

树的双亲表示法、孩子表示法和孩子兄弟表示法

在使用[树](#)结构描述实际问题时，大多数不是[二叉树](#)，更多的是普通的树结构，在存储之间具有普通树结构的数据时，经常使用的方法有3种：

- 1. [双亲表示法](#)
- 2. [孩子表示法](#)
- 3. [孩子兄弟表示法](#)

双亲表示法

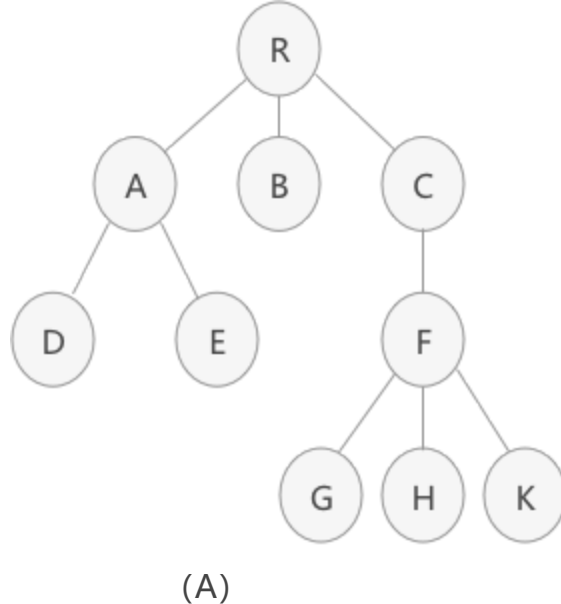
取一块连续的内存空间，在存储每个结点的同时，各自都附加一个记录其父结点位置的变量。

在树结构中，除了树根外，每个结点都只有一个父结点（又叫“**双亲结点**”）。

代码表示：

```
01.  #define tree_size 100//宏定义树中结点的最大数量
02.  #define TElemType int//宏定义树结构中数据类型
03.  typedef struct PTNode{
04.      TElemType data;//树中结点的数据类型
05.      int parent;//结点的父结点在数组中的位置下标
06.  }PTNode;
07.  typedef struct {
08.      PTNode nodes[tree_size];//存放树中所有结点
09.      int r,n;//根的位置下标和结点数
10.  }PTree;
```

例如，使用双亲表示法存储[图 1](#)（A）中的树结构时，数组存储结果为（B）：



数组下标 data parent

0	R	-1
1	A	0
2	B	0
3	C	0
4	D	1
5	E	1
6	F	3
7	G	6
8	H	6
9	K	6

(A)

(B)

图 1 双亲表示法

当算法中需要在树结构中频繁地查找某结点的父结点时，使用双亲表示法最合适。当频繁地访问结点的孩子结点时，双亲表示法就很麻烦，采用孩子表示法就很简单。

孩子表示法

将树中的每个结点的孩子结点排列成一个线性表，用链表存储起来。对于含有 n 个结点的树来说，就会有 n 个单链表，将 n 个单链表的头指针存储在一个线性表中，这样的表示方法就是孩子表示法。

如果结点没有孩子（例如叶子结点），那么它的单链表为空表。

代码表示：

```
01. #define TElemType int
02. #define Tree_Size 100
03. //孩子表示法
04. typedef struct CTNode{
05.     int child;//链表中每个结点存储的不是数据本身，而是数据在数组中存储的位置下标
06.     struct CTNode * next;
07. }*ChildPtr;
08. typedef struct {
09.     TElemType data;//结点的数据类型
10.     ChildPtr firstchild;//孩子链表的头指针
11. }CTBox;
12. typedef struct{
13.     CTBox nodes[Tree_Size];//存储结点的数组
14.     int n,r;//结点数量和树根的位置
15. }CTree;
```

