

第四章作业答案

4.17

```
Status Replace(SString &s,SString T,SString V){
    //用 V 替换主串 S 中出现的所有与 T 相等的非重叠的字串
    for(int i=1;i<=S[0]-T[0]+1;i++){
        while(k<=T[0]&&S[j]==T[k]){j++;k++;} //匹配
        if(k>T[0]){
            if(T[0]==V[0]){//两个子串长度相等
                for(int m=1;m<=T[0];m++)
                    S[i+m-1]=V[m];
            }//两个子串长度不等
            else if(T[0]<V[0]){
                for(int m=S[0]-V[0]+T[0];m>=i+T[0];m--)
                    S[m+V[0]-T[0]]=S[m];
                for(int m=1;m<=V[0];m++)
                    S[i+m-1]=V[m];
            }
            else{
                for(int m=i+V[0];m<=S[0]+V[0]-T[0];m++)
                    S[m]=S[m-V[0]+T[0]];
                for(int m=1;m<=V[0];m++)
                    S[i+m-1]=V[m];
            }
            S[0]=S[0]+V[0]-T[0]; //修改长度
            i+=V[0];
        }
    }
    return OK;
}
//没有考虑溢出问题，如果感兴趣，自己判断下
//这题有的同学用 Index(),StrDelete()跟 StrInsert()等函数来写，用的对的话也没算错。
```

4.21

//结点大小为 1（且附头结点）的链表结构表示串。
//实现串的基本操作 StrAssign,StrCopy,StrCompare,StrLength,Concat,SubString 的函数
//这题写法比较多，思路正确的都可以

```
typedef struct{
    char ch;
    struct LNode *next;
}LNode,*LString;
```

```

Status StrAssign(LString &S, char *chars){
    //生成一个其值等于 chars 的串 S
    LString p,q;
    int len;
    char *c=chars;
    p=S->next;
    while(p){
        q=p;p=p->next;free(q);
    }
    for(len=0;*c!='\0';c++,len++);
    S=(LString)malloc(sizeof(LSNode));
    S->next=NULL;
    q=S;
    for(int i=0;i<len;i++){ //生成
        p=(LString)malloc(sizeof(LSNode));
        p->ch=chars[i];
        p->next=q->next;
        q->next=p;
        q=p;
    }
    return OK;
}

```

```

Status StrCopy(LString &T,LString &S){
    //由串 S 复制得串 T
    LString p,q,r;
    p=T->next;
    while(p){
        q=p;p=p->next;free(q);
    }
    T=(LString)malloc(sizeof(LSNode));
    T->next=NULL;
    p=S;
    r=S->next;
    while(r){ //复制
        q=(LString)malloc(sizeof(LSNode));
        q->ch=r->ch;
        q->next=p->next;
        p->next=q;
        p=q;
        r=r->next;
    }
    return OK;
}

```

}//和 StrAssign 类似

```
int StrCompare(LString S,LString T){
    //比较两个串, 若 S>T, 返回 1; 若 S=T, 返回 0; 若 S<T, 返回 -1;
    LString p,q;
    p=S->next;
    q=T->next;
    if(p==NULL&&q==NULL)
        return 0;
    while(p!=NULL&&q!=NULL){
        if(p->ch!=q->ch) //不相等
            return p->ch - q->ch;
        p=p->next;
        q=q->next;
    }
    //两个串长度不同
    if(p!=NULL)
        return 1;
    else if(q!=NULL)
        return -1;
    else
        return 0;
}
```

```
int StrLength(LString S){
    //返回串 S 的长度
    int len=0;
    LString p;
    p=S->next;
    while(p!=NULL){
        len++;
        p=p->next;
    }
    return len;
}
//有的同学直接写 S->curlen 或者 Length(S)
//考查的是基本操作的具体实现
```

```
Status Concat(LString &T,LString S1,LString S2){
    //用 T 返回由 S1 和 S2 联接而成的新串, 串 S1,S2 存在
    LString p,q;
    p=T->next;
```

```

while(p){
    q=p;p=p->next;free(q);
}
T=(LString)malloc(sizeof(LSNode));
p=S1->next;
T->next=S1->next;
while(p->next)
    p=p->next;
p->next=S2->next;
return OK;
}

```

书上的函数定义里，串 S1、S2 存在，考虑了不存在情况的也正确

```

Status SubString(LString &Sub,LString S,int pos,int len){
    //用 Sub 返回串 S 的第 pos 个字符起长度为 len 的子串
    LString p,q,r;
    p=Sub->next;
    while(p){
        q=p;p=p->next;free(q);
    }
    Sub=(LString)malloc(sizeof(LSNode));
    if(!len)
        Sub->next=NULL;
    else{
        p=S->next;
        q=Sub;
        for(int i=1;i<pos;i++){
            p=p->next;
        }
        for(int i=1;i<=len;i++){
            r=(LString)malloc(sizeof(LSNode));
            r->ch=p->ch;
            r->next=q->next;
            q->next=r;
            q=r;
            p=p->next;
        }
    }
    return OK;
}

```

4.23

以块链结构表示串。
 判别给定串是否具有对称性。要求时间复杂度为 $O(\text{StringLength}(S))$

```

typedef struct Chunk{

```

```

    char ch[CHUNKSIZE];
    struct Chunk *next;
}Chunk;

typedef struct{
    Chunk *head,*tail; //串的头和尾指针
    int curlen; //串的当前长度
}LString;

Status Judge(LString T){
    if(!(T	curlen))
        return OK;
    Stack S;
    Initstack(&S);
    Chunk *p;
    p=T.head;
    int i,j,e;
    int m=T	curlen/2;
    for(i=1,j=0;i<=m;i++){ //进栈
        push(&S,p->ch[j])
        j=(j+1)%CHUNKSIZE;
        if(!j)
            p=p->next;
    }
    if(T	curlen%2){//如果串的长度为奇数
        i++;
        j=(j+1)%CHUNKSIZE;
        if(!j)
            p=p->next;
    }
    while(!StackEmpty(S)&&i<=T	curlen){ //比较
        pop(&S,&e);
        if(e!=p->ch[j])
            return ERROR;
        i++;
        j=(j+1)%CHUNKSIZE;
        if(!j)
            p=p->next;
    }
    if(StackEmpty(S)&&i>T	curlen)
        return OK;
}

```

//这题有的同学没有考虑串的长度为奇数，直接当偶数做的。

