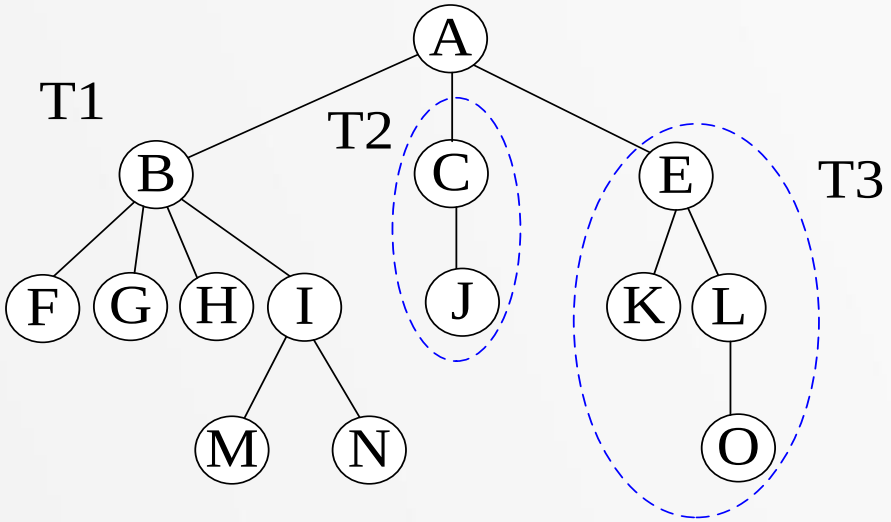
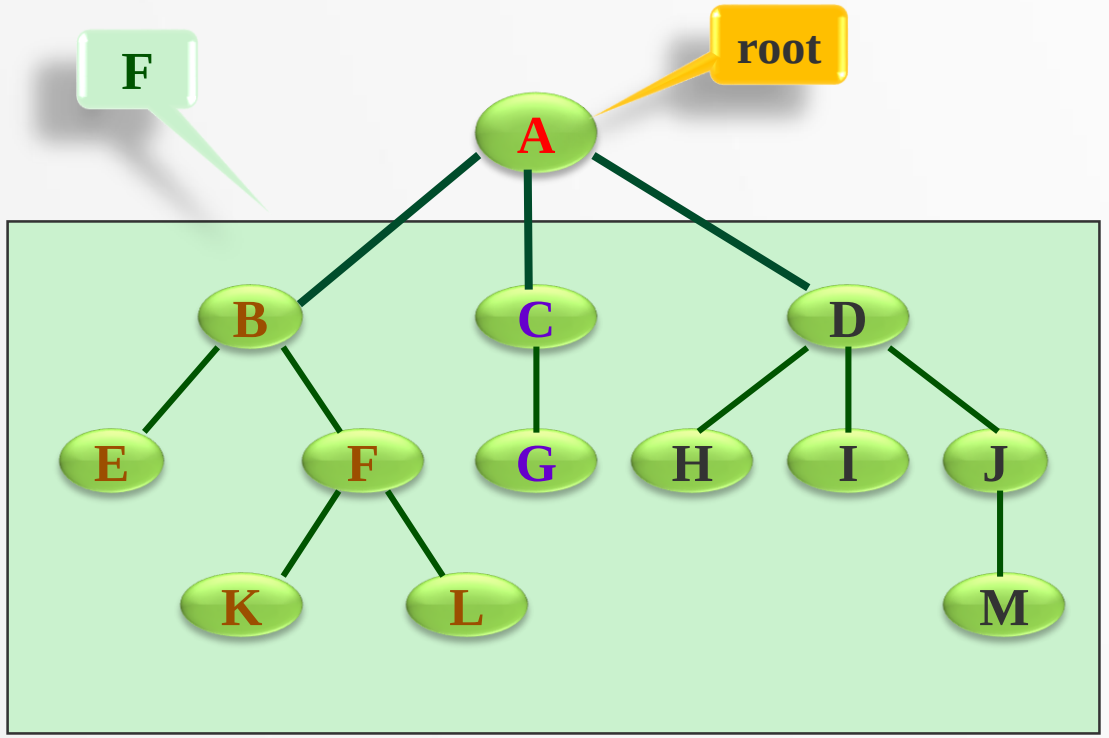
4.3 树和森林

