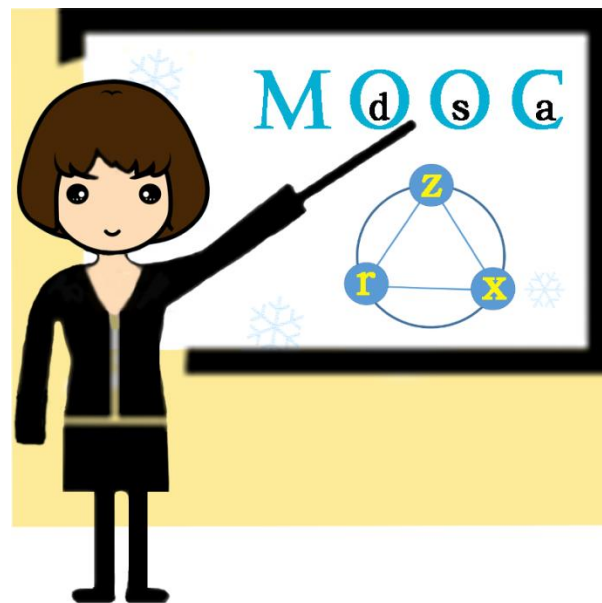
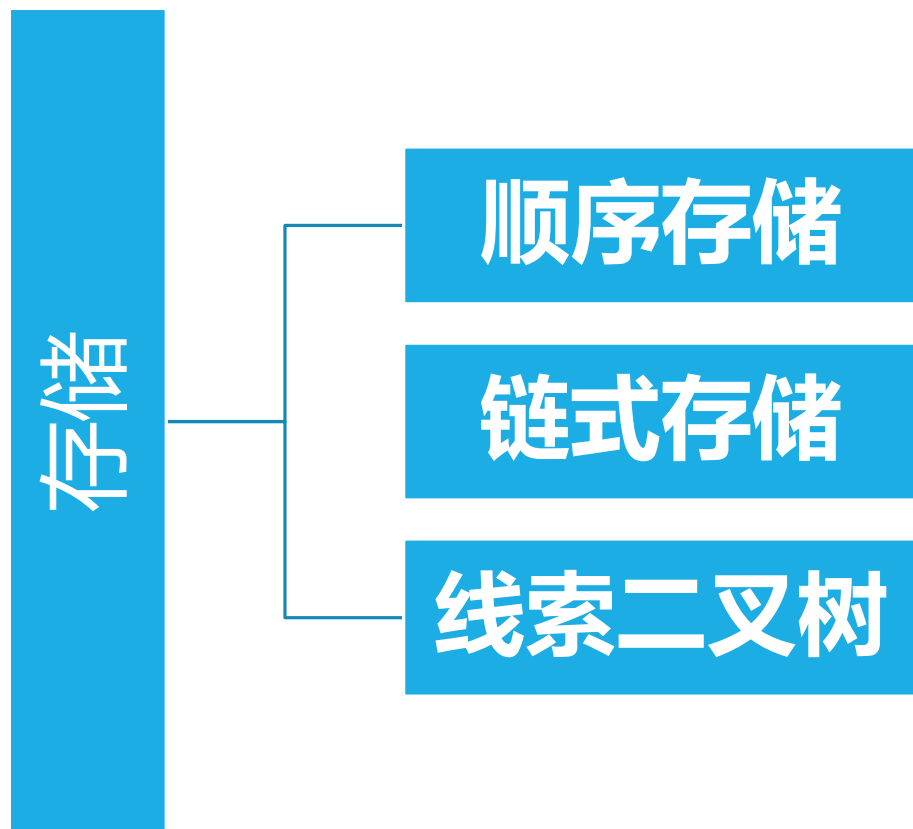


