一.字符串

1.串类型定义

串(String)是由零个或多个字符组成的有限序列。一般记作S=‘a1a2a3…an’，其中S 是串名，单引号括起来的字符序列是串值；

ai(1≦i≦n)可以是字母、数字或其它字符；串中所包含的字符个数称为该串的长度。

空串(Empty String)：长度为零的串。它不包含任何字符。

子串：串中任意个连续字符组成的子序列。

主串：包含子串的串。通常将子串在主串中首次出现时的该子串的首字符对应的主串中的序号，定义为子串在主串中的序号（或位置）。

串的基本操作集：StrLength,StrAssign,SteConcat,SubStr,StrCmp,StrIndex,StrInsert,StrDelete,StrRep.

2.串的表示和实现

（1）串的定长顺序存储结构

#define MAXSIZE 256

char s[MAXSIZE];

typedef struct

{ char data[MAXSIZE];

int curlen;

} SeqString;

SeqString s;

3.串的链式存储结构（带头结点的单链表）

结点的类型定义：

Typedef struct node

{ char data; //存放字符

Struct node \* next; //指针域

}Linkstring；

4.链式串基本运算算法

（1）赋值运算

将一个C/C++字符数组t赋给串s.

Void StrAssign (LinkString \* &s, char t[])

{ int i= 0;

LinkString \*q, \* tc;

s= (Linkstring \* ) malloc (sizeof (LinkString)); //建立头结点

s- > next = NULL; //初始化链表指针

tc= s;

while (t[i]!= '\0') //将整个串逐个字符地申请结点并建立链表及相应的值

{ q= (LinkString \*)malloc (sizeof(LinkString));

q-> data= t[i];

tc- > next= q; tc= q;

i++ ;

}

tc-> next = NULL;

}

(2)串连接运算

将串t连接到串s之后,返回其结果。

LinkString \* Concat( LinkString\* s, LihkString \* t)

{ LinkString \* p= s-> next, \* q, \* tc,\* r;

r= (LinkString \* )malloc (sizeof(LinkString)); //建立头结点

r- > next = NULL;

tc= r; //tc总是指向新链表的最后-个结点

while (p!= NULL) //将s串复制给r

{ q= (LinkString \* ) malloc (sizeof (LinkString));

q-> data= P-> data;

tc->next=q;tc=q;

P= P- > next;

}

p= t- > next;

while (p!= NULL) //将t串复制给r

{ q= (LinkString兴)malloc (sizeof (LinkString) ) ;

q-> data= p-> data;

tc-> next= q; tc= q;

P= P- > next;

}tc- > next = NULL;

return(r) ;

}

二.树

1.树的定义

树是由n（n>=0）个结点的有穷集合，其中：（1）每个元素称为结点(node)；（2）有一个特定的结点被称为根结点或树根（root);

(3)除根节点以外的其余数据元素被分为m个互不相交的集合T1，T2...Tm-1,其中每一个集合Ti本身也是一颗树，被称作原树的字树（subtree).

2.树的基本术语

结点---表示树中的元素，包括数据项及若干指向其子树的分支.

结点的度---结点拥有的子树数.

叶子---度为0的结点， 也叫终端结点.

分支结点---度不为0的结点，也叫非终端结点.

内部结点--- 除根结点之外， 分支结点也称为内部结点.

树的度---一棵树中最大的结点度数.

孩子---结点子树的根称为该结点的孩子.

双亲---孩子结点的上层结点叫该结点的双亲.

兄弟---同一双亲的孩子之间互成为兄弟.

祖先--- 结点的祖先是从根到该结点所经分支上的所有结点.

子孙--- 以某结点为根的子树中的任一结点都成为该结点的子孙.

结点的层次---从根结点算起，根为第一层， 它的孩子为第二层...

深度---树中结点的最大层次数.

有序树---如果将树中结点的各子树看成从左至右是有次序的(即不能互换)，则称该树为有序树,否则称为无序村。在有序树中最左边的子树的根称

为第一一个孩子，最右边的称为最后一个孩子.

森林---m(m>=0)棵互不相交的树的集合.

3.树的基本操作

(1)双亲结点parent(2)左孩子结点leftChild(3)右兄弟结点rightSibling(4)遍历树traverse(vs)

4.树的存储结构

（1）树的存储表示方法

双亲表示法，孩子表示法，双亲孩子表示法，孩子兄弟表示法等。

孩子兄弟表示法存储结构：

typedef struct TreeNode{

elemtype data;

struct TreeNode \* son;

struct TreeNode \* next;

}NodeType;

5.树的遍历

（1）先根遍历

树的先根遍历递归算法为:

1)访问根结点;2)按照从左到右的次序先根遍历根结点的每一棵子树。

（2）后根遍历

树的后根遍历递归算法为:

1)按照从左到右的次序后根遍历根结点的每一棵子树;2) 访问根结点。

（3）层次遍历

从树的第一层（根节点）开始，从上至下逐层遍历，在同一层中，则按从左到右的顺序对结点逐个访问。