4.3.1 树和森林的定义

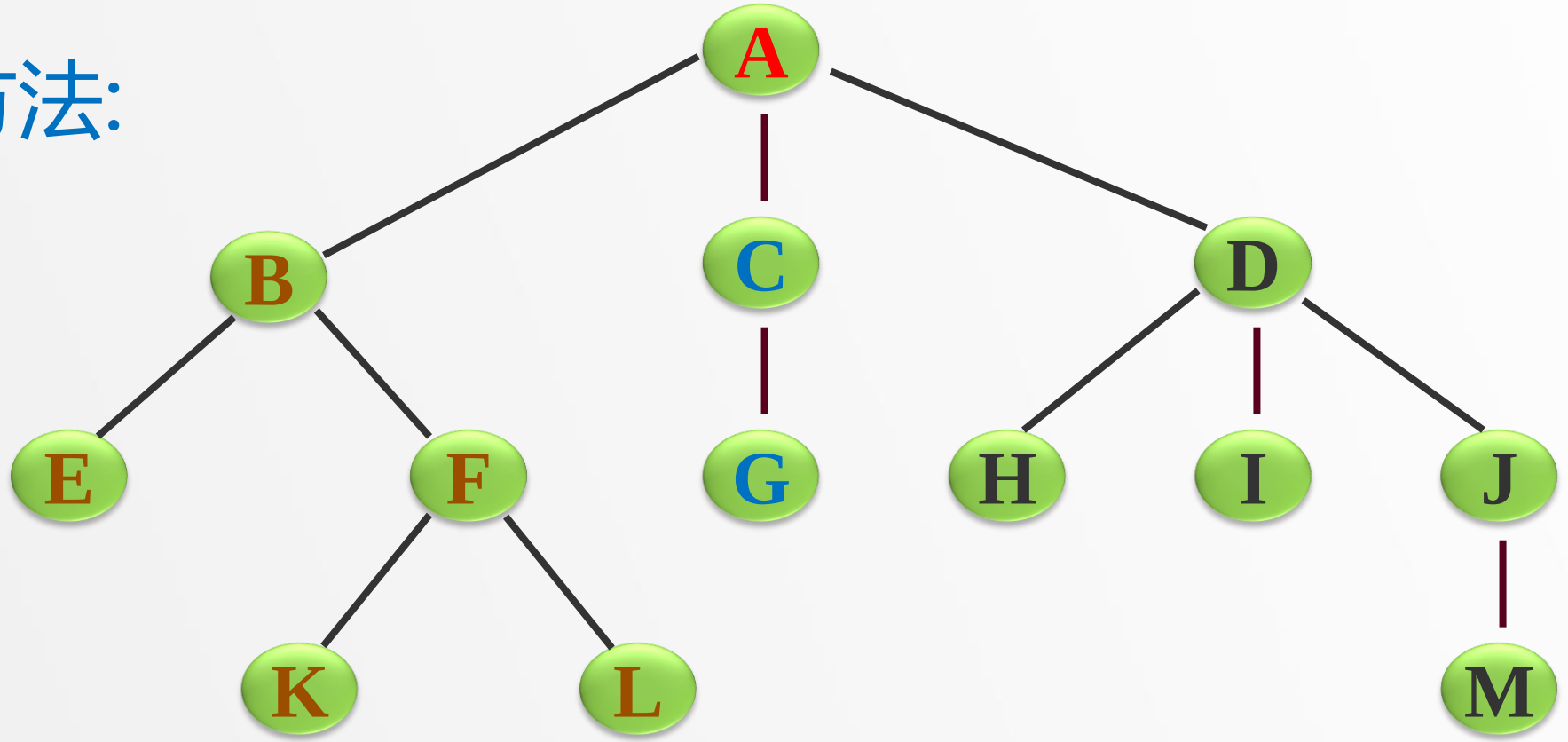
1. 树的定义



2.森林定义



表示方法:



A(B(E, F(K, L)), C(G), D(H, I, J(M)))

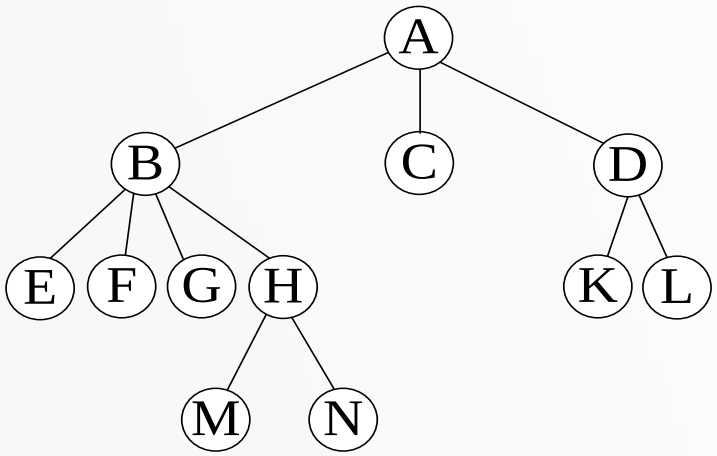
树根

T_1 T_2 T_3

4.3.2 树和森林的存储结构

1. 树的双亲表示法

```
#define MAX_TREE_SIZE 100
typedef struct PTNode{
    DataType data;
    int parent;
} PTNode;
typedef struct PTree{
    PTNode nodes[MAX_TREE_SIZE];
    int r, n;
} PTree;
```



