

## 4.1.2 二叉树的存储结构

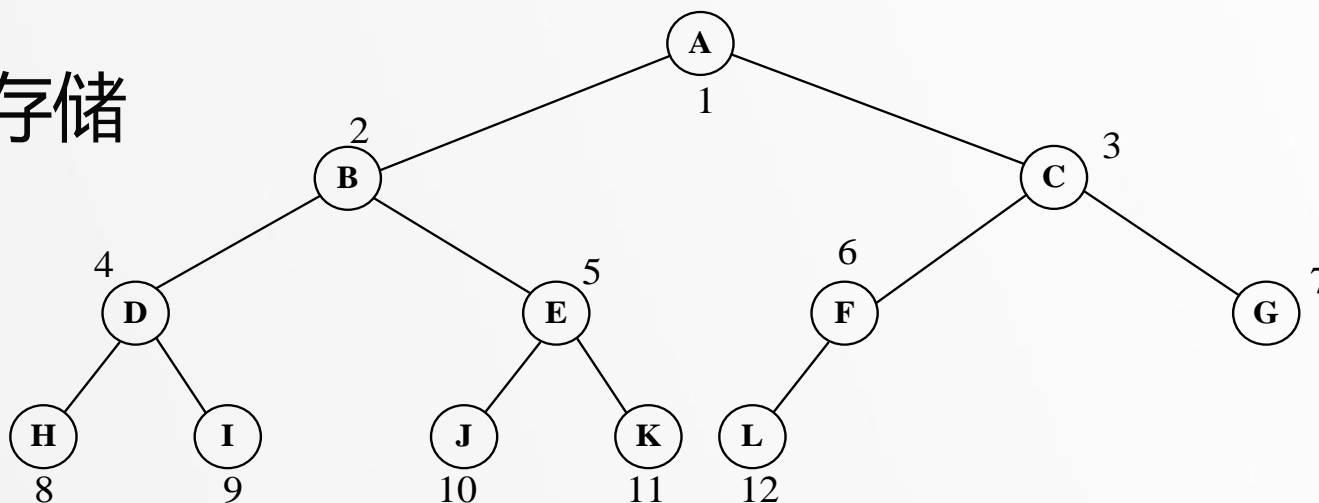
一、顺序存储

二、链式存储

## 一、二叉树的顺序存储结构

用一组地址连续的存储单元，以层序顺序存放二叉树的数据元素，结点的相对位置蕴含着结点之间的关系。

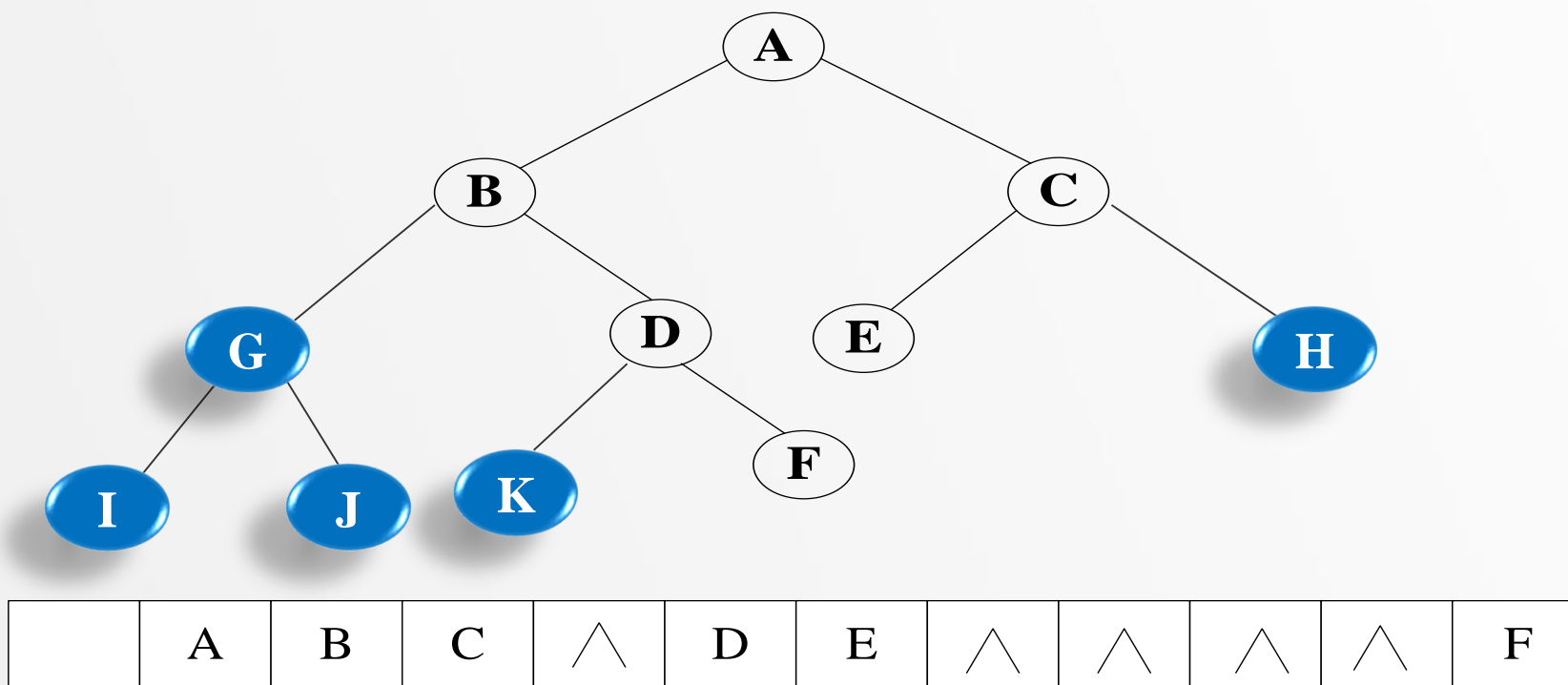
完全二叉树的顺序存储