4.8 二叉树的链式存储



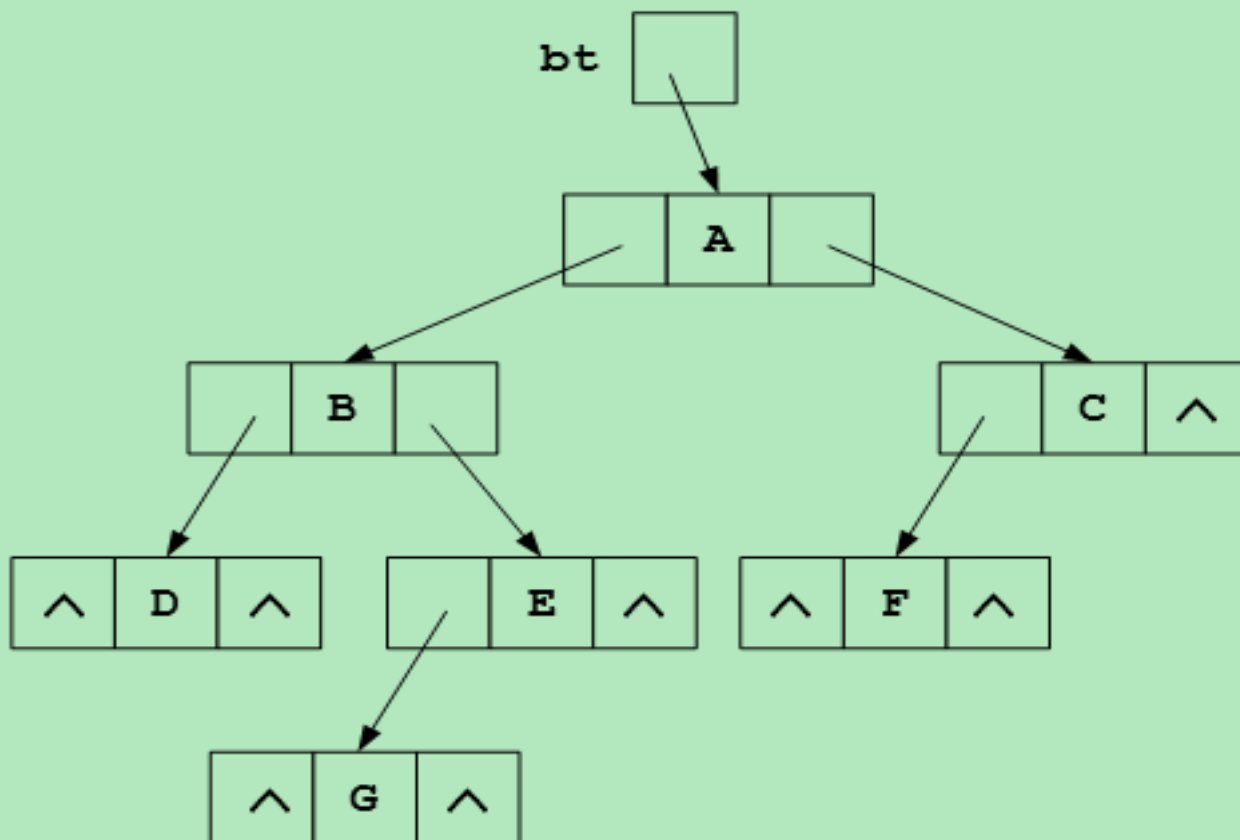
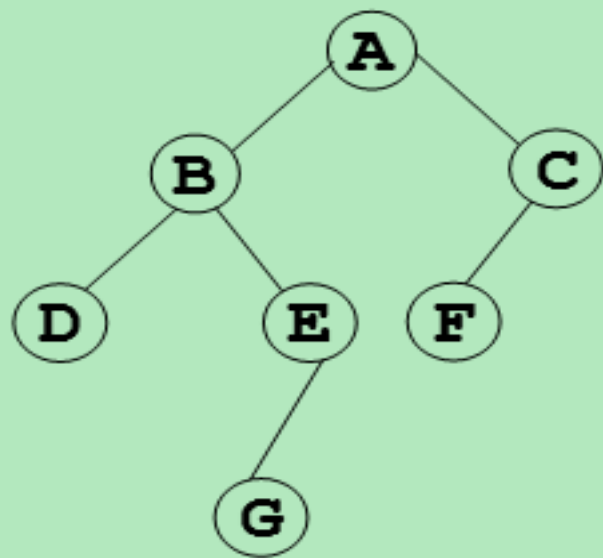
二叉树的链式存储