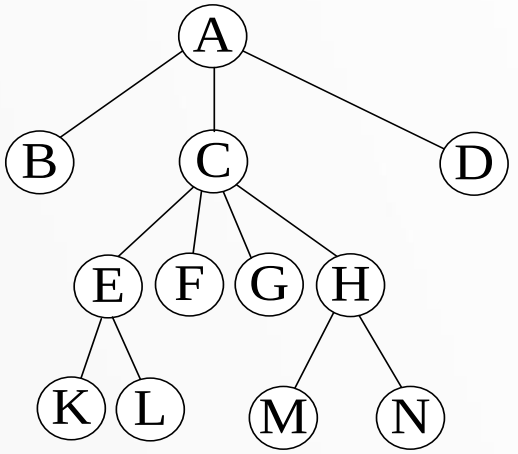
数组下标

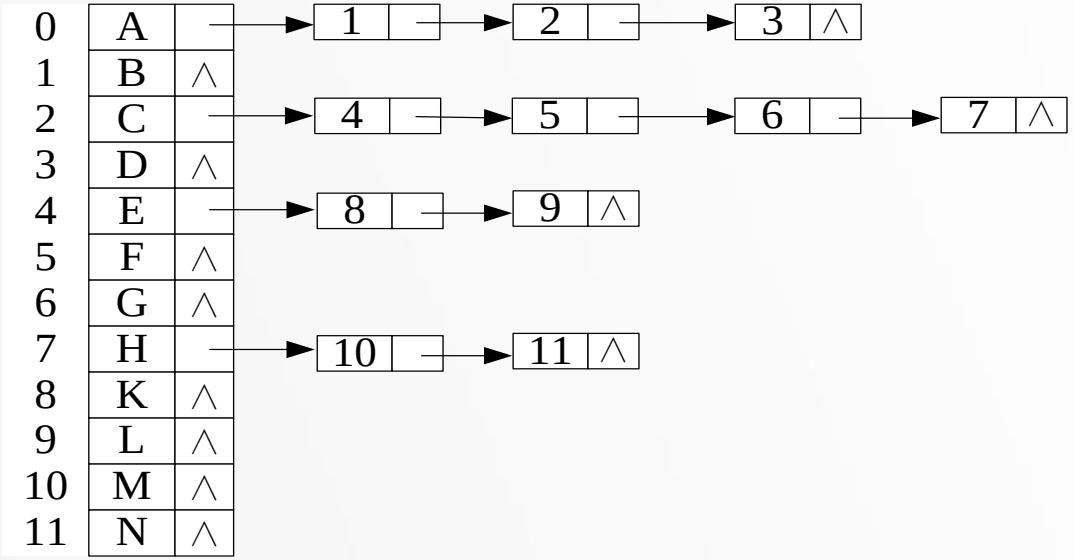
0	A	-1
1	B	0
2	C	0
3	D	0
4	E	1
5	F	1
6	G	1
7	H	1
8	K	3
9	L	3
10	M	7
11	N	7

4.3.2 树和森林的存储结构

2.树的孩子表示法

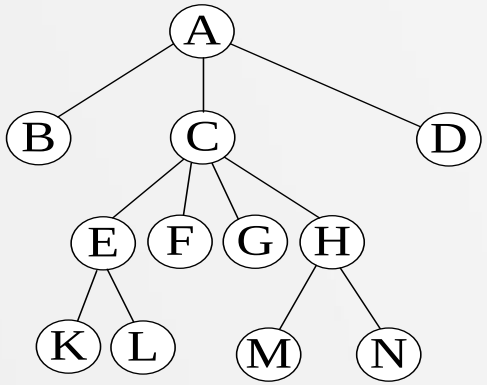
```
typedef struct CTNode{
    int child;
   struct CTNode *next;
}CTNode;
typedef struct CTBox{
    DataType data;
   CTNode *firstchild;
}CTBox;
typedef struct CTree{
    CTBox nodes[MAX_TREE_SIZE];
    int n,r;
} CTree;
```

