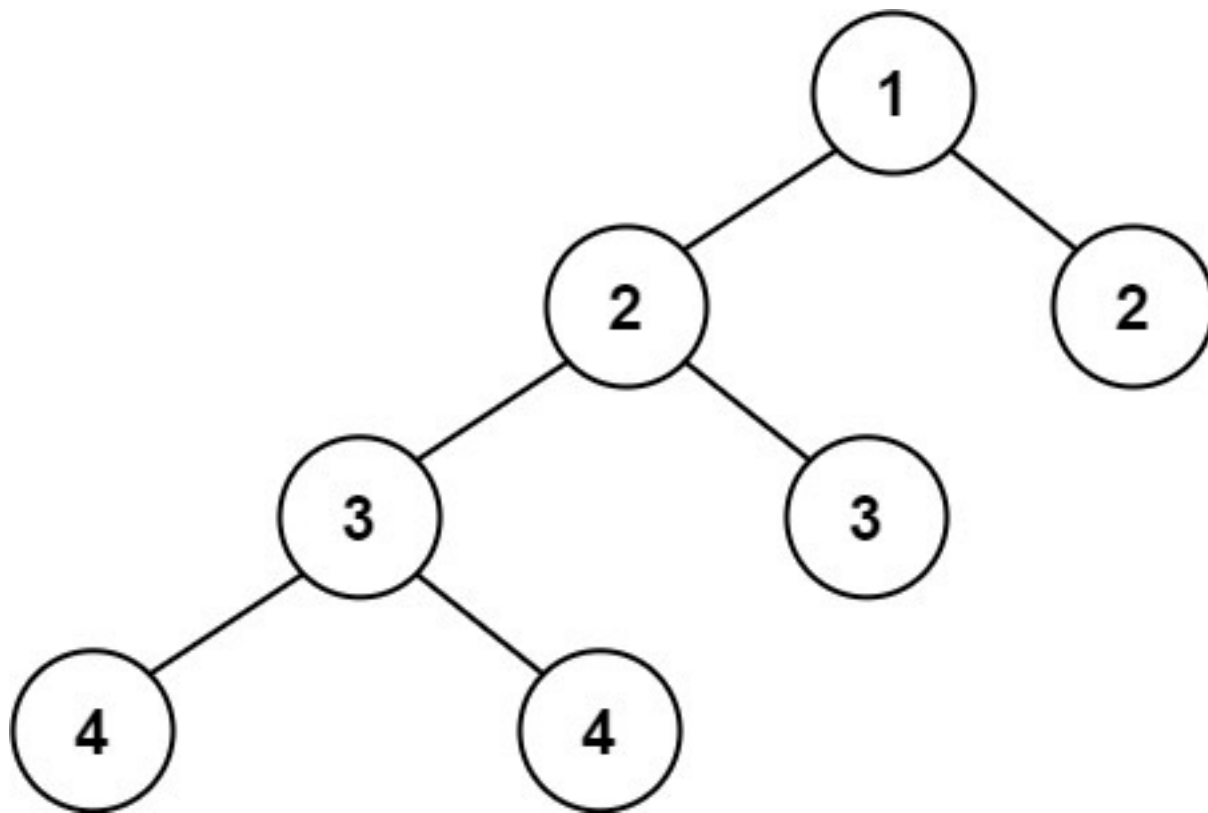
实验任务1



- 给定一个二叉树，判断它是否是高度平衡的二叉树（即一个二叉树每个节点的左右两个子树的高度差的绝对值不超过1）。



输入：root = [3,9,20,null,null,15,7]
输出：true



输入：root = [1,2,2,3,3,null,null,4,4]
输出：false



实验任务2

- 输入两棵二叉树A和B，判断B是不是A的子结构。(约定空树不是任意一个树的子结构)

需要在报告中有以下两个示例的通过示例截图

```
struct TreeNode {  
    int val;  
    struct TreeNode *left;  
    struct TreeNode *right;  
};
```

输入：A=[3,4,5,1,2], B=[4,1]

输出：true

输入：A=[3,null,5,1,2], B=[1,2]