例如，使用孩子表示法存储图 1 （A）， 存储效果如图 2：

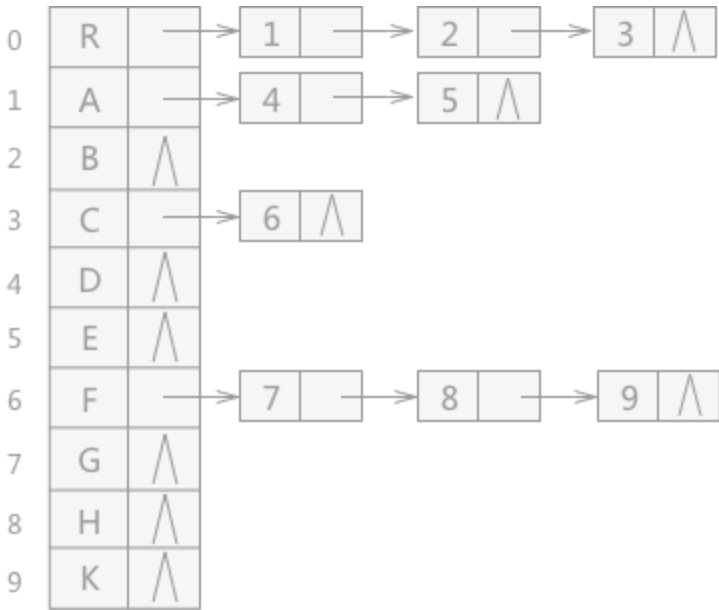


图 2 孩子表示法

使用孩子表示法存储的树结构，正好和双亲表示法相反，适用于查找某结点的孩子结点，不适用于查找其父结点。可以将两种表示方法合二为一， 存储效果如图 3：

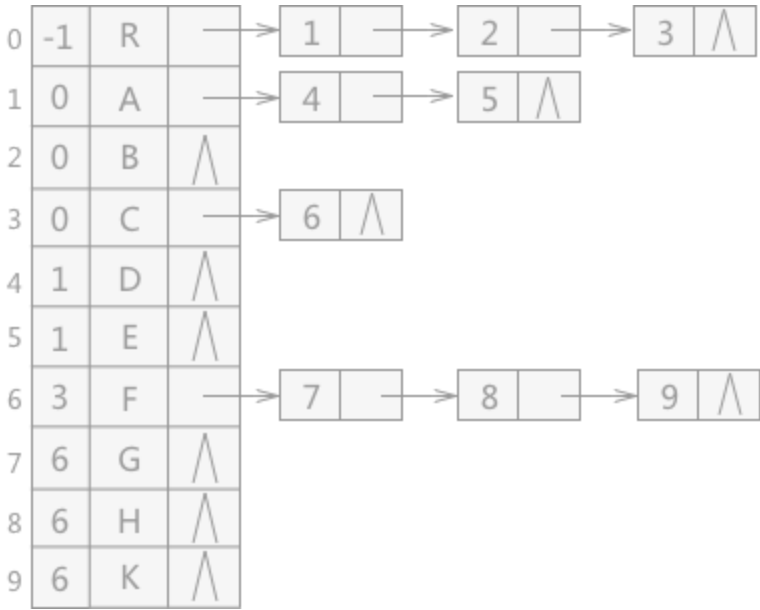


图 3 孩子双亲表示法

孩子兄弟表示法

使用链式存储结构存储普通树。链表中每个结点由 3 部分组成：



图 4 结点构成

其中孩子指针域，表示指向当前结点的第一个孩子结点，兄弟结点表示指向当前结点的下一个兄弟结点。

代码表示：

```
01.  #define ElemType int
02.  typedef struct CSNode{
03.      ElemType data;
04.      struct CSNode * firstchild,*nextsibling;
05.  }CSNode,*CSTree;
```

通过孩子兄弟表示法，普通树转化为了二叉树，所以孩子兄弟表示法又被称为 “**二叉树表示法**” 或者 “**二叉链表表示法**” 。

例如，用孩子兄弟表示法表示图 1 （A）的普通树，存储结果为：

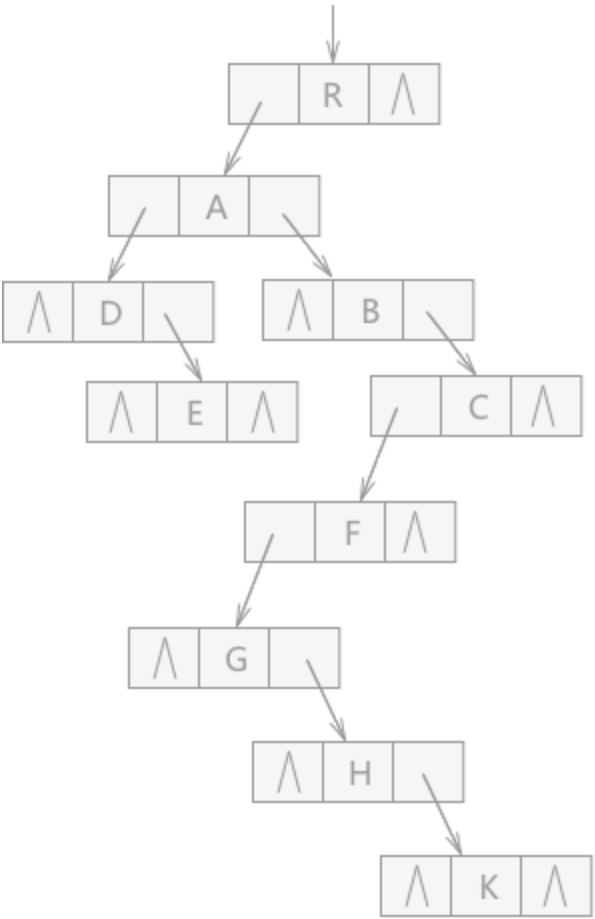


图 5 二叉链表表示法

补：森林和二叉树的相互转化

通过孩子兄弟表示法的学习，对于任意一棵树，都可以找到唯一的一棵二叉树与之对应。

普通树转化成的二叉树，其根结点都没有右孩子，即普通树对应的二叉树肯定没有右子树。

而森林是由多棵树组成，为了便于对森林的遍历等操作，需要将森林中的所有树都组合成一颗大的二叉树，转化步骤为：

- 1. 首先将森林中树各自转化为二叉树；
- 2. 森林中第一棵二叉树的树根作为转化后二叉树的树根；
- 3. 其他树的树根作为第一棵树树根的兄弟结点，进行连接；

