

第2章 线性表

2.1 线性表类型

1. 概念

线性表（表）、（直接）前驱、（直接）后继、长度、空表、位序、有序表、无序表



2. 线性结构

p. 30 第1段

3. 特点

在任何合理的位置都能插入和删除元素。

4. 规格说明

(1) 元素

(2) 结构

线性

(3) 操作

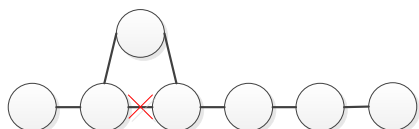
①clear() 清除线性表

②isEmpty() 判线性表空

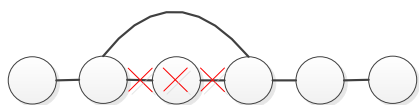
③length() 求线性表长度

④get() 取线性表元素

⑤insert() 在线性表中插入元素



⑥remove() 在线性表中删除元素



⑦indexOf() 定位（查找、搜索）线性表元素

⑧display(), 改为 traversal() 遍历（周游）线性表

还有初始化线性表、销毁线性表、求线性表当前元素的前驱、求线性表当前元素的后继等操作。

5. 例子

(1) 《数据结构（C语言版）》 p. 20 例 2-1

设 $LA = (8, 4, 5, 2)$, $LB = (5, 1, 9)$ 。

(2) 《数据结构（C语言版）》 pp. 20-21 例 2-2

2.2 顺序实现

1. 顺序表

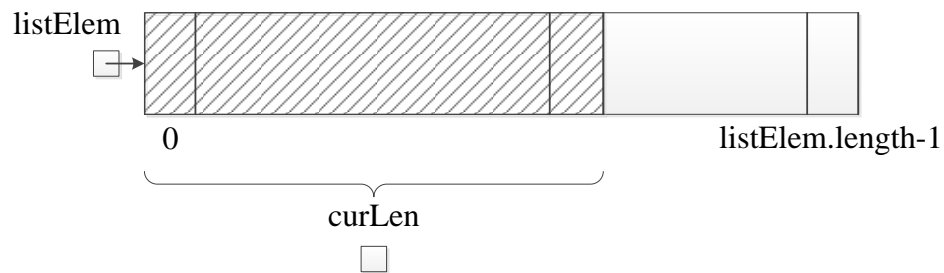
用顺序存储结构实现的线性表称为顺序表。

2. 特点

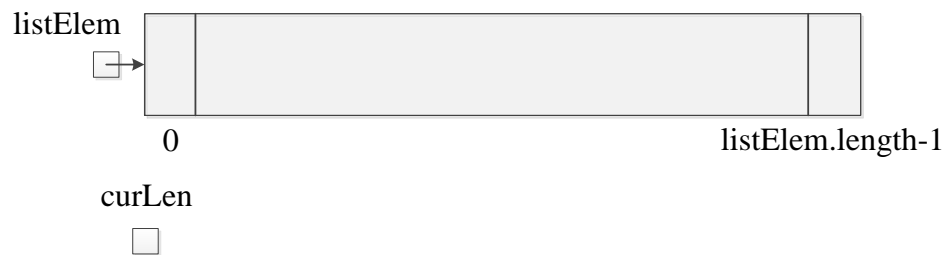
随机存取

3. 示意图

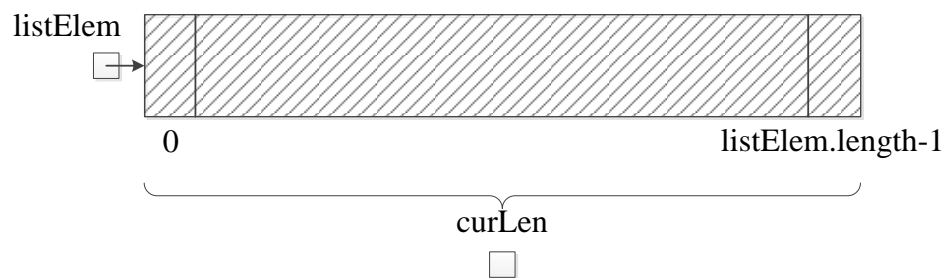
(1) 非空非满



(2) 空



(3) 满

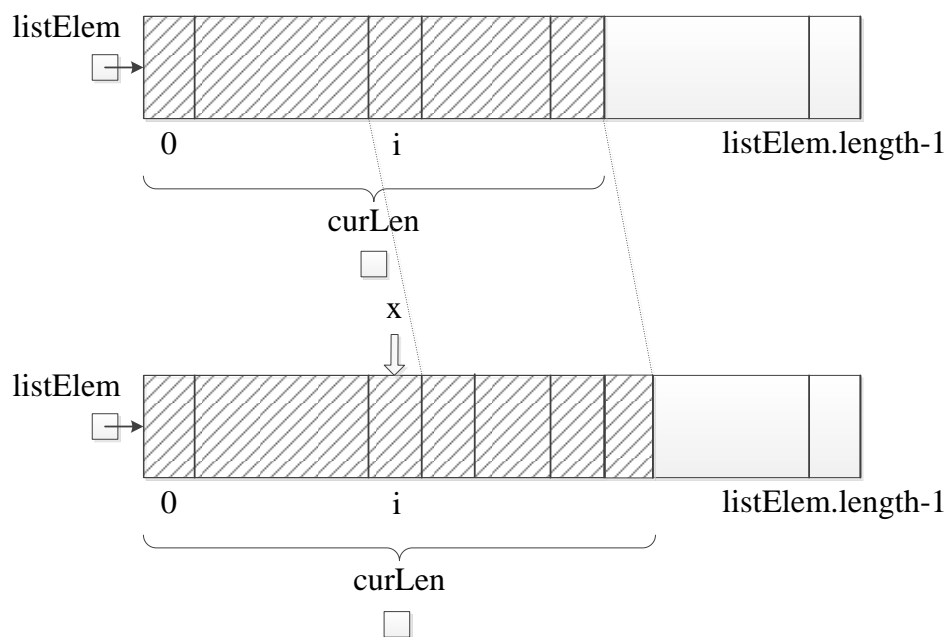


4. 类定义

pp. 32-33

5. 操作实现

(1) 在线性表中插入元素

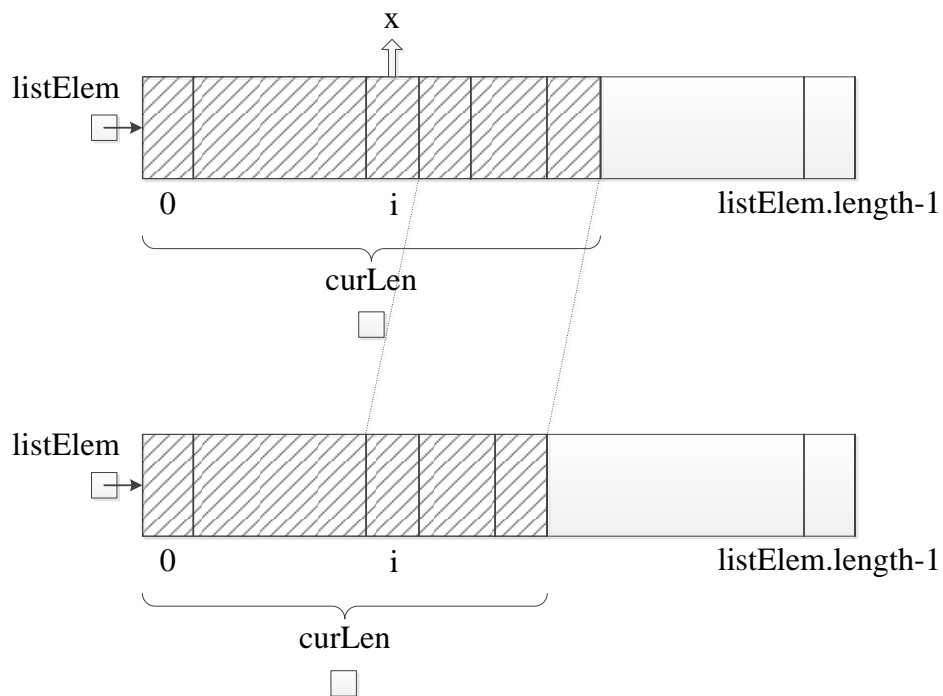


