

#### 算法 D（数字树形查找）

//给定一个文件，它以数字“0 < 1”为序形成一株二进制数字查找二叉树形。本算法查找一个给

//定的变元 k，若 k 不在文件中，则把一个包含关键词 k 的新结点插入到树形的适当位置

DSTNode\* DSTree::Search\_Insert(char k[],int n) //为简便起见，用数组 k 表示二进制位,n 为 k 的长度

```
{
    int i;
    char b;
    if(root==NULL)
    {
        root=new DSTNode(k,NULL,NULL);
        return NULL;
    }
    DSTNode *p=root;
    char *k1=new char[n];
    strcpy(k1,k);
    while(p!=NULL)
    {
        if(equal(k,p->key,n)==true) return p;    //查找成功,equal 判断两个二进制串是否相等
        else
        {
            b=k1[0];
            for(i=1;i<n;i++) k1[i-1]=k1[i];    //左移一位
            k1[n-1]='0';
            if(b=='0')
            {
                if(p->llink!=NULL) p=p->llink;
                else break;
            }
            else
            {
                if(p->rlink!=NULL) p=p->rlink;
                else break;
            }
        }
    }
    DSTNode *q=new DSTNode(k,NULL,NULL);
    if(b=='0') p->llink=q;
    else p->rlink=q;
    return NULL;
}
```