1. 双链表的结构定义

```
#include <iostream>
using namespace std;

typedef struct DLNode{
    int data;
    struct DLNode *next;
    struct DLNode *prior;
} DLNode;
```

2. 双链表的基本操作

2.1 尾插法建立双链表

```
void createDListR(DLNode *&L, int a[], int n){
    DLNode *s, *r;
    int i;

    // 头结点
    L = (DLNode *)malloc(sizeof(DLNode *));
    L->next = NULL;
    r = L;

    for(i=0; i<n; i++){
        s = (DLNode *)malloc(sizeof(DLNode *));
        s->data = a[i];
        // printf("%d\n", a[i]);

        // 将 s 插入在 L 的尾部并且 r 指向 s,
        r->next = s;
        s->prior = r;
        // printf(" %d ",s->data);

        r = s;
    }
    r->next = NULL;
}
```

2.2 指定节点插入新节点

在双链表中 p 所指的结点之后插入一个结点 s， 先将要插入的结点两边链接好，这样有什么好处?对了， 就是可以保证不会发生链断之后找不到结点的情况。

指针变化过程如图 2.13 所示。

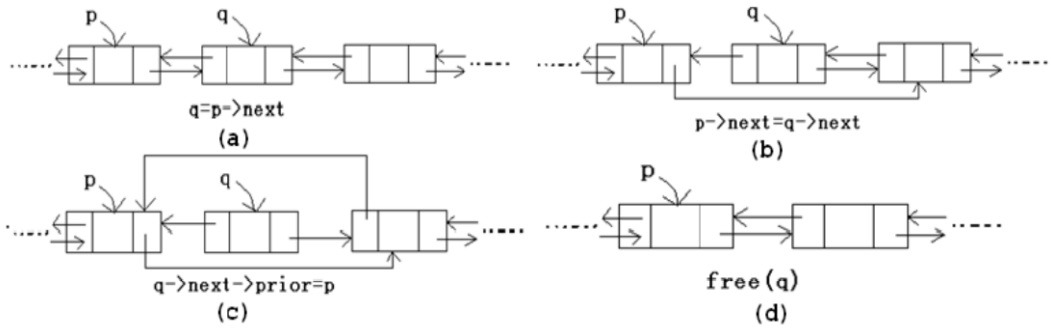

图 2.13 双链表结点删除过程图

```
bool insertAfterNode(DLNode *&p, DLNode *&s){
    if(p==NULL || s==NULL){
        return false;
    }
    // 先将要插入的结点两边链接好，这样有什么好处?对了， 就是可以保证不会发生链断之后找不到结点的情况。
    s->next = p->next;
    s->prior = p;
    p->next = s;

    // p 不为最后一个元素
    if(s->next != NULL)
        s->next->prior = s;  // s 不是 p

    return true;
}
```

2.3 指定节点删除后继节点

删除双链表中 p 结点的后继结点

指针变化过程如图 2.13 所示。

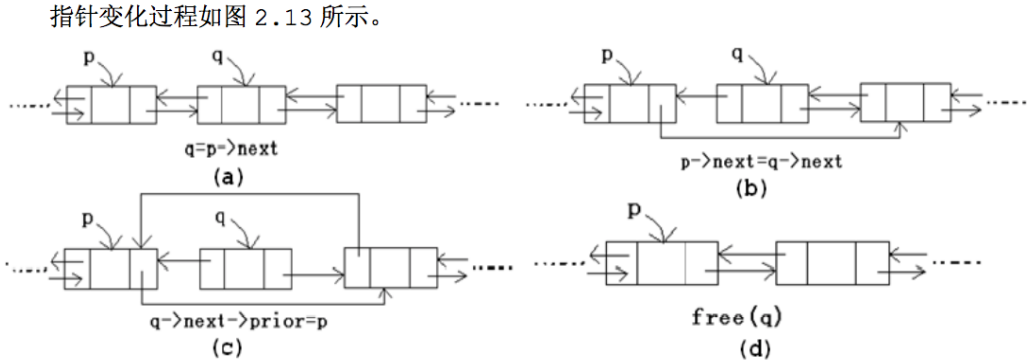

图 2.13 双链表结点删除过程图

```
int delAfterNode(DLNode *&p){
    if(p==NULL || p->next==NULL){
        return 0;
```

```
    }
    DLNode *q = p->next;
    p->next = q->next;
    if(q->next != NULL)
        // 都正常 printf(" %d\n ",q->next->next->data);
        q->next->prior = p;
    // osx 环境下编译执行会变成地址(ubuntu正常)，不知道为啥
    // -> printf(" %d\n ",q->next->next->data);
    free(q);
    return 1;
}
```

2.4 按元素值查找结点

从第一个结点开始，边扫描边比较，若找到这样的结点，则返回结点指针，否则返回 NULL。

```
DLNode* findNode(DLNode *L, int x){
    if(L==NULL){
        return NULL;
    }
    DLNode *tmp = L->next;
    while(tmp!=NULL){
        if(tmp->data == x){
            break;
        }
        tmp = tmp->next;
    }
    return tmp;  // 如果找到则 p 中内容是结点地址(循环因 break 结束)，
                 // 没找到 p 中内容是 NULL(循环因 p 等于 NULL 而结束)
}
```

2.5 打印双链表

```
void printDLinkedListH(DLNode *a){
    if(a==NULL){
        printf("双链表为空!!\n");
        return;
    }
    DLNode *tmp = a->next;
    printf("双链表: [");
    while(tmp!=NULL){
        printf(" %d ",tmp->data);
        tmp = tmp->next;
    }
    printf("]\n");
}
```