i 的合理取值范围: $0 \leq i \leq \text{长度}$

p. 34 算法 2.1

插入时平均移动 $\frac{n+(n-1)+(n-2)+\dots+1+0}{n+1} = \frac{n}{2}$ 个原有元素。

(2) 在线性表中删除元素



i 的合理取值范围: $0 \leq i \leq \text{长度}-1$

p. 35 算法 2.2

删除时平均移动 $\frac{(n-1)+(n-2)+\cdots+1+0}{n} = \frac{n-1}{2}$ 个原有元素。

(3) 定位（查找、搜索）线性表元素

p. 36 算法 2.3

可以在最后一个元素后面临时添加一个值为 x 的元素，称为增加哨兵的顺序查找法。

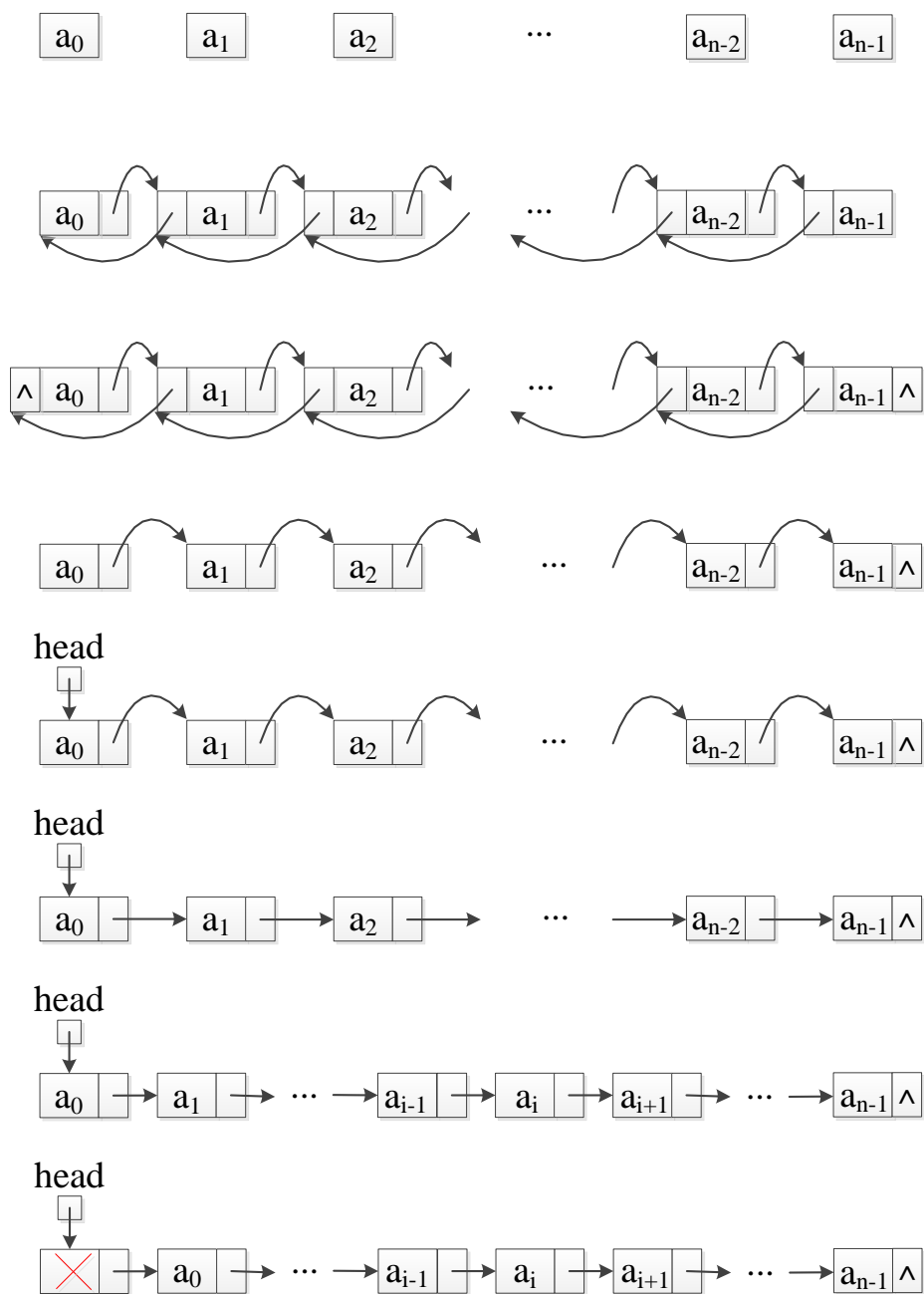
2.3 链式实现

1. 链表

用链式存储结构实现的线性表称为链表。

2. 概念