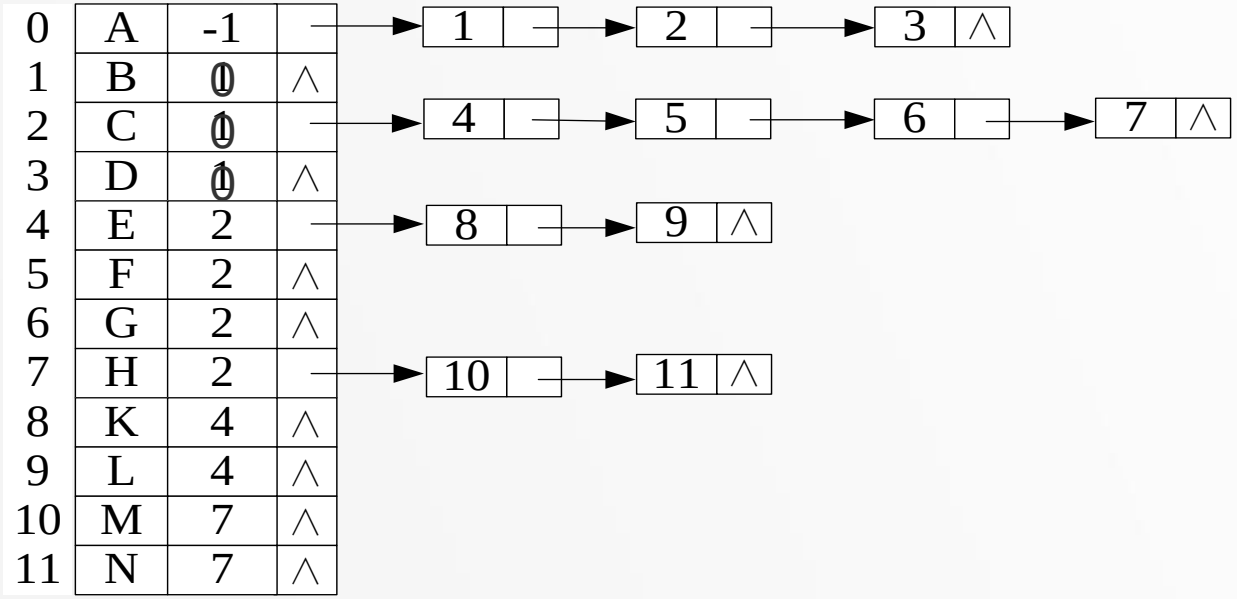




4.3.2 树和森林的存储结构

3.树的双亲孩子表示法



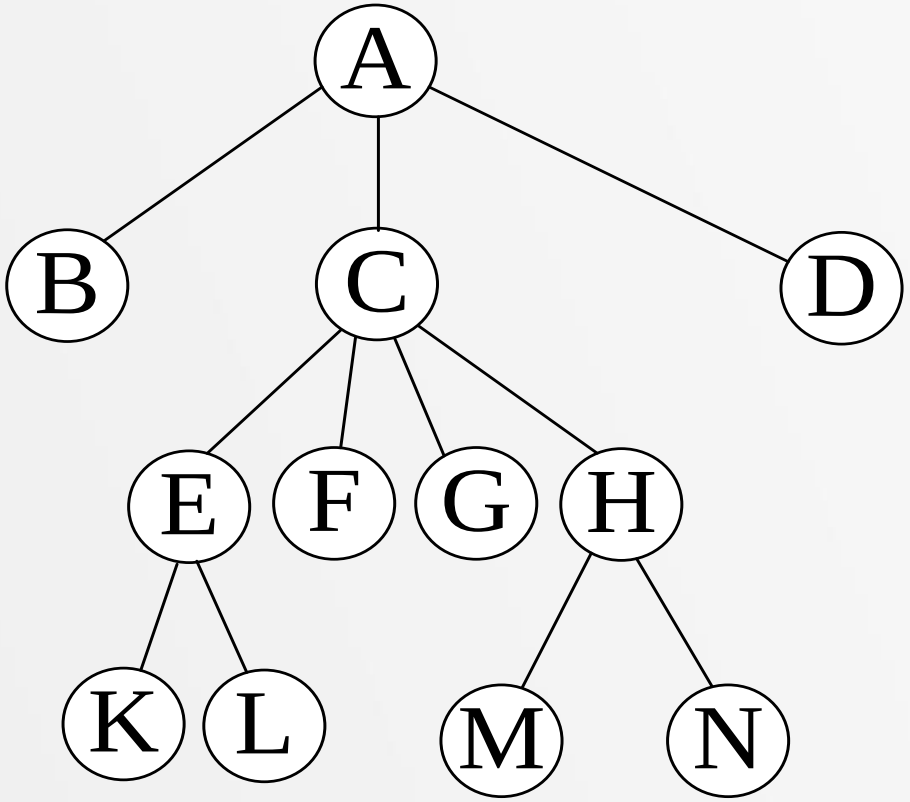


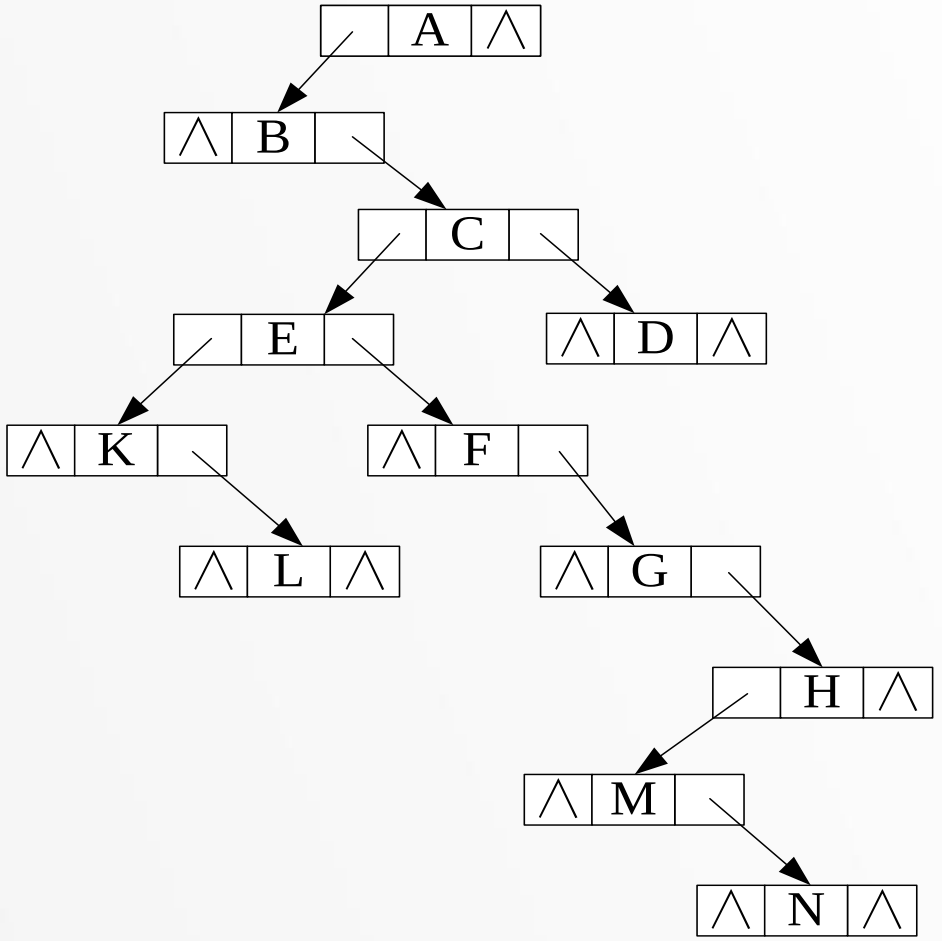
4.3.2 树和森林的存储结构

4.树的孩子兄弟表示法

结点结构描述如下：

```
typedef struct CSNode{
    DataType data;
    struct CSNode *firstchild, *nextsibling;
} CSNode;
```


