

数据结构作业(第四次)

PB20111686 黄瑞轩

4.22

块链结构定义如下，这里默认头尾都储存信息。

```
#define CHUNKSIZE 80

class Chunk {
    char ch[CHUNKSIZE];
    Chunk* next;
}

class LString {
    Chunk* head, * tail; //串的头尾指针
}
```

算法实现如下：

```
int PosLocated(Chunk* ck, char c) {
    // 在块ck中寻找字符c，并返回位置索引，没找到就返回-1
    for (int j = 0; j < CHUNKSIZE; j++) {
        if (ck->ch[j] == c) {
            return j;
        }
        else if (ck->ch[j] == '#') {
            return -1;
        }
    }
}

void Insert(LString* t, LString* s, char c) {
    // 在t中寻找字符c，并把s插在c后面
    Chunk* t_trav = t->head;
    // t的块选择指针
    int where = -1;
    for (; t_trav != t->tail; t_trav = t_trav->next) {
        where = PosLocated(t_trav, c);
        if (where != -1)break;
    }
    // 上面这个检查从head到tail-1的所有块，如果都没找到最后t_trav指向tail
    if (where == -1) {
        // 如果都没找到，说明where仍是-1，再找tail
        where = PosLocated(t_trav, c);
    }
    if (where == -1) {
        // 说明都没找到，直接拼接
        t->tail->next = s->head;
        return;
    }
    else {
        // 此时需要把c所在的块拆成两半
        Chunk* newChunk = new Chunk;
        for (int j = 0; j <= where; j++) {
            newChunk->ch[j] = '#';
        }
        // 把前半部分置'#'
        for (int j = where + 1; j < CHUNKSIZE; j++) {
            newChunk->ch[j] = t_trav->ch[j];
        }
        // 后半部分复制t_trav的后半部分
        for (int j = where + 1; j < CHUNKSIZE; j++) {
            t_trav->ch[j] = '#';
        }
        // t_trav的后半部分置'#'
        s->tail->next = newChunk;
        newChunk->next = t_trav->next;
        t_trav->next = s->head;
        return;
    }
}
```

