

# 插入一个结点

郝斌

# 链表中每个结点的存储结构【预备知识】

## ■ typedef struct node

{

int data; //存储数据本身

struct node \*pNext; //pNext指向一个和它本身存储指向下一个结点的指针

}NODE, \*PNODE;

■ 解释:

■ NODE等价于struct node

■ PNODE等价于struct node \*

## ■ PNODE p = (PNODE)malloc(sizeof(NODE));

■ 解释:

■ 将动态分配的新节点的地址赋给p

## ■ free p; //删除p指向结点所占的内存, 不是删除p本身所占内存

## ■ p->pNext; //p所指向结构体变量中的pNext成员本身

■ 把q所指向的结点插到p所指向的结点后面

■ 能否写

■ `p->pNext = q;`

■ `q->pNext = p->pNext;`

■ 错

## ■ 把q所指向的结点插到p所指向的结点后面

■ 方法一、先临时定义一个指向p后面结点的指针r

■ `r = p->pNext;` //r指向p后面的那个结点

■ `p->pNext = q;`

■ `q->pNext = r;`

## ■ 把q所指向的结点插到p所指向的结点后面

■ 方法二、

■ `q->pNext = p->pNext;`

■ `p->pNext = q;`

■ 注意：这两行代码的顺序不可以倒过来