$(a_0, a_1, a_2, \cdots, a_{n-2}, a_{n-1})$



数据域+指针域=结点

结点：头结点、（表）结点

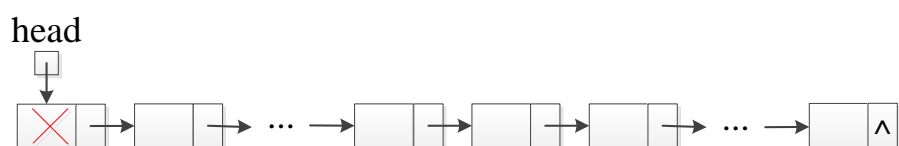
头指针

指针：内部指针、外部指针

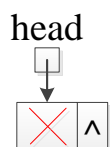
3. 单链表

（1）示意图

①非空

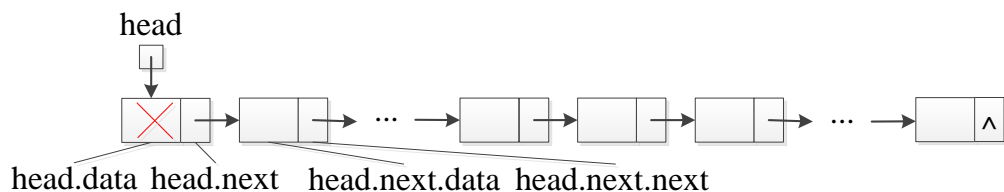


②空

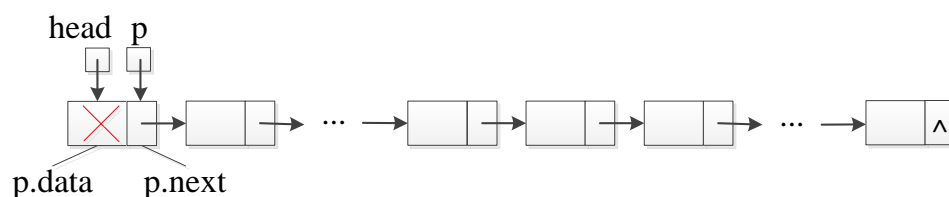


(2) 结点类定义

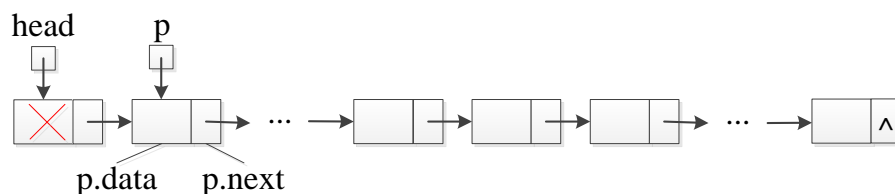
pp. 38-39



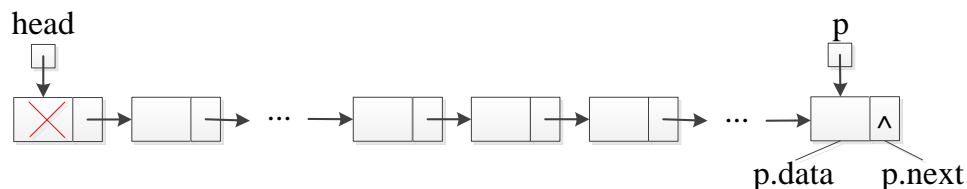
让 p 指向头结点: $p = head$;



让 p 指向下一个结点 (如果有的话): $p = p.next$;



