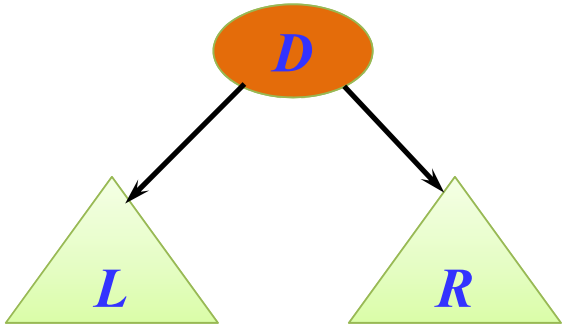
6.3 二叉树的遍历

6.3.1 遍历二叉树

- 二叉树的遍历是指按一定的次序访问树中的所有结点，使每个结点恰好被访问一次。其中遍历次序保证了二叉树上每个结点均被访问一次且仅有一次。
- 遍历就是把二叉树结点按某种次序排列成线性序列。
- 遍历是二叉树最基本的运算，是二叉树中其他运算的基础。

QDU-LZR

二叉树的组成：



- DLR
- LDR
- LRD

只考虑L → R的情况

- ~~DRL~~
- ~~RDL~~
- ~~RLD~~

- 二叉树常用的遍历有先序（根）遍历、中序（根）遍历、后序（根）遍历。
- 所谓先序、中序、后序，区别在于访问根结点的顺序。

QDU - LZR

1. 先序遍历

若二叉树非空，则：

① 访问根结点；

② 先序遍历左子树；

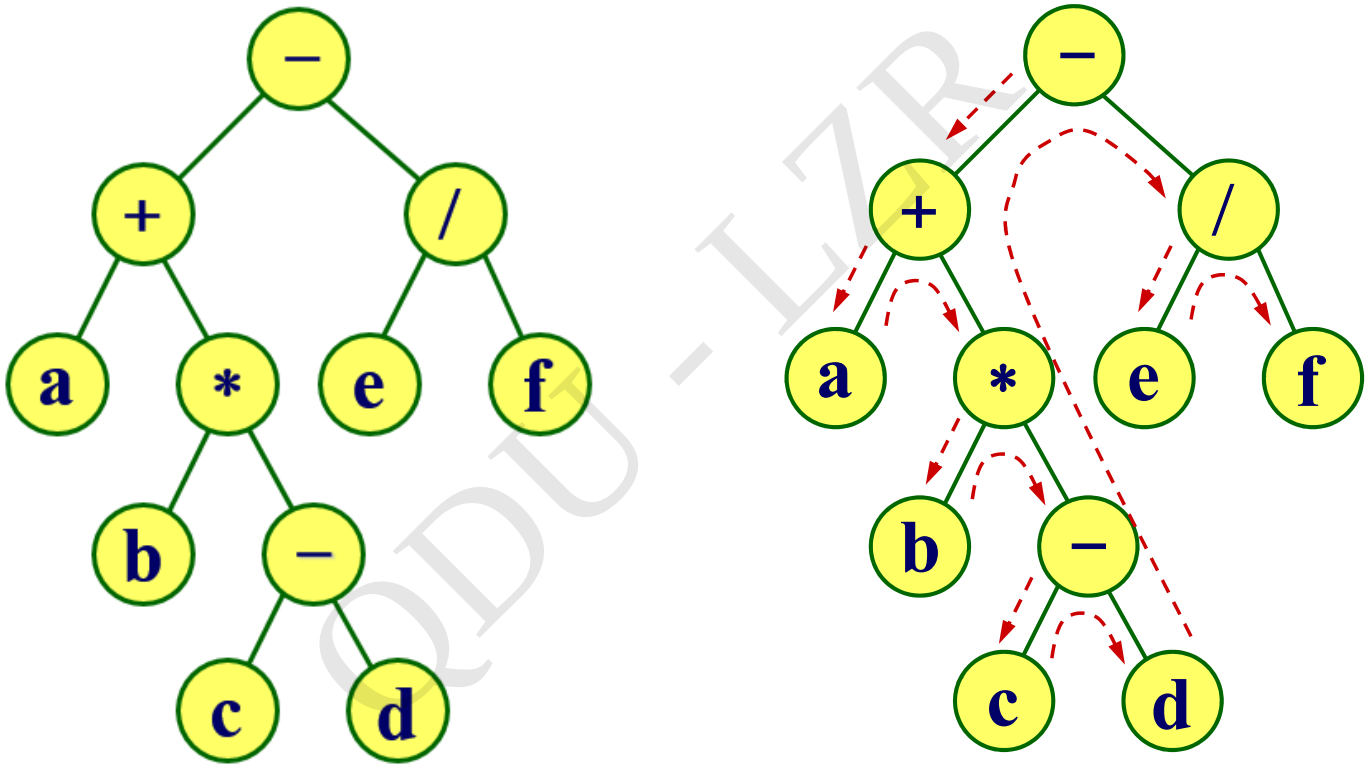
③ 先序遍历右子树。

```
void PreOrder(BiTNode *bt)
{   if (bt!=NULL)
    {   printf("%c ",bt->data);
        PreOrder(bt->lchild);
        PreOrder(bt->rchild);
    }
}
```

