

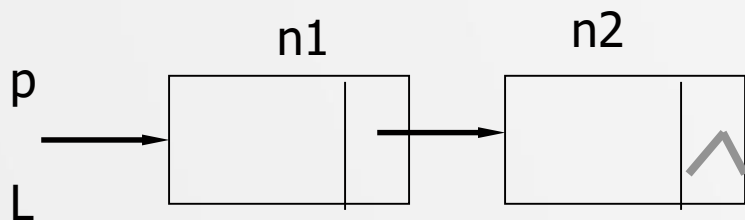
二、单链表的C语言实现

类型定义

```
typedef struct Node
{
    ElemType elem;
    struct Node *next;
}Node,*Ptr;
typedef Ptr *SqListPtr;
```

变量定义和使用

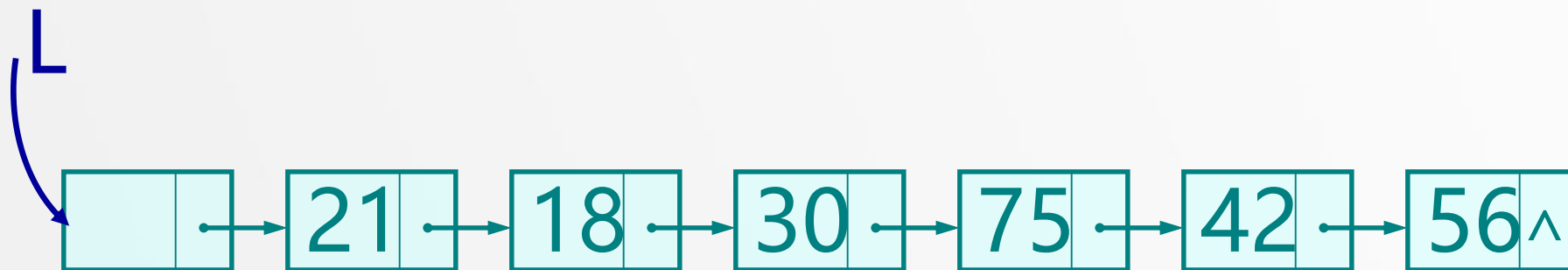
```
Node n1, n2;
/* 定义2个结点变量*/
Ptr p = &n1;
/* 定义一个指向结点的指针变量p, 并存放n1的地址(指针) */
n1.next=&n2; /* 结点n1的指针域存放结点n2的地址 */
SqListPtr L=p; //定义一个单链表L
N2.next=NULL;
```



