# **二叉树理解**

单链表和数组的遍历可以是迭代的，也可以是递归的，二叉树这种结构无非就是二叉链表，由于没办法简单改写成迭代形式，只要是递归形式的遍历，都可以有前序位置和后序位置，分别在递归之前和递归之后。

### 二叉树遍历框架

Text

Description automatically generated with medium confidence

* **如果要求倒序打印链表上的所有节点的值**—> 用后序遍历， 本质上是利用了堆栈实现倒序遍历

# Diagram Description automatically generated

Text

Description automatically generated

# **二叉树框架**

Diagram

Description automatically generated

* **你可以发现每个节点都有「唯一」属于自己的前中后序位置**，所以我说前中后序遍历是遍历二叉树过程中处理每一个节点的三个特殊时间点。
* 二叉树的所有问题，就是让你在前中后序位置注入巧妙的代码逻辑，去达到自己的目的，**你只需要单独思考每一个节点应该做什么，其他的不用你管，抛给二叉树遍历框架，递归会在所有节点上做相同的操作**。

# **二叉树的解题思路**

**二叉树题目的递归解法可以分两类思路，第一类是遍历一遍二叉树得出答案，第二类是通过分解问题计算出答案，这两类思路分别对应着**[**回溯算法核心框架**](https://labuladong.github.io/algo/4/29/103/)**和**[**动态规划核心框架**](https://labuladong.github.io/algo/3/23/66/)。

# Python 实现二叉树的前序、中序、后序、层次遍历（递归和非递归版本）<https://blog.csdn.net/m0_37324740/article/details/82763901>

Diagram

Description automatically generated with medium confidence

前序遍历：根节点->左子树->右子树（1245367）

中序遍历： 左子树->根节点->右子树 (4251637)

后序遍历：左子树->右子树->根节点 (4526731)

层次遍历：逐层遍历 (1234567）