QDU_LZR

- 先序遍历序列的特点：其第一个元素值为二叉树中根结点值。

先序遍历DLR的演示



遍历结果：−+a*b−cd/ef

QDU - LZR

2. 中序遍历

若二叉树非空，则：

① 中序遍历左子树；

② 访问根结点；

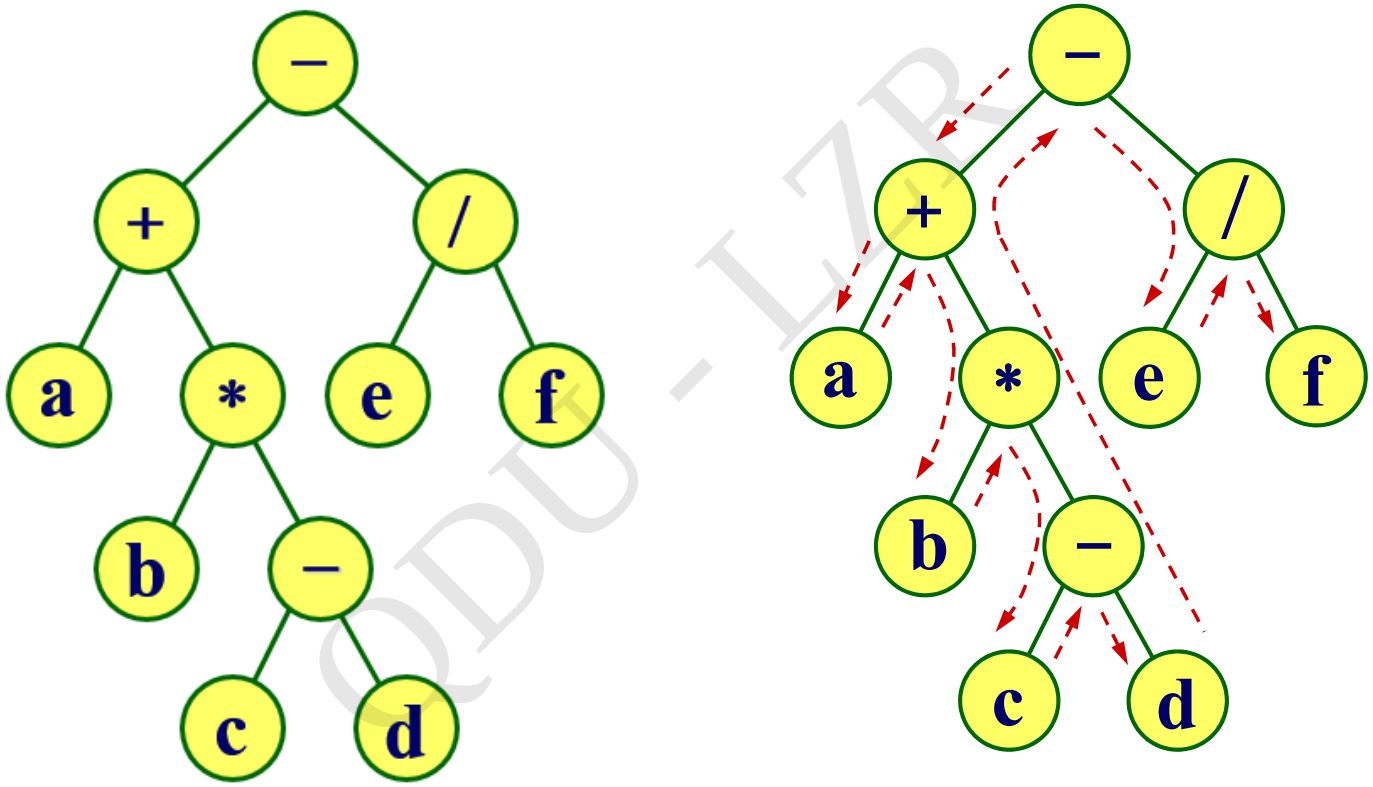
③ 中序遍历右子树。

```
void InOrder(BiTNode *bt)
{   if (bt!=NULL)
    {   InOrder(bt->lchild);
        printf("%c ",bt->data);
        InOrder(bt->rchild);
    }
}
```