□ 一个二叉链表由头指针唯一确定

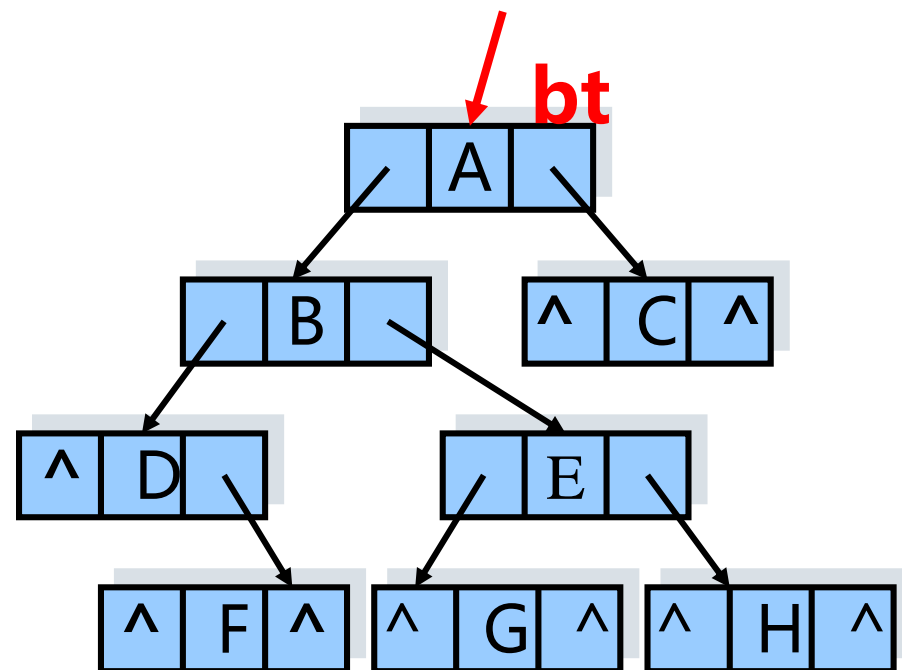
□ 若二叉树为空，则 $bt = \text{NULL}$

□ 若结点的某个孩子不存在，则相应的指针为空



二叉树的链式存储

```
1 typedef char DataType;  
2 typedef struct BTreeNode  
3 {  
4     DataType data;  
5     struct BTreeNode *leftchild;  
6     struct BTreeNode *rightchild;  
7 } BinTreeNode;  
8 typedef BinTreeNode *BinTree;
```



二叉树的链式存储

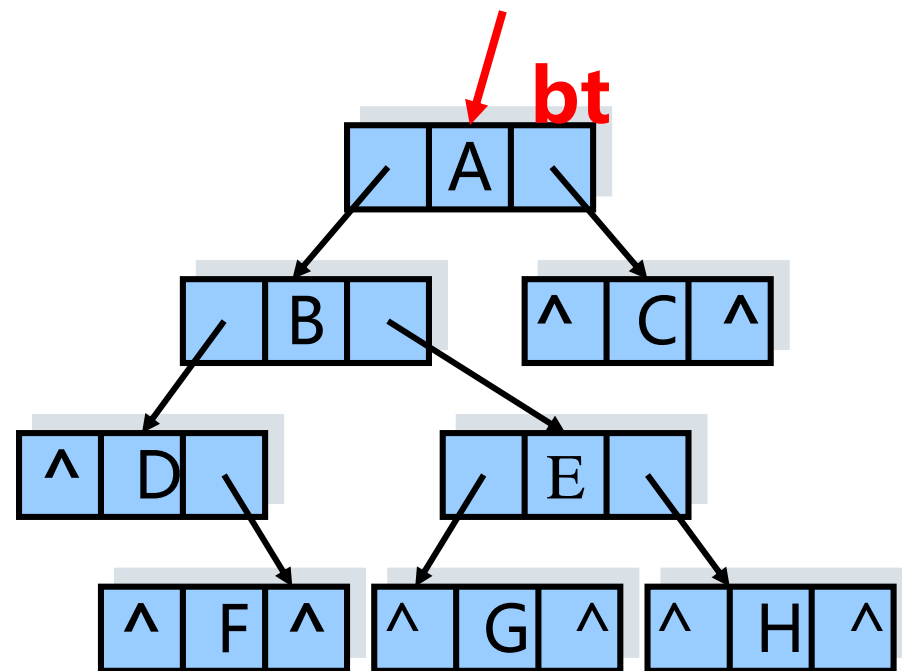
优点： 二叉链表结构灵活，操作方便

缺点： 在二叉链表中无法由结点直接找到其双亲

◆ 具有 n 个结点的二叉树中，一共有 $2n$ 个指针域，其中只有 $n-1$ 个用来指示结点的左、右孩子，其余的 $n+1$ 个指针域为空？

◆ 如何利用这些空指针域呢？

◆ 线索二叉树



三叉链表



- ◆ 每个结点由四个域组成，parent域为指向该结点双亲结点的指针
- ◆ 既便于查找孩子结点，又便于查找双亲结点
- ◆ 相对于二叉链表存储结构而言，它增加了空间开销