三、单链表基本操作及实现

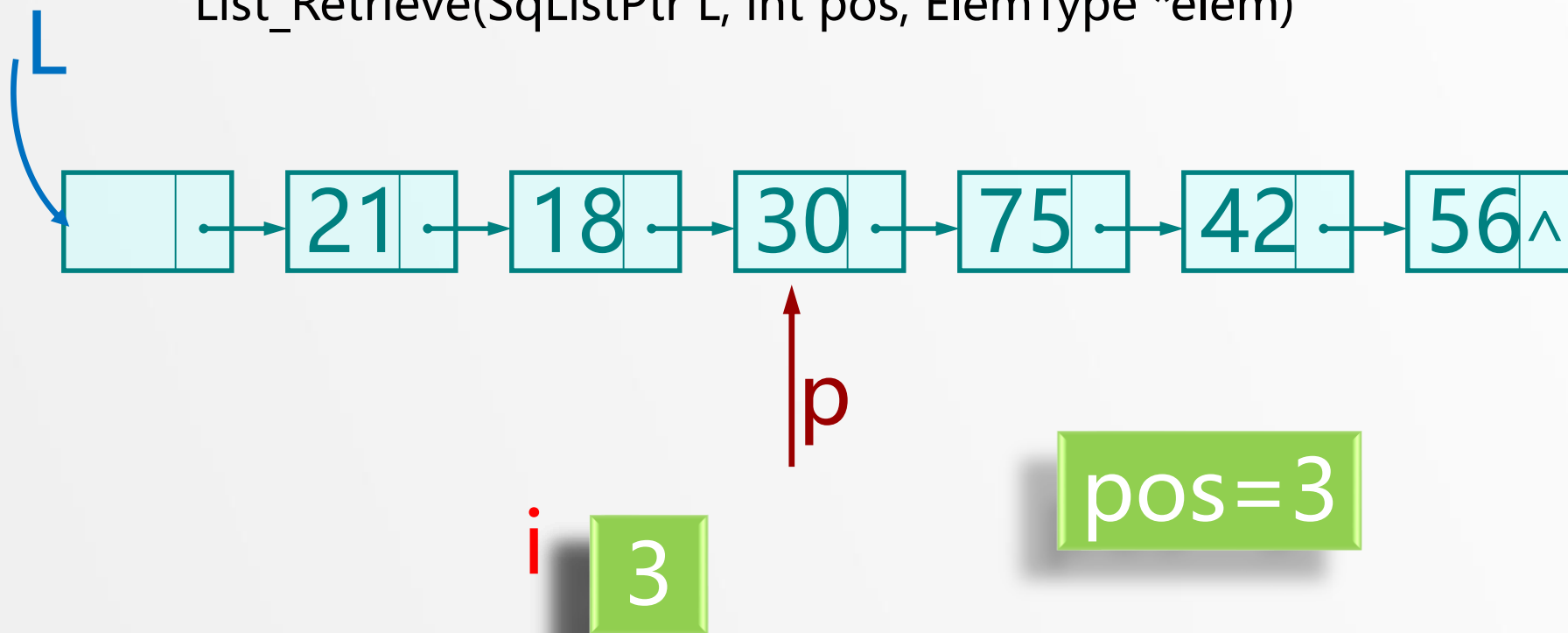
①查找——按位置查找

- 问题：在给定的带头结点的单链表L中，查找指定位置的数据元素，如果存在，则返回success，同时取回相应结点的数据
- 方法：链表的操作只能通过从头指针出发，顺着链域next逐个结点比较，直到搜索到指定位置的结点为止



在单链表中获取第pos个元素:

List_Retrieve(SqListPtr L, int pos, ElemType *elem)



三、单链表基本操作及实现

①查找——按位置查找

```
Status List_Retrieve(SqListPtr L, int pos, ElemType *elem){
```

置初值部分

```
while (循环控制条件)
```

```
{ 循环体 }
```

出循环后找到的判定条件及相应处理

```
}
```

三、单链表基本操作及实现

①查找——按位置查找

```
Status List_Retrieve(SqListPtr L, int pos, ElemType *elem){
```

置初值部分

```
while (循环控制条件)
```

```
{ 循环体 }
```

出循环后找到的判定条件及相应处理

```
}
```

问题：如何置初值？

怎样对pos的合法性进行检查？

找到的判定条件？

出循环的判断条件是 $i == pos$ 且p指针指向结点不空

三、单链表基本操作及实现

```
Status List_Retrieve(SqListPtr L, int pos, ElemType *elem)
{
    Status s = range_error;
    Ptr p = (*L)->next;; /* 带头结点, 移动p指向第一个元素结点*/
    int i = 1; /*计数器*/
    while (p && i < pos){ /* p指向的结点存在,且未到达指定位置*/ /*条件1 防止pos > 表长, 条件2
    控制取第pos个, 防pos < 1 */
    {
        i++;
        p = p->next;
    }
    if (p && i == pos) /*找到指定位置,且该结点存在*/
    {
        *elem = p->data;
        s = success;
    }
    return s;
}
```

初始化:
设置状态变量**status**
设置移动指针p
计数变量i

如果只用:
if (**i==pos**){
 *elem = p->data;
 status = success;
}

行不行?

如果用:
if (**p=NULL**)
 status = **range_error**;
else{*elem = p->data;
 status = success;}

又行不行?

①查找——按位置查找 P.30 算法2.7

循环条件分析:

```
p = (*L)->next; i = 1;
```

```
while (p && i < pos) { i++; p = p->next; }
```

条件 1 防止 $\text{pos} > \text{表长}$, 条件 2 控制取第 pos 个, 并防止了 $\text{pos} < 1$

两个条件有 6 种组合:

1. $p == \text{NULL} \ \&\& \ i < \text{pos}$ 空表且 $\text{pos} > 1$ 或 $\text{pos} > \text{表长} + 1$, `range_error`
2. $p == \text{NULL} \ \&\& \ i == \text{pos}$ 空表且 $\text{pos} = 1$ 或 $\text{pos} = \text{表长} + 1$, `range_error`
3. $p == \text{NULL} \ \&\& \ i > \text{pos}$ 空表且 $\text{pos} < 1$, `range_error`
4. $p \neq \text{NULL} \ \&\& \ i < \text{pos}$ 继续循环
5. $p \neq \text{NULL} \ \&\& \ i == \text{pos}$ 确定第 pos 个结点, 正常出循环, `success`
6. $p \neq \text{NULL} \ \&\& \ i > \text{pos}$ $\text{pos} < 1$, `range_error`

三、单链表基本操作及实现

①查找——按值查找

- 算法思想描述：
 1. 初始化：P指针指向线性表第一个结点；位置 $i=1$ ；操作状态s为错误
 2. 只要线性表不空，循环做下面操作
 3. 如果待查找数据==当前结点数据，则退出循环；
 4. 否则p指针指向下一个结点；位置计算器i增1
 5. 循环外面判断 $p \neq \text{NULL}$ ，则找到数据，将位置i赋值给pos参数；修改操作状态s为成功；
 6. 返回操作状态s

三、单链表基本操作及实现

①查找——按值查找

```
Status List_Locate(SqListPtr L, ElemType elem, int *pos)
{
    Status s = range_error;
    Ptr p = (*L)->next;
    int i = 1;
    while (p != NULL){
        if (p->elem == elem)break;
        i++;
        p = p->next;
    }
    if (p){
        *pos = i;
        s = success;
    }
    return s;
}
```

也可用复合条件将循环改写为：

```
while(p && p->data != elem ){
    i++;
    p=p->next;
}
```