QDU - LZR

- 中序遍历序列的特点：若已知二叉树的根结点值，以该值为界，将中序遍历序列分为两部分，前半部分为左子树的中序遍历序列，后半部分为右子树的中序遍历序列。

中序遍历LDR的演示



遍历结果： a + b * c − d − e / f

QDU - LZR

3. 后序遍历

若二叉树非空，则：

① 后序遍历左子树；

② 后序遍历右子树；

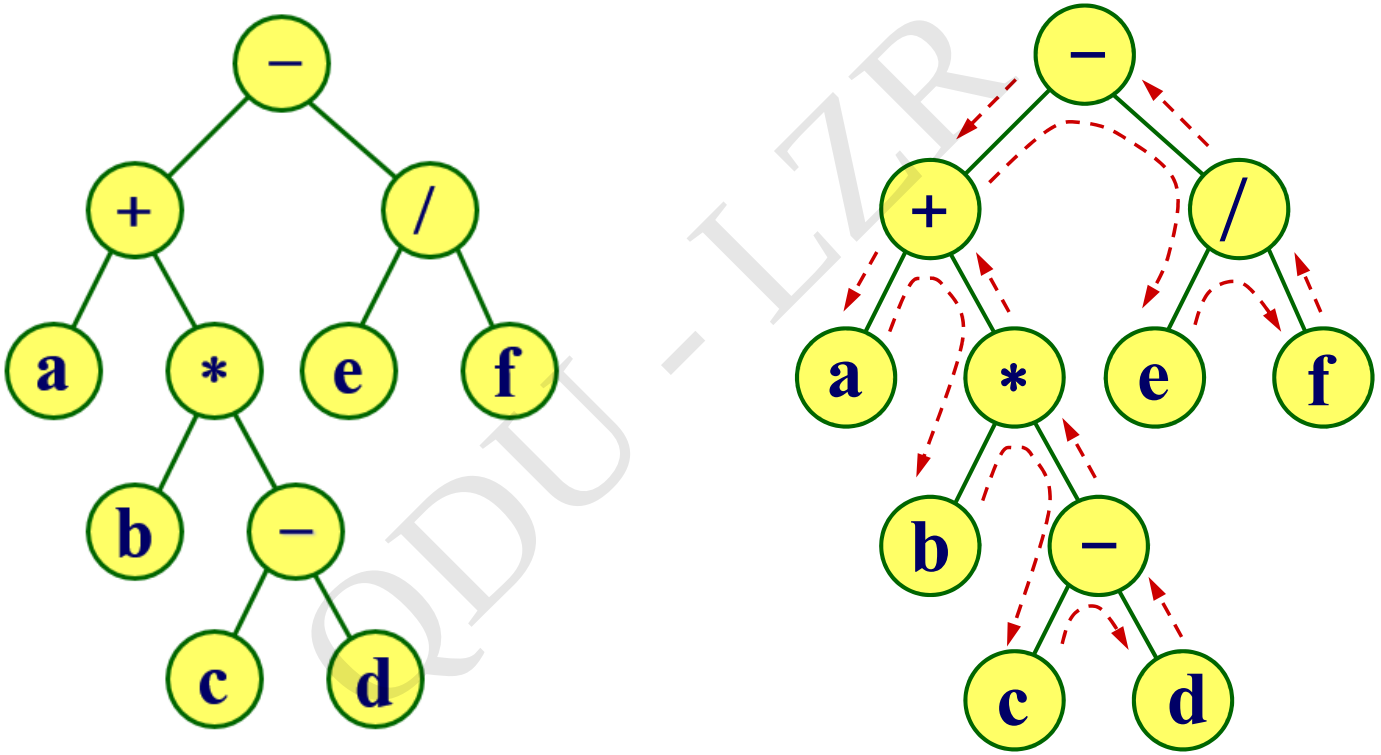
③ 访问根结点。

```
void PostOrder(BiTNode *bt)
{   if (bt!=NULL)
    {   PostOrder(bt->lchild);
        PostOrder(bt->rchild);
        printf("%c ",bt->data);
    }
}
```

QU - LZR

- 后序遍历序列的特点：最后一个元素值为二叉树中根结点值。

后序遍历LRD的演示



QDU - LZR

遍历结果： a b c d − * + e f / −

END

QDU-LZR