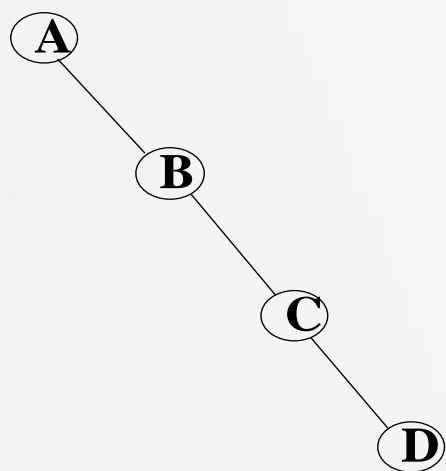
	A	B	C	D	E	F	G	H	I	J	K	L
0	1	2	3	4	5	6	7	8	9	10	11	12

## 一般二叉树的顺序存储

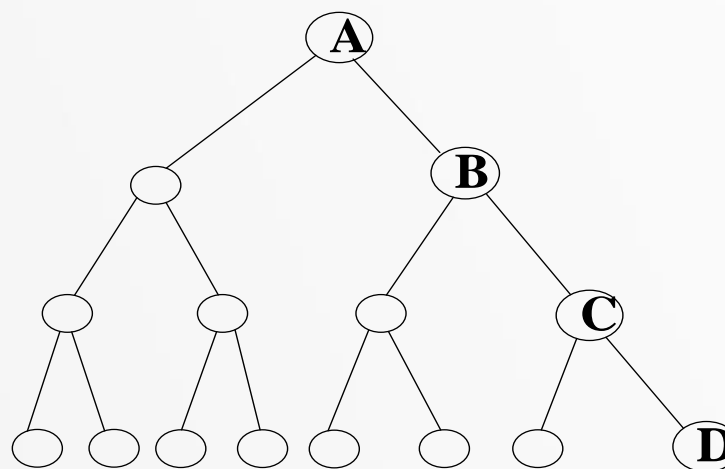
把一般的二叉树先补成完全二叉树，然后按照完全二叉树的顺序存储方式进行存储，而新补上去的结点只占位置，不存放结点数据。