当 p 指向最后一个结点时, $p.next == null$ 成立。

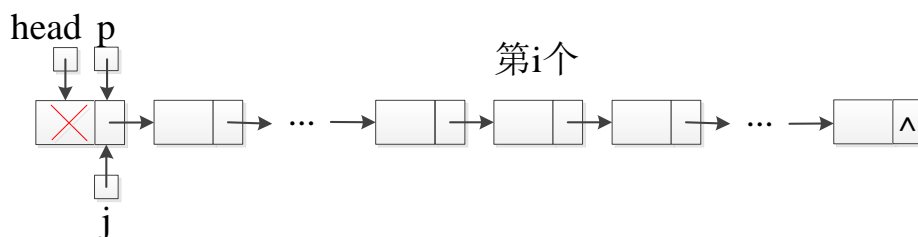


(3) 单链表类定义

pp. 39-40

(4) 操作实现

①取线性表元素



p. 41 算法 2.4

注意, 该算法开始执行时, 让真指针 p 和假指针 j 同时指向第 0 个结点, 而不是头结点, 与上图有差异。

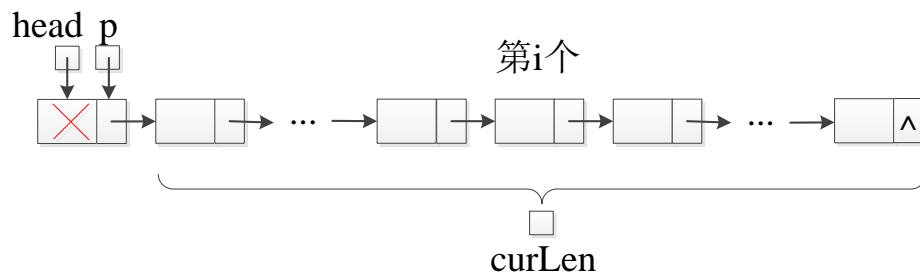
a. 若 $i < 0$, 则取线性表元素失败。循环结束时, $j > i$ 。

b. 若 $0 \leq i \leq \text{长度}-1$ ，则取线性表元素成功。循环结束时， $j=i$ 。

c. 若 $i = \text{长度}$ ，则取线性表元素失败。循环结束时， $j=i$ 。

d. 若 $i > \text{长度}$ ，则取线性表元素失败。循环结束时， $j < i$ 。

应该引入存储线性表长度的整型变量 curLen 。



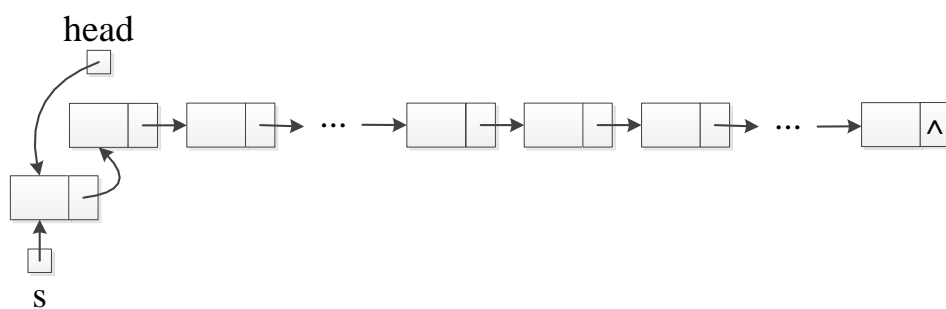
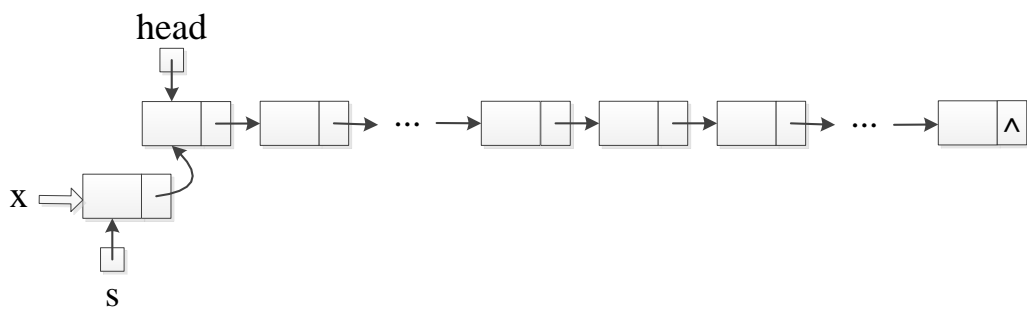
②定位（查找、搜索）线性表元素

