

## 问题:

实现线性表的带表头结点的单链表存储结构,并分析每种基本操作(算法)的时间复杂度。

## Code

```
struct celltype
{
    char date;
    celltype *next;
};
typedef celltype *LIST;
typedef celltype *position;
position End(LIST L)
{
    position q;
    q = L;
    while(q -> next != NULL)
    {
        q = q -> next;
    }
    return q;
}
void Insert(char x, position p)
{
    position temp;
    temp = p -> next;
    p -> next = new celltype;
    p ->next -> date = x;
    p -> next -> next = temp;
}
void Delete(position p)
{
    position q;
    if(p -> next != NULL)
    {
        q = p -> next;
        p -> next = q -> next;
        delete q;
    }
}
position Locate(char x, LIST L)
{
    position q;
```

```

position p;
p = L;
while(p -> next != NULL)
{
    if(p -> next -> date == x)
    {
        return p;
    }
    else
    {
        p = p -> next;
    }
}
return p;
}
position MakeNull(LIST &L)
{
    position p;
    p -> next = NULL;
    return L;
}

```

### 时间复杂度分析:

1. **position End(LIST L):** 做循环, 知道 P->NEXT 为空停止, 循环里面的操作为基本操作, 所以最坏的时间复杂度为  $O(N)$ .
2. **void Insert(char x, position p):** 直接做相应的插入操作, 所以最大的时间复杂度为  $O(1)$ .
3. **void Delete(position p):** 直接做相应的删除操作, 所以最大的时间复杂度为  $O(1)$ .
4. **position Locate(char x, LIST L):** 直接在循环里做基本操作, 所以最坏的时间复杂度为  $O(N)$ .