A
B
C
D
E
F
G
H
K
L
M
N


A
B
C
E
D
K
F
L
G
H
M
N

4.3.2 树和森林的存储结构

5. 森林的存储结构

双亲表示法

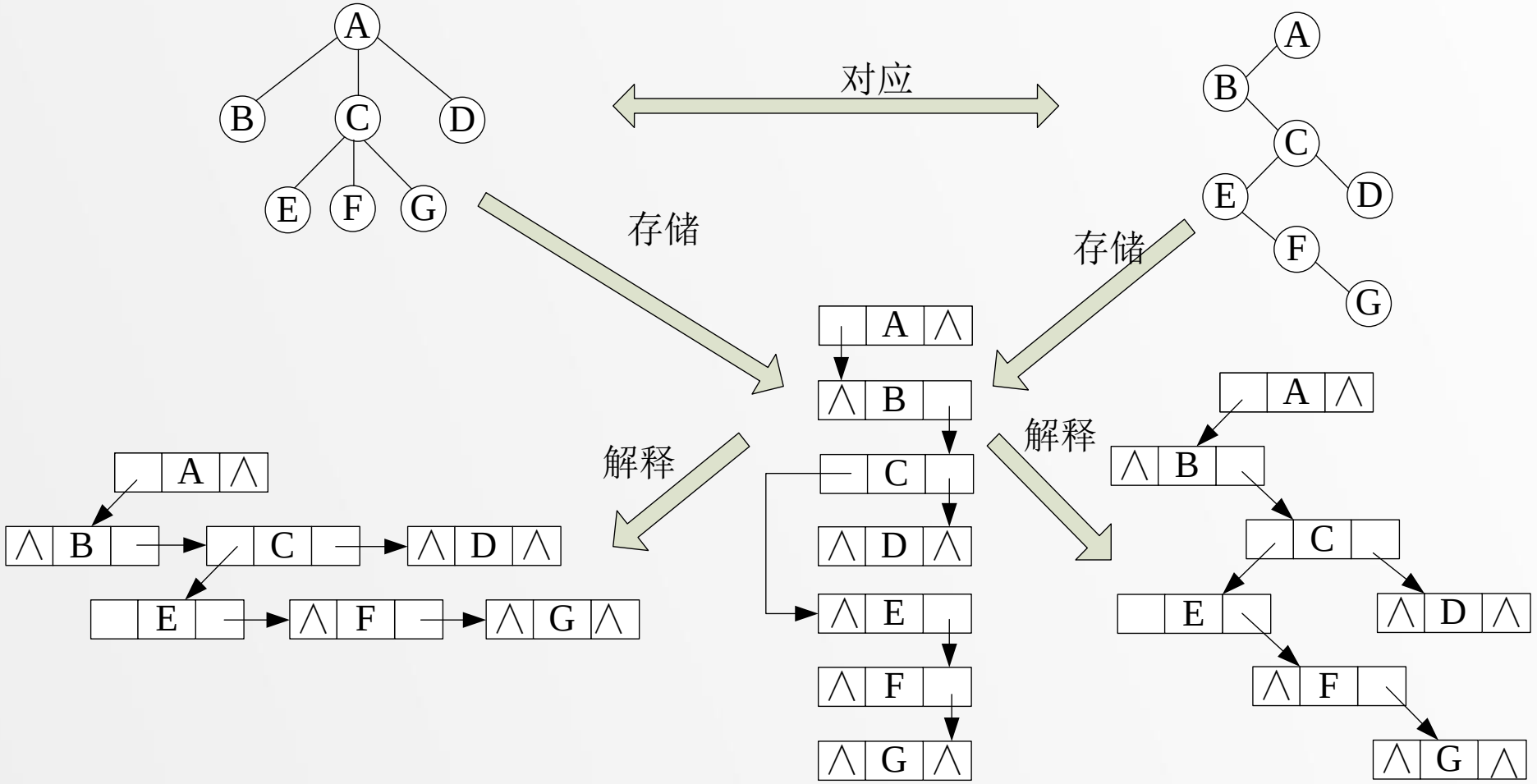
孩子表示法

孩子兄弟表示法

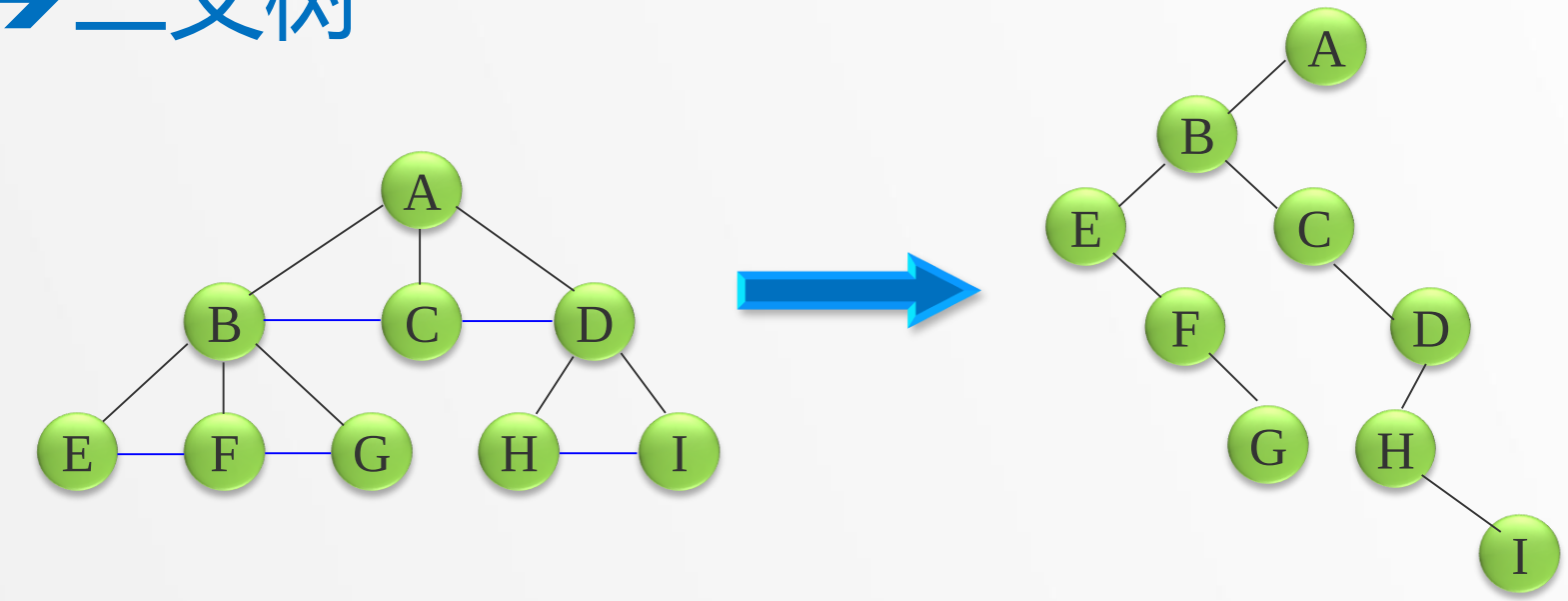
4.3.3 树和森林的基本操作

树及森林和二叉树的相互转换