p. 41 算法 2.5

③在线性表中插入元素

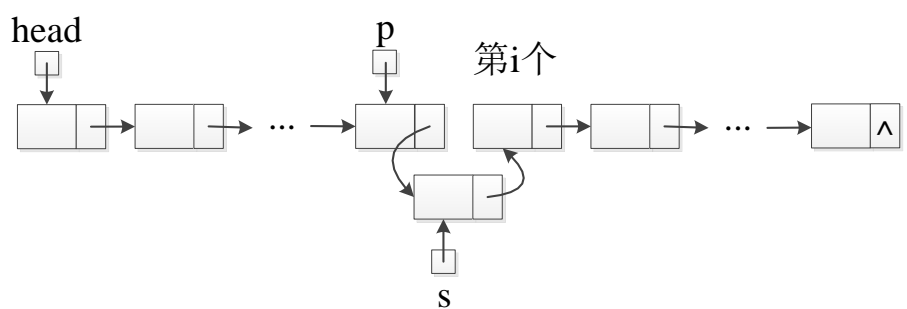
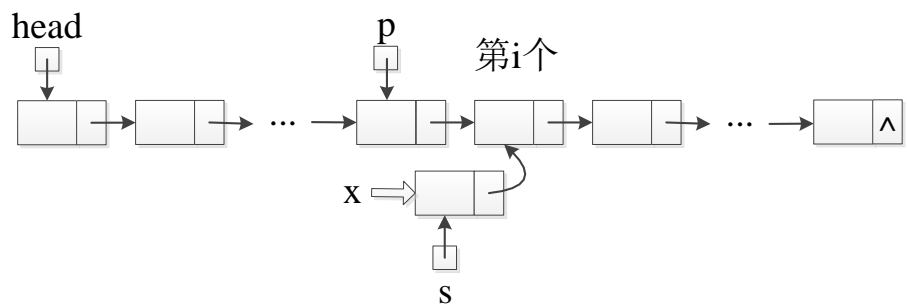
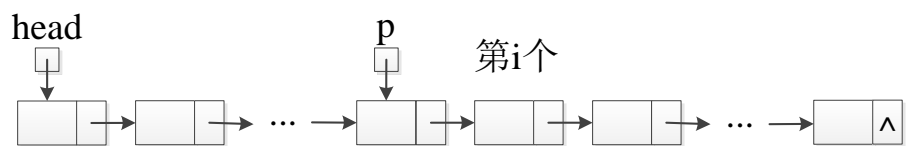
对于不带头结点的单链表：

在第 0 个元素前插入新元素 x



在非第 0 个元素前插入新元素 x

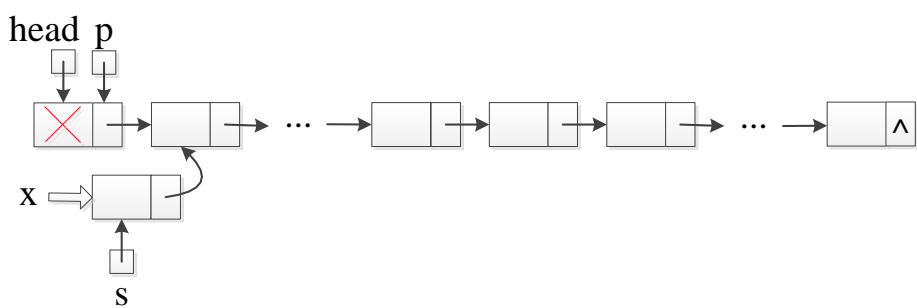
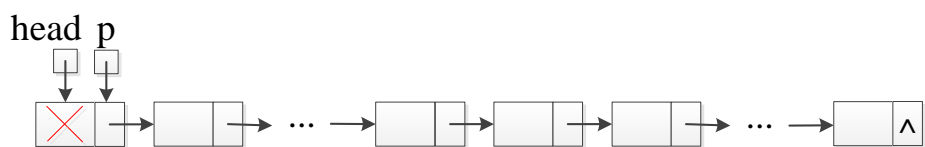