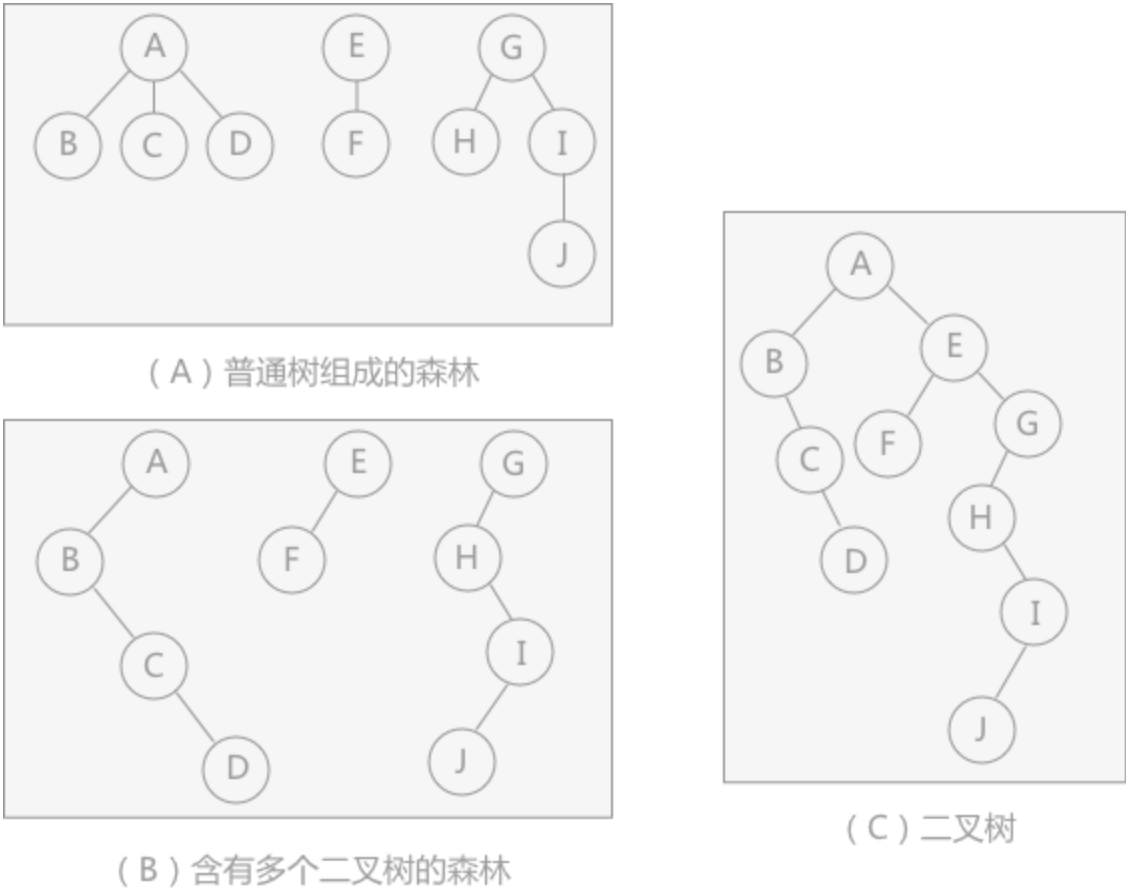


图 6 森林转化成二叉树

如图 6 所示，（A）中由三棵普通树组成的森林，首先三棵普通树采用孩子兄弟表示法各自转化成二叉树，如（B）所示；然后由（B）转（C）时，将森林中第一棵树的树根作为转化后的整棵二叉树的树根，其他数的树根作为第一棵树的树根的兄弟结点，如（C）所示。

转化成二叉树的森林，做的最多的操作就是查找树中的结点。在遍历转化后的二叉树时，遍历方式有先序遍历、中序遍历和后序遍历。

对森林使用先序和中序遍历的结果和对转化后的二叉树使用先序和中序遍历得到的序列是一样的，而使用后序遍历得到的结果不同。例如图 6（B）中森林采用中序遍历和（C）中二叉树采用中序遍历得到的结果是相同的，遍历序列都为：BCDAFEHJIG。

总结

树的三种表示方法中，双亲表示法和孩子表示法在实际算法中的应用场景正好相反：双亲表示法应用于解决查找某结点的父结点，而孩子表示法应用于查找某结点的孩子结点。

孩子兄弟表示法可以将普通树转化成二叉树存储，在实际操作中，可以应用二叉树的性质来解决普通树或者森林的问题。

联系方式 **购买教程（带答疑）**