树及森林和二叉树的相互转换

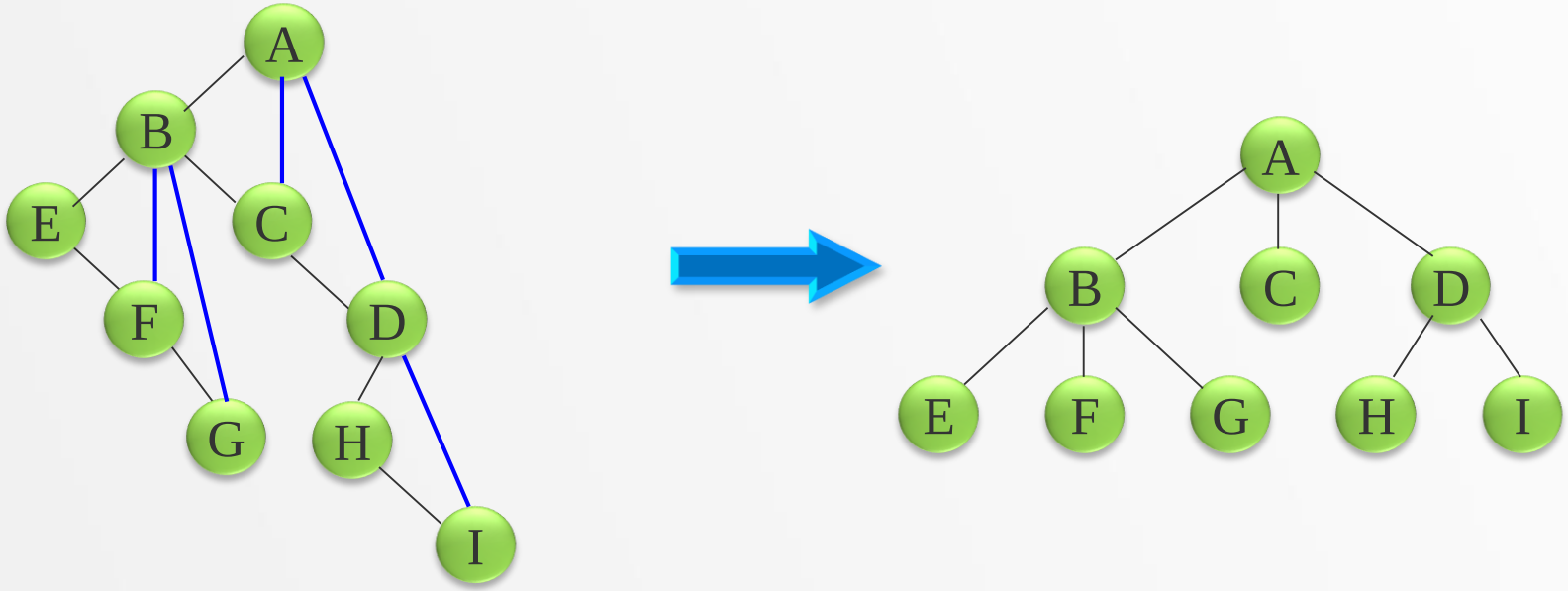


树➔二叉树

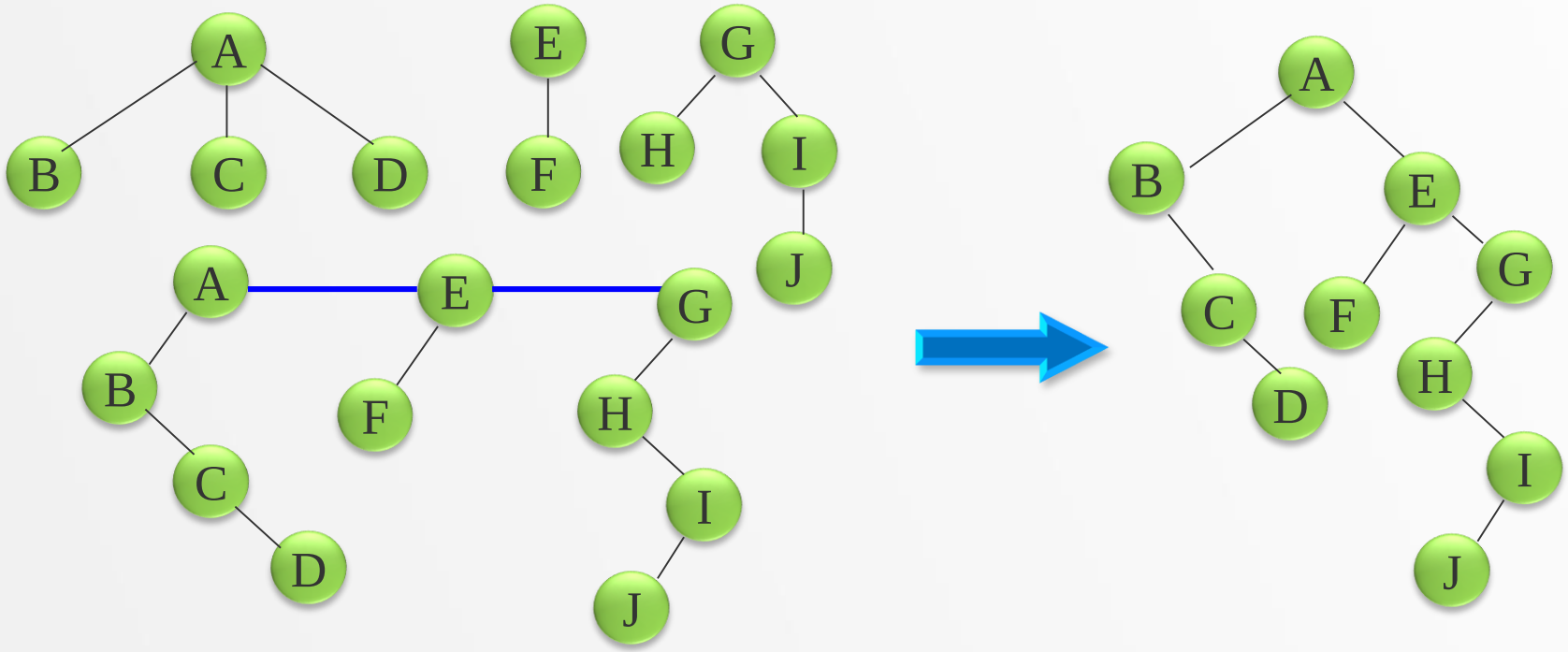


树转换成的二叉树其根结点的右子树一定为空

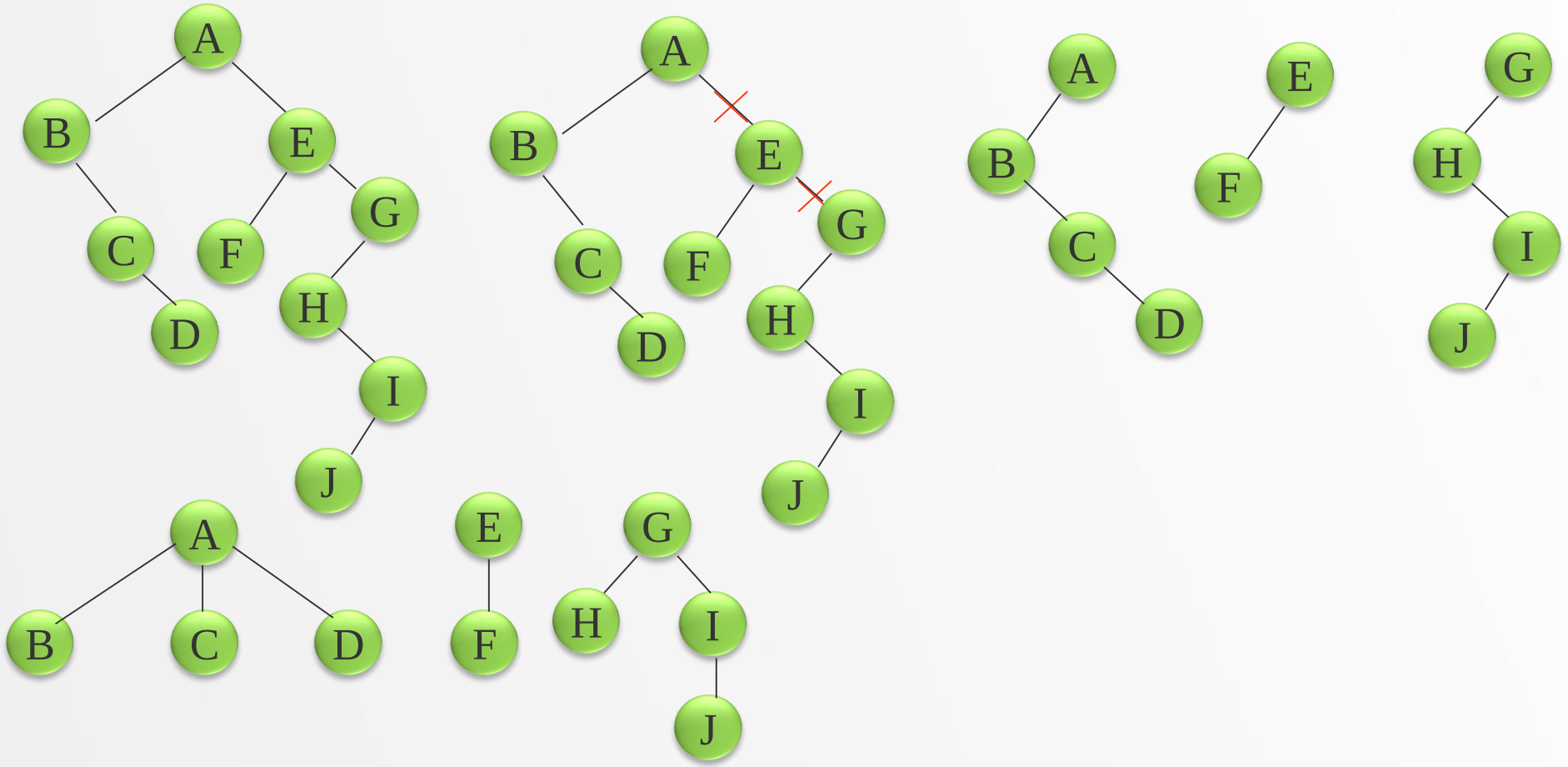
树➔二叉树



森林➔二叉树



二叉树➔森林



4.3.3 树和森林的基本操作