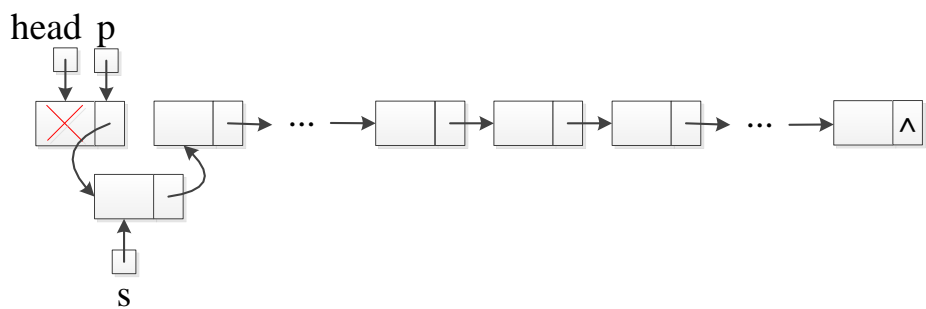


pp. 43-44 算法 2.7

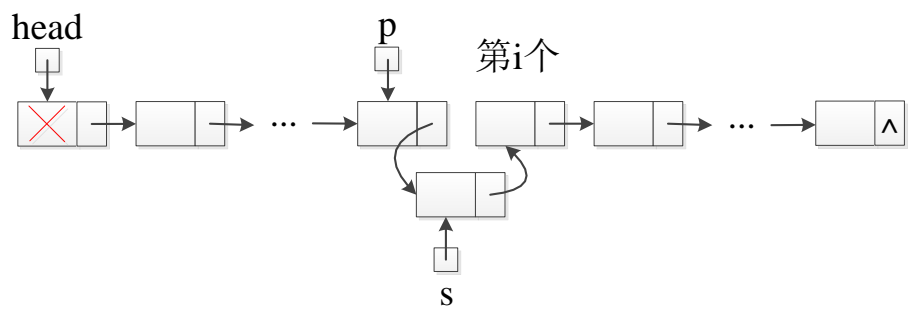
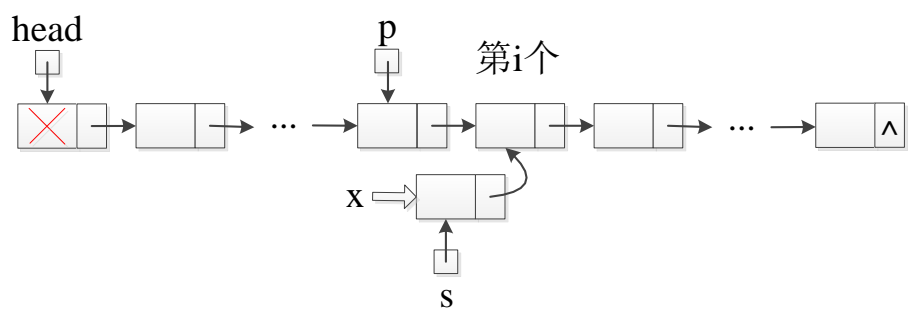
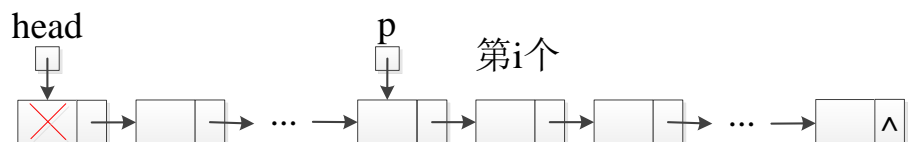
对于带头结点的单链表:

