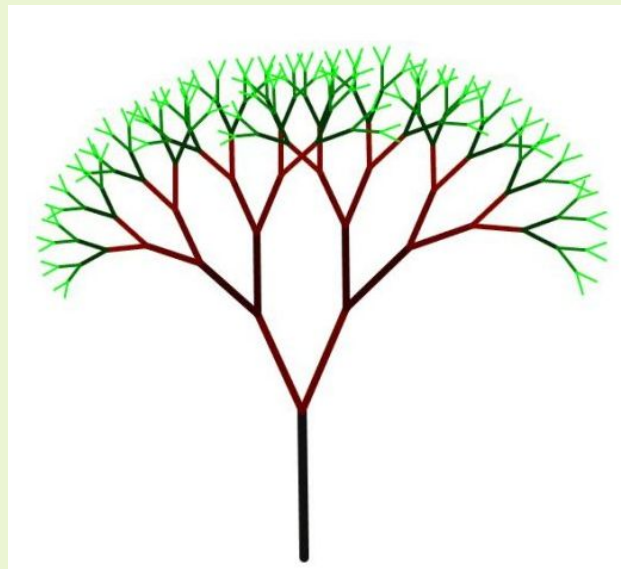


本节主题:

二叉树的存储结构

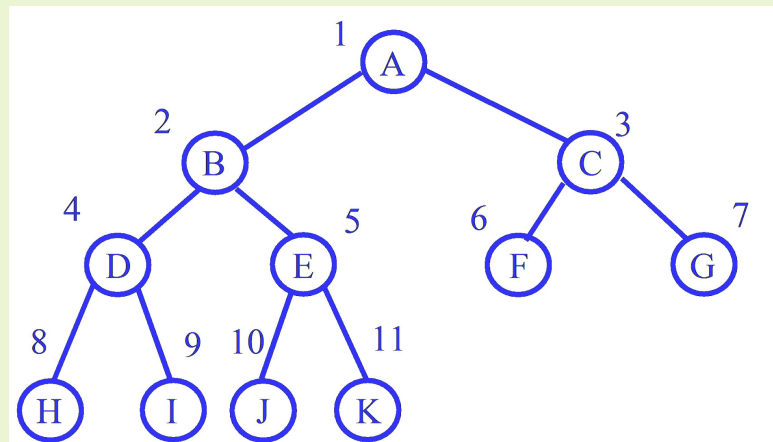


顺序存储结构

按编号次序存储结点

对树中每个节点进行编号

其编号从小到大的顺序就是节点在连续存储单元的先后次序。



[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] ...

—	A	B	C	D	E	F	G	H	I	J	K				
---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--

若编号为 i 的节点的左孩子节点的编号为 $2i$ ；右孩子节点的编号为 $(2i+1)$ 。

除树根节点外,若一个节点的编号为 i ，则它的双亲节点的编号为 $\lfloor i/2 \rfloor$

对一般的二叉树

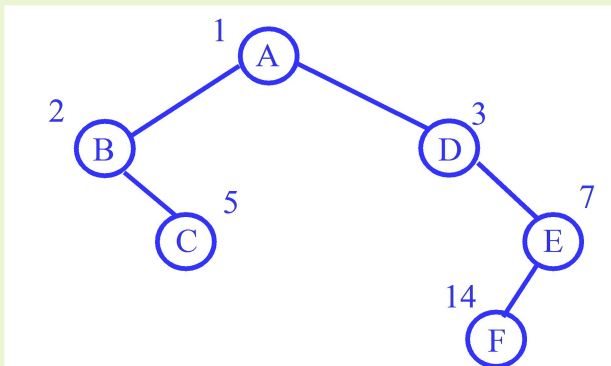
先用空节点补全成为完全二叉树

再对节点编号

[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] ...

最后确定存储

--	A	B	D	#	C	#	E	#	#	#	#	#	#	F	#
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



```
typedef ElemType SqBTree[MaxSize];  
SqBTree bt="#ABD#C#E#####F";
```

bt[i]的子女 : bt[2*i], bt[2*i+1]

bt[i]的父母 : bt[$\lfloor i/2 \rfloor$]

二叉树的链式存储结构

```
typedef struct node
{
    ElemType data;
    struct node *lchild,*rchild;
} BTreeNode;
```

其中,

- data表示值域,用于存储对应的数据元素,
- lchild和rchild分别表示左指针域和右指针域,用于分别存储左孩子节点和右孩子节点(即左、右子树的根节点)的存储位置。

