

选7、链表的概念与基本操作2–插入与删除...

讨论：如何在链表中插入节点

在单向链表中插入节点需要做两件事情：

- 1、存储链表节点数据，将该节点的后趋指针修改为前趋节点的后趋指针
- 2、将前趋节点的后趋指针指向该节点

1、在链表头部插入节点

- 1、存储链表节点数据，并将该节点的后趋指针修改为head头指针数据
- 2、将head头指针修改为该节点位置

2、在链表中部插入节点

- 1、存储链表节点数据，将该节点的后趋指针修改为前趋节点的后趋指针
- 2、将前趋节点的后趋指针修改为该节点位置

3、在链表尾部插入节点

- 1、存储链表节点数据，将该节点的后趋指针修改为结束标记
- 2、将前趋节点的后趋指针修改为该节点位置

4、在降序链表中插入n后，继续保持有序：

```
#插入n后链表继续有序
n=int(input("n:"))
ka=head_a
qa=head_a
while ka!=-1:
    if data_a[ka][0]>n:
        qa=ka
        ka=data_a[ka][1]
    else:
        if ka==head_a:
            data_a.append([n,head_a])
            head_a=len(data_a)-1
            qa=head_a
        else:
            data_a.append([n,ka])
            data_a[qa][1]=len(data_a)-1
            qa=data_a[qa][1]
        break
if ka==-1:
    data_a.append([n,-1])
    data_a[qa][1]=len(data_a)-1

show_chain(data_a,head_a)
```

5、删除链表元素

删除链表元素需要做几件事情：

1、将前趋节点的后继指针设置为待删除节点的后继指针

2、删除该节点

```
#删除第一个等于n的链表节点
n=int(input("n:"))
ka=head_a
while ka!=-1:
    if data_a[ka][0]!=n:
        qa=ka
        ka=data_a[ka][1]
    else:
        if ka==head_a:
            head_a=data_a[qa][1]
            data_a[ka]=None
        else:
            data_a[qa][1]=data_a[ka][1]
            data_a[ka]=None
        break

show_chain(data_a,head_a)
```