2. 树的遍历

遍历方法

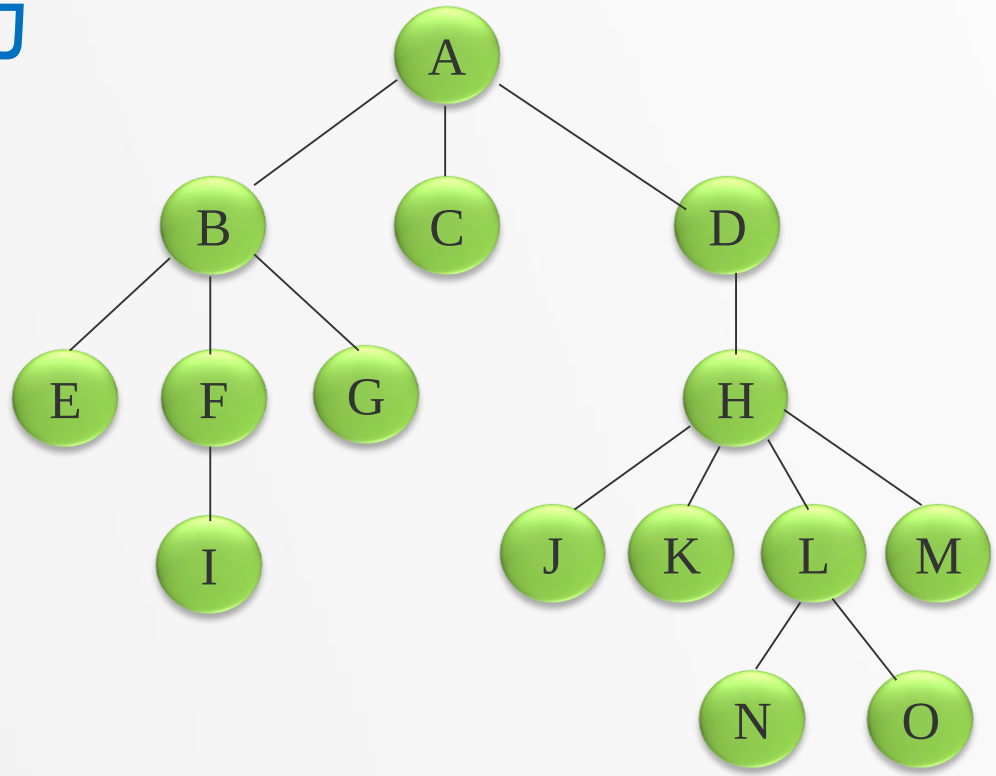
- 先根（序）遍历
- 后根（序）遍历
- 层次遍历

讨 论

为什么树没有中序遍历算法?

2.树和森林的遍历

树的遍历



先序遍历： A B E F I G C D H J K L N O M

后序遍历： E I F G B C J K N O L M H D A

