



# 中级篇

## 第四部分：链表之江湖论剑

链结点定义同前

对链表的假设：链表由 head 指针管理，可空。链表待添加新元素 value 由 Nnode 申请空间，( 即 `node* Nnode=new node; Nnode->data=value; Nnode->next=nullptr;`  )，待删除元素 value

与数组的工作做比较，讨论添加删除的各种情况。

对数组的假设：数组 a，容量 n，元素个数 t ( $t < n$ )，待添加元素 value，待删除元素 value

### 一、不考虑数据集中数据的特征，添加元素 value

#### 1) 数组的快速添加：添加到数组 a 尾部

```
if(t<N) {
    a[t]=value;
    t++; //集合元素个数增加
}
else cout<<"放不下"; //后面不再重复讨论空间的满
```

对比：链表添加到尾部

```
if(head){
    node *tail=head;
    while(tail->next) tail=tail->next;
    tail->next= Nnode;
}
else{//空表
    head= Nnode;
}
```

#### 2) 链表的快速添加：添加到链表 head 头部

```
//node* Nnode;
// Nnode->data=value; Nnode->next=nullptr;
Nnode->next=head;
head= Nnode;
```

对比：数组添加到头部

```
for(int i=t-1;i>=0;--i) a[i+1]=a[i];
a[0]=value;
t++;
```

### 二、假设数据集有升序特征（数据从小到大排列），考虑有序添加

#### 1) 数组的有序添加：寻找第一个比添加值小的元素所在的位置，添加在其后

```
i=t-1;
while(i>=0&& a[i]>value)
{
    a[i+1]=a[i];
    i--;
}
```

```

}
a[i+1]=value;
t++;

```

- 2) 链表的有序添加: 寻找第一个比添加值大的元素所在的位置, 添加在其前  
 node\*p,\*pre; //pre 为 p 前一个结点的指针, 两指针同步

```

p=head;
pre=nullptr;
while(p&& p->data<value) { pre=p; p=p->next; }
if(pre!=nullptr){ //pre 不是头结点
    Nnode ->next=p;
    pre->next= Nnode;
}
else //p 为头结点或者表为空
{
    Nnode ->next=head;
    head= Nnode;
}

```

说明: (前提: head==nullptr) 那么上述处理: Nnode->next=nullptr; head=Nnode;  
 等价于表为空的处理: head=Nnode;

### 三、添加元素到指定位置

- 1) 添加元素到数组 a 的 pos 位置, 假设 pose[0,t]

```

for(i=t-1;i>=pos;--i)
    a[i+1]=a[i];
a[pos]=value;
t++;

```

- 2) 添加元素到链表 pos 位置 (pos 为指向某结点的指针, 注意理解链表添加到该位置的概念, 新结点代替 pos 结点在链表中的逻辑位置)

```

if(pos==head) { //新结点要成为首结点
    Nnode ->next=pos; // Nnode ->next=head;
    head= Nnode;
}
else{ //定位 pre, 是 pos 的前一个结点
    node* pre=head;
    while(pre->next!=pos) pre=pre->next;
    pre->next= Nnode;
    Nnode ->next=pos;
}

```

**\*扩展: 若添加在 pos 后**

```

Nnode->next=pos->next;
pos->next=Nnode;

```

### 四、删除

删除的两种情况：1) 删除指定值；2) 删除指定位置的值。可以统一为第二种删除指定位置的值。即通过增加查找指定值，定位指定值所在位置。

接下来的讨论全部围绕删除指定位置的值展开。

- 1) 数组删除指定位置的值，假设位置为 `pos`，`pos` 的有效范围`[0,t-1]`，考虑和添加的差异。

```
for(int i=pos+1;i<=t-1;++i) a[i-1]=a[i];
```

```
--t; //集合规模缩小
```

- 2) 链表删除指定位置的值，假设位置为 `pos`

```
if(pos==head) //删除头结点
```

```
    head=head->next;
```

```
else{ //其他位置
```

```
    node* pre=head;
```

```
    while(pre->next!=pos) pre=pre->next; //寻找待删除位置的前一个结点
```

```
    pre->next=pos->next;
```

```
}
```

```
delete pos;
```