输出：false

注意

- B是A的子结构，即A中有出现和B相同的结构和节点值
- 要求用递归求解

注：输入的树按层序遍历的形式输入的，参照任务1

实验任务3



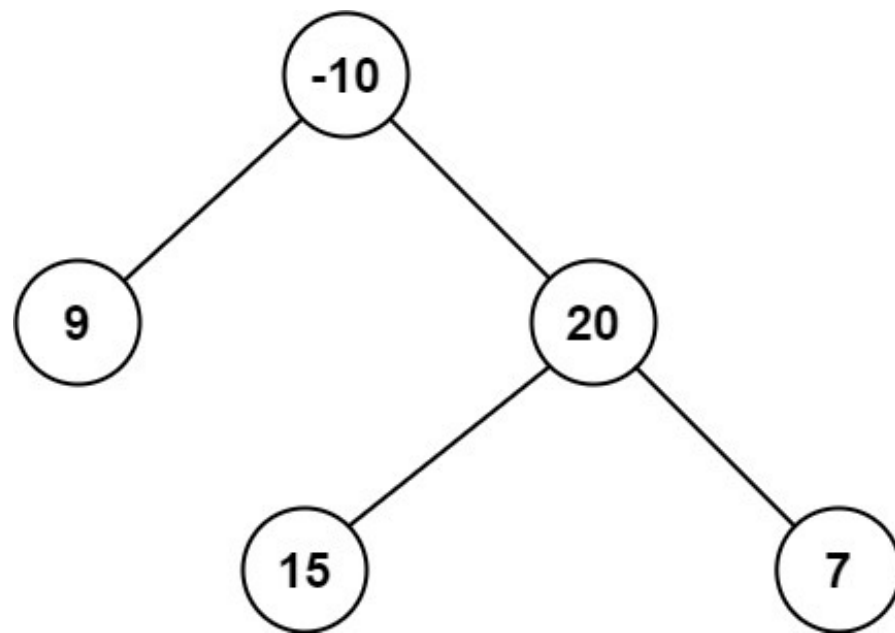
- 路径被定义为一条从树中任意节点出发，沿父节点-子节点连接，达到任意节点的序列。同一个节点在一条路径序列中至多出现一次。该路径至少包含一个节点，且不一定经过根节点。路径和是路径中各节点值的总和。给你一个二叉树的根节点 `root`，返回其最大路径和。

示例

输入：root = [-10,9,20,null,null,15,7]

输出：42

解释：最优路径是15->20->7，路径和为15+20+7=42



实验任务4



- 给定一个插入序列就可以唯一确定一棵二叉搜索树。然而，一棵给定的二叉搜索树却可以由多种不同的插入序列得到。例如分别按照序列{2, 1, 3}和{2, 3, 1}插入初始为空的二叉搜索树，都得到一样的结果。于是对于输入的各种插入序列，你需要判断它们是否能生成一样的二叉搜索树。
- 输入格式：输入包含若干组测试数据。每组数据的第1行给出两个正整数 N (≤ 10) 和 L ，分别是每个序列插入元素的个数和需要检查的序列个数。第2行给出 N 个以空格分隔的正整数，作为初始插入序列。随后 L 行，每行给出 N 个插入的元素，属于 L 个需要检查的序列。简单起见，我们保证每个插入序列都是1到 N 的一个排列。当读到 N 为0时，标志输入结束，这组数据不要处理。

实验任务4



输入格式：输入包含若干组测试数据。每组数据的第1行给出两个正整数 N (≤ 10) 和 L ，分别是每个序列插入元素的个数和需要检查的序列个数。第2行给出 N 个以空格分隔的正整数，作为初始插入序列。随后 L 行，每行给出 N 个插入的元素，属于 L 个需要检查的序列。简单起见，我们保证每个插入序列都是1到 N 的一个排列。当读到 N 为0时，标志输入结束，这组数据不要处理。

输出格式：对每一组需要检查的序列，如果其生成的二叉搜索树跟对应的初始序列生成的一样，输出 “Yes”，否则输出 “No”。

输入：

4 2
3 1 4 2
3 4 1 2
3 2 4 1
2 1
2 1
1 2
0

输出：

Yes
No
No

示例

提示：你可能需要定义二叉树的结点数据结构、建树函数、给二叉树插入结点的函数，以及比较树的结点是否相同的函数

课堂报告提交说明



- **注意：本次实验需要到对分易提交实验课堂报告(zip压缩包格式)**

- **压缩包中应包含：**

- **已实现的任务文件源码**（不需要包括拓展的任务）

- **一个doc或pdf文档**，上面需要有：

- 开头说明 **“整体实现参考 + 2-3句简要体会（如教训、思路、拓展应用等）”**，如：

- “自行实现。挑战最大的是任务2，初始时报了什么错，通过什么方式解决。”

- “任务3参考xxx同学/xxx网址。思考不出算法思路，探究后学习到了什么方法。”

- **各任务的成功运行截图**，示例（命名）如右：

- **评分说明：**

- **占总评2分**。当堂课后一周内提交得2分。

- 延误提交扣1分，助教随机抽查发现没有实现

或说明文档不完善，各扣1分，扣完为止。

```
C:\Users\gugua\Desktop\zhujiao\1.commonList.exe
5 8 13 22 50 100 -1
3 11 77 101 220 -1
请按任意键继续...
```

任务1_测试1.jpg