在第 0 个元素前插入新元素 x





在非第 0 个元素前插入新元素 x

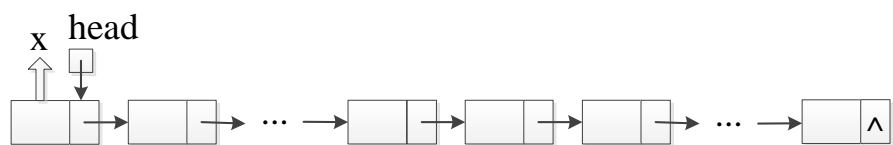


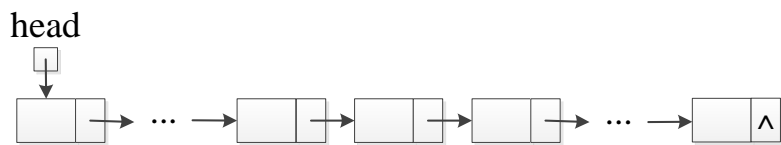
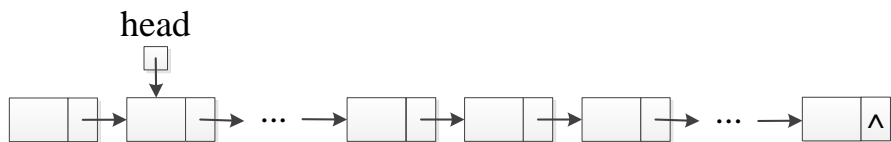
p. 42 算法 2.6

④在线性表中删除元素

对于不带头结点的单链表:

删除第 0 个元素



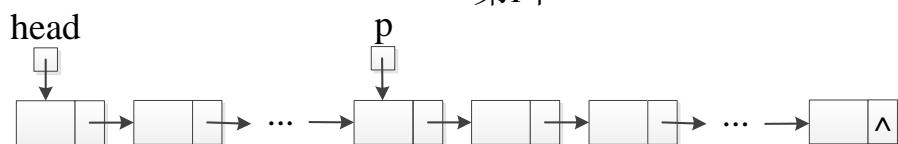


删除非第 0 个元素

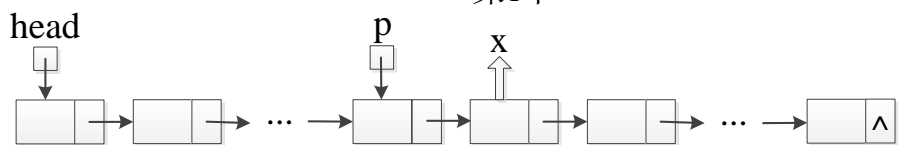
第i个



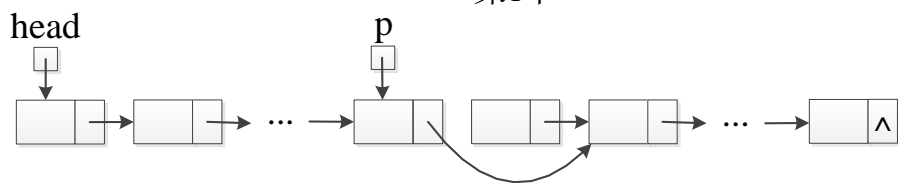
第i个



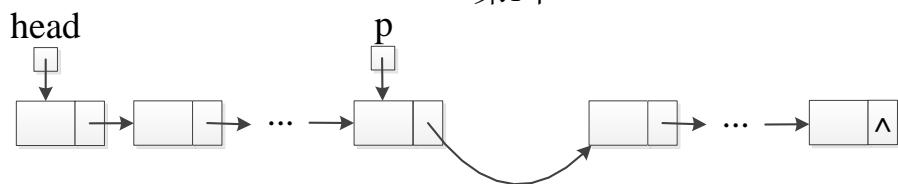
第i个



第i个

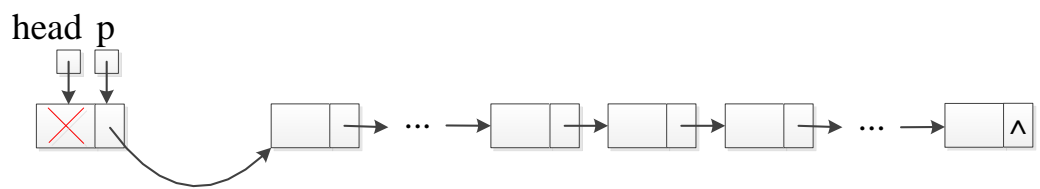