



# 二叉树和节点的逻辑设计

2022 - 2023 秋冬 数据结构与算法

作者：王笑同

组织：数学与应用数学（强基计划）2101

时间：November 4, 2022



截至本次作业发布时，我们已经在课上学习了几种数据结构：单链表、双链表、二叉树、二叉搜索树、平衡二叉树。而这些数据结构又可以进一步概括为二叉树和链表两种。若将这两者进一步抽象，又可以得到它们的共性：

- 数据存放的单位为“结点”；
- 相邻“结点”之间可以互相遍历（或至少可以单向遍历）。

由此可见，二叉树和链表本质都可看作树，甚至是更一般的图。对于一张图  $G = (V, E)$ ，其每一个结点可以设计成抽象的基类 `Node`，其应该包含以下成员：

1. 数据域 `data`，代表结点存放的数据；
2. 邻接结点表 `neighbor`，依次存储每一个与该结点直接相连的结点的地址。

在其基础上，我们可以派生出适用于链表的结点类（模板）`ListNode`，甚至不用添加额外的成员，只需用 `neighbor[0]` 和 `neighbor[1]` 分别代表结点的前驱和后继即可。类似地，可以派生出二叉树结点类 `BinaryTreeNode`，用 `neighbor` 的前三个元素依次代表父亲、左儿子、右儿子的结点地址。利用 `BinaryTreeNode`，可以派生出平衡树的结点类 `AVLNode`，只需额外添加公有成员“子树的高度” `height`，照这样的思路，还可以派生出一般的树和图的结点类。

进一步，我们可以设计二叉树类 `BinaryTree`，其中可以包含以下成员：

**私有成员：**

1. 根节点 `root`；
2. ....

**公有成员：**

1. 为指定结点添加左右儿子的函数 `inserLeftChild()` 和 `insertRightChild()`；
2. 进行前/中/后序遍历，输出树中元素；
3. ....

在 `BinaryTree` 的基础上，可以派生出二叉搜索树类 `BinarySearchTree` 和 AVL 平衡树类 `AVLTree`，这里仅以后者为例，其包含的成员除基类中成员外，还应该有：

---

**私有成员：**

1. 旋转函数 `rotate()`;
2. ....

**公有成员：**

1. 向树中插入节点的函数 `insert()`;
2. 从树中删除结点的函数 `remove()`;
3. 将树置为空树的函数 `makeEmpty()`;
4. 判断树是否为空的函数 `isEmpty()`;
5. 升序输出二叉树中所有元素的函数 `printTree()`;
6. ....

也可以在二叉树的基础上实现一般的树和图的类模板，具体实现不再写出，其思路是完全类似的. 另外，关于以上函数的更详细解释，可参考头文件代码或 `latex` 目录下的 `doxygen` 文档.