

1. 定义

n个结点的二叉链表中含有n+1个空指针域。利用二叉链表中的空指针域，存放指向结点在某种遍历次序下的前趋和后继结点的指针（这种附加的指针称为“**线索**”）。

这种加上了线索的二叉链表称为**线索链表**，相应的二叉树称为**线索二叉树** (Threaded Binary Tree)。根据线索性质的不同，线索二叉树可分为**前序线索二叉树**、**中序线索二叉树**和**后序线索二叉树**三种。

注意：

线索链表解决了二叉链表找左、右孩子困难的问题，出现了无法直接找到该结点在某种遍历序列中的前趋和后继结点的问题。

2. 线索链表的结点结构

线索链表中的结点结构为：

lchild	ltag	data	rtag	rchild
--------	------	------	------	--------

其中：

ltag和rtag是增加的两个标志域，用来区分结点的左、右指针域是指向其左、右孩子的指针，还是指向其前趋或后继的线索。

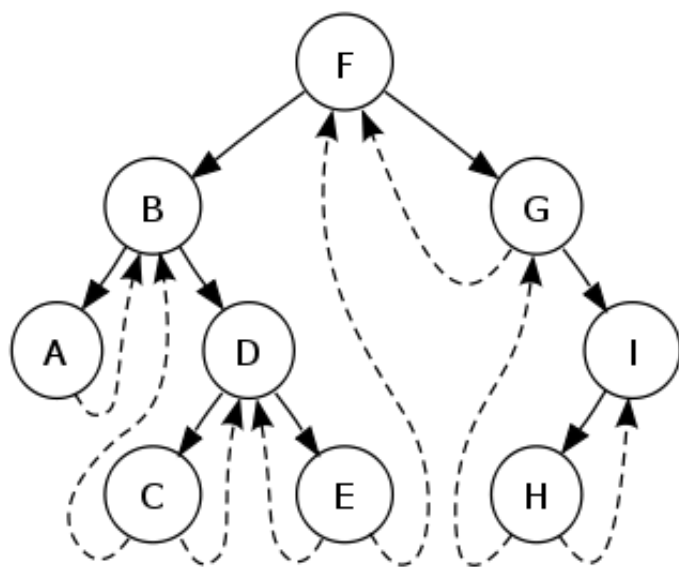
左标志ltag = $\begin{cases} 0: lchild \text{ 是指向结点的左孩子的 指针} \\ 1: lchild \text{ 是指向结点的前趋的左 线索} \end{cases}$

右标志rtag = $\begin{cases} 0: rchild \text{ 是指向结点的右孩子的 指针} \\ 1: rchild \text{ 是指向结点的后继的右 线索} \end{cases}$

3. 线索二叉树的表示

【例】下面(a)图所示的**中序线索二叉树**，其线索链表如下面(b)图所示。

中序遍历为：A B C D E F G H I（红色的节点有空指针, 可加线索）



注意：

图中的实线表示指针，虚线表示线索。

结点A的左线索为空，表示A是中序序列的开始结点，无前趋；

结点I的右线索为空，表示I是中序序列的终端结点，无后继。

线索二叉树中，一个结点是叶结点的充要条件为：左、右标志均是1。