

# 编译原理与汇编关系



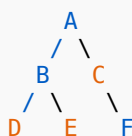
请向我展示一棵二叉树的后序历顺序且给出几个示例

**后序遍历 (Post-order Traversal)** 是二叉树的一种遍历方式，遍历顺序是：

1. 左子树
2. 右子树
3. 根节点

举个例子，假设有如下二叉树：

mathematica



后序遍历的过程是：

1. 先访问 **左子树**，即 `B` 的子树。
2. 再访问 **右子树**，即 `C` 的子树。
3. 最后访问根节点。

**步骤：**

- 对 `B` 进行后序遍历，先访问 `D`、然后访问 `E`，再访问 `B` 本身。
- 对 `C` 进行后序遍历，先访问 `F`，再访问 `C` 本身。
- 最后访问根节点 `A`。

**后序遍历的顺序：**

mathematica

D E B F C A

**示例 1：**

对于以下二叉树：

markdown



后序遍历的顺序是：

- 对 `2` 的子树进行后序遍历：`4`、`5`、`2`
- 对 `3` 的子树进行后序遍历：`3`
- 最后访问根节点 `1`。

后序遍历的顺序：

```
4 5 2 3 1
```

## 示例 2：

对于以下二叉树：

markdown



后序遍历的顺序是：

- 对 `5` 的子树进行后序遍历：`3`、`7`、`5`
- 对 `20` 的子树进行后序遍历：`25`、`20`
- 最后访问根节点 `10`。

后序遍历的顺序：

```
3 7 5 25 20 10
```

## 总结：

后序遍历的顺序是 **左子树** → **右子树** → **根节点**，你可以通过递归实现这一遍历方式。如果你需要具体的代码示例或者如何在程序中实现，随时告诉我！