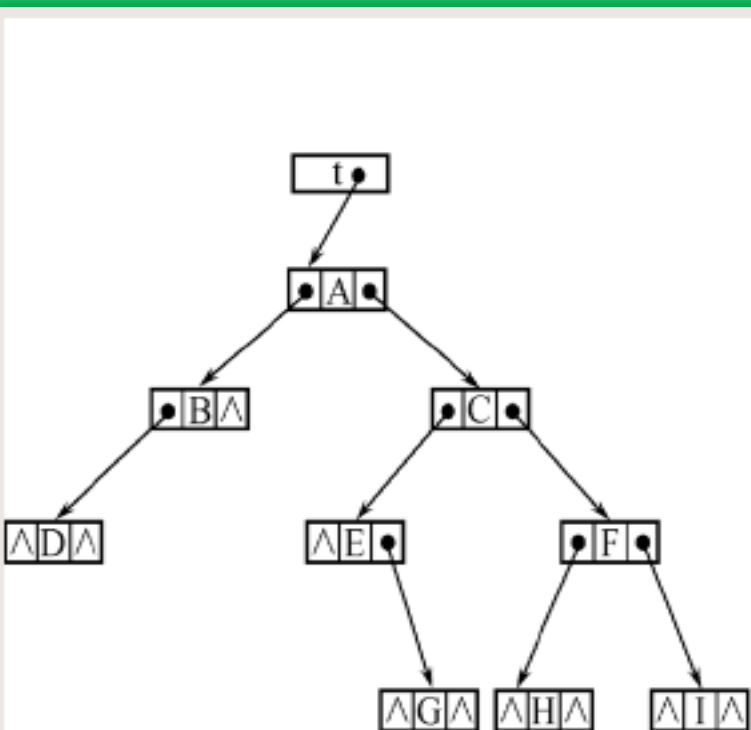
三叉链表

leftchild

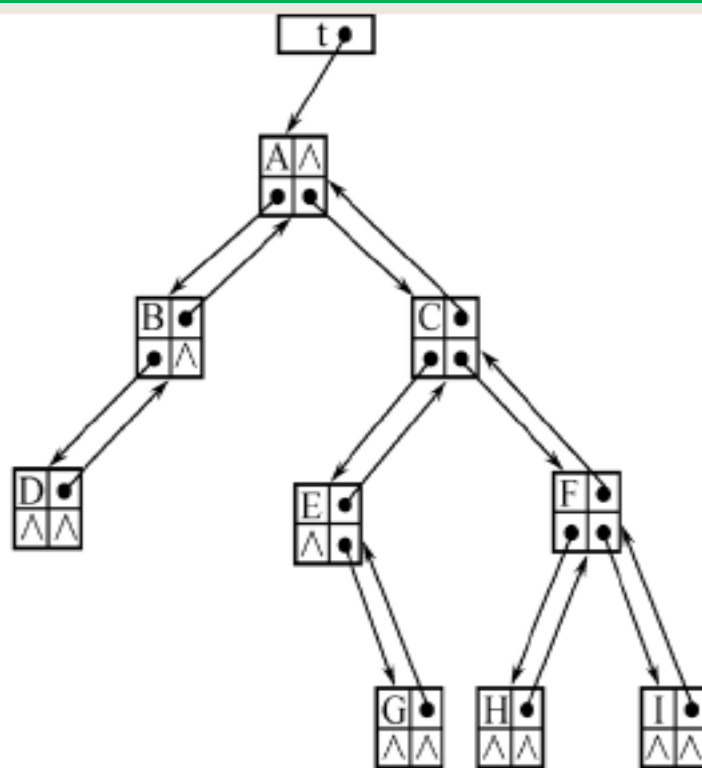
data

rightchild

parent



(a) 左-右指针表示



(b) 三叉链表表示

- 每个结点由四个域组成，parent域为指向该结点双亲结点的指针
- 既便于查找孩子结点，又便于查找双亲结点
- 相对于二叉链表存储结构而言，它增加了空间开销