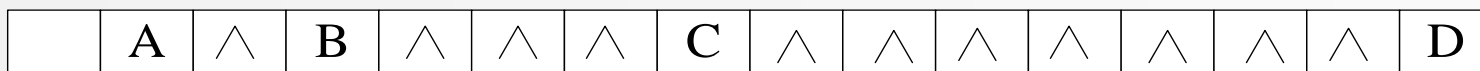
# 一般二叉树的顺序存储



(a) 右偏斜二叉树



(b) 补全后的完全二叉树



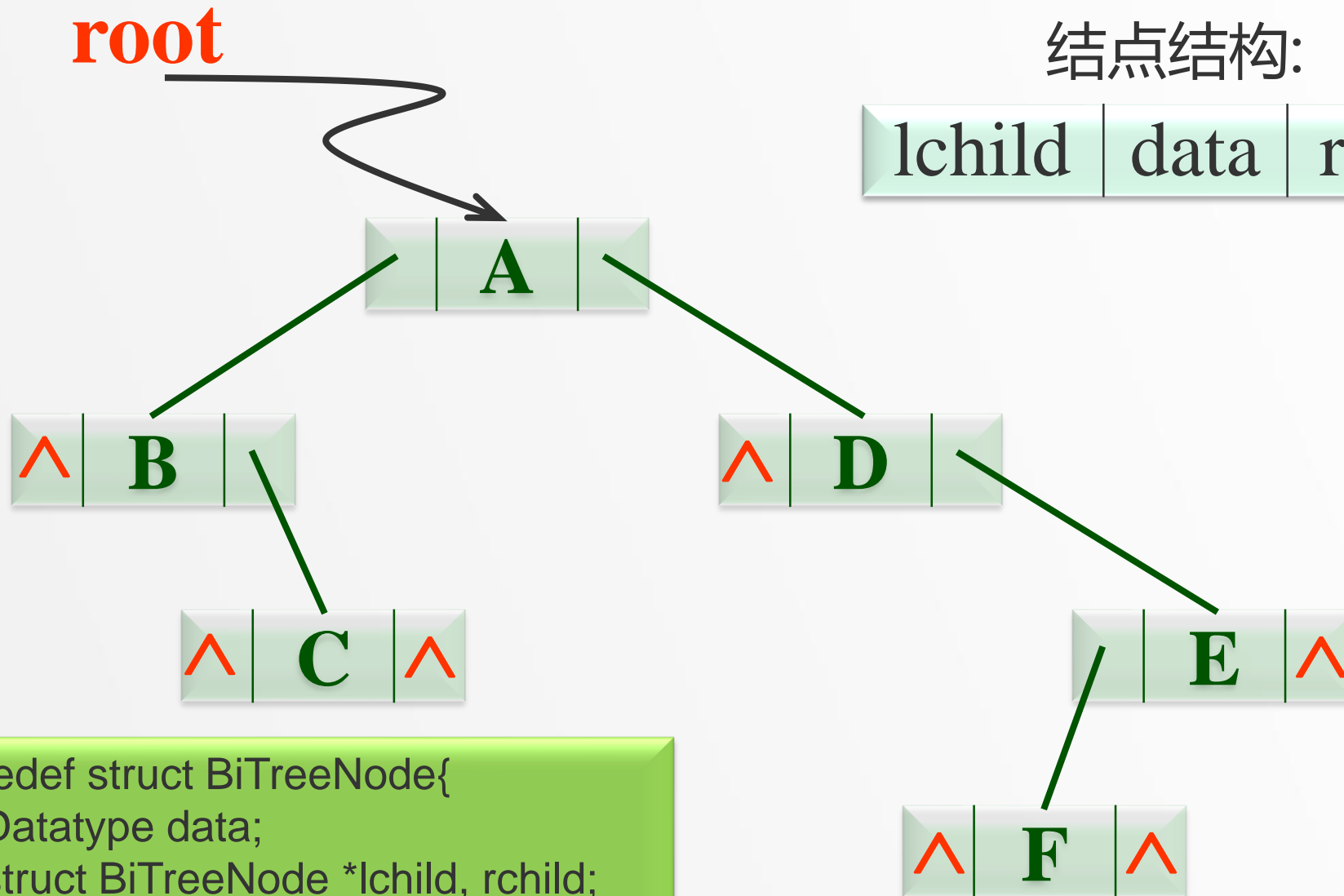
(c) 右偏斜二叉树的顺序存储示意图

## 二、二叉树的链式存储

1. 二叉链表
2. 三叉链表

# 1. 二叉链表 root

结点结构:



```
typedef struct BiTreeNode{
    Datatype data;
    struct BiTreeNode *lchild, rchild;
}BiTreeNode, *BiTree;
```

## 2. 三叉链表

结点结构:

parent	lchild	data	rchild
--------	--------	------	--------

```
typedef struct BiTreeNode{
    Datatype data;
    struct BiTreeNode *lchild, *rchild, *parent;
}BiTreeNode, *BiTree;
```

