

## 练习 2 - 树木检查

以 2 人一组的方式完成以下编程示例。编程语言的选择取决于您（C、C++、C#、Java）。但是，数据结构必须自己实现！

### 任务

任务分为两部分。

1. 读入一棵树，输出统计信息
2. 在树中搜索

### 1. 读树

实现一个程序来检查二叉树是否是 AVL 树并返回有关树的统计信息。

主程序从文本文件中读取整数键值（文件名作为参数传递），并用这些值依次构建二叉搜索树。插入时丢弃重复的键值。

接下来应该**递归的**开发了输出二叉搜索树的每个节点的平衡因子的函数，从而检查给定的树是否是 AVL 树。如果节点中违反了 AVL 条件（平衡因子  $>1$  或  $<-1$ ），则应单独输出。此外，还应输出树的统计数据（最小键值、最大键值）和平均键值（所有键值的算术平均值）。该数据也应该通过遍历树来计算，而不是从输入文件中确定。

在实现之前，请考虑递归函数的结构（终止条件、参数、返回值……）并记录您的想法。此外，根据输入文件中整数值的数量  $N$  使用  $O$  表示法估计函数的工作量。

### 笔记程序

程序调用应如下所示：

**树检查 文件名**

### 注释文件格式

输入文件是一个文本文件，每行包含一个键。所有键都是整数值，并且允许任意数量。

例子：

5  
3  
17  
9  
23  
54  
11  
79  
30  
12

### 笔记数据结构

为二叉搜索树的节点定义适当的数据结构。在 C 中，结构可能如下所示：

```
结构节点{  
    整数键;  
    结构 tnode *left;  
    结构 tnode *right;  
};
```

节点  $\text{bal}(k)$  的平衡因子定义为

$\text{bal}(k) = h(\text{右子树}) - h(\text{左子树})$ 。

### 笔记输出

平衡因子按以下格式按节点输出：

球（键） = x

如果违反 AVL 条件，则由以下输出指示：

$\text{bal}(\text{key}) = x$ （违反 AVL！）

然后输出该树是否为 AVL 树（即是否所有节点都满足 AVL 条件）：

输出自 AVL：是的如果它是 AVL 树，或者 AVL：没有 如果它不是 AVL 树。

最后输出统计数据 min: x, max: y, avg: z

上述测试数据的参考输出：

球（79） = 0  
球（30） = 0

```
球 (54) = 0
bal(23) = 2 (违反 AVL! )
bal(12) = 0
球(11) = 1
bal(9) = 2 (违反 AVL! )
bal(17) = 0
球(3) = 0
bal(5) = 3 (违反 AVL! )
AVL: 否
最小: 3, 最大: 79, 平均: 24.3
```

您的程序生成的输出必须符合这种格式!

## 2. 在树中搜索

在任务的第一部分，您阅读了 AVL 树。现在您应该在树中查找条目和子树。

为此，您的程序在启动时必须接受两个参数。第一个参数表示具有描述搜索树的键值的文件，第二个参数表示包含要搜索的子树的文件。

与第一个任务相同的文件格式适用于搜索树和子树。子树必须至少包含一个键值，但也可以包含多个键值，代表一棵树。

因此，我们区分了两种搜索情况

- 简单搜索
- 搜索子树

在这两种情况下，您都不能在输入列表中进行搜索，而是在搜索树中进行搜索。请设计您的程序，使该规范在代码审查期间清晰可见。

### 笔记程序

程序调用应如下所示：

**树检查** 文件名搜索树文件名子树

### 简单搜索的提示

在简单搜索中，子树文件中只有一个键值。根和键值之间的所有节点都应作为成功搜索的结果输出。如果该条目不在树中，则输出要搜索的键值和“未找到！”。

例子：

-在搜索树 5、3、8、7、22、2 输出中成功搜索到 7：7 找到 5, 8, 7

-在搜索树 5、3、8、7、22、2 输出中搜索 1 失败：1 没找到！

### 搜索子树的注意事项

搜索子树时，应检查子树的结构是否存在于树中。如果子树的键值顺序也符合搜索树的键值顺序，则该结构被包含在搜索树中。

由于此解决方案很简单，因此需要先构建搜索树，然后**只要**通过递归遍历树来搜索结构。如果未找到子树，则成功搜索的结果应该是“找到子树”和“找不到子树！”。

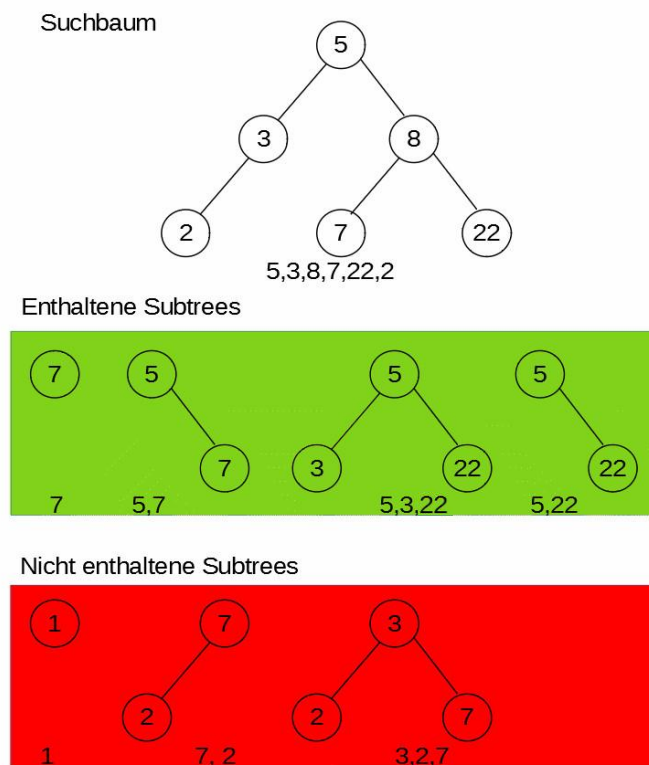
例子：

-在搜索树 5, 3, 8, 7, 22, 2 中成功搜索到 5, 7 输出：找到子树

-在搜索树 5, 3, 8, 7, 22, 2 中搜索 7, 2 不成功 输出：未找到子树！

### 搜索注意事项

下图显示了包含和不包含子树的示例。



### 征收

必须将 .zip 或 .tgz 文件提交到提交系统。这应该包括：

- 所有来源，包括代码注释！可执行程序
- 带有工作量估计描述的日志 这 递归的 职能 和
- 带有键值的测试文件

提交必须在第二次代码审查时提交，此练习最多可以达到 15 分。