层次遍历： A B C D E F G H I J K L M N O

2. 树和森林的遍历

森林的遍历

–先序遍历

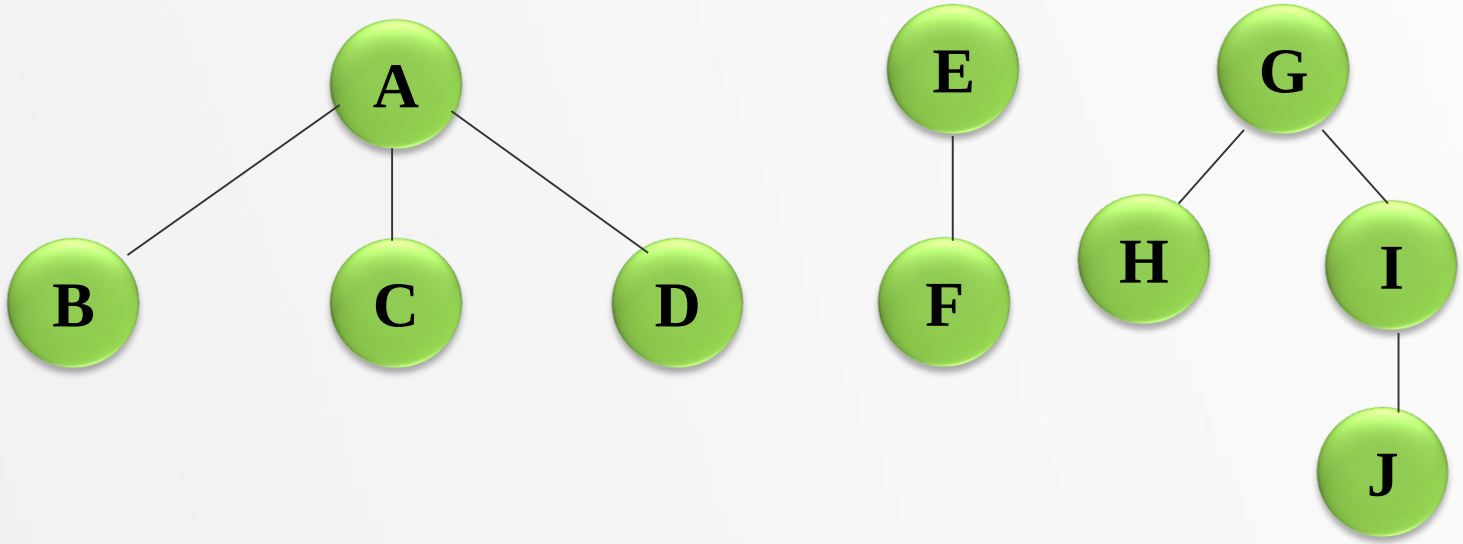
–中序遍历

讨 论

为什么森林没有后序遍历算法?

2. 树和森林的遍历

森林的遍历



先序遍历：A B C D E F G H I J

中序遍历：B C D A F E H J I G

作业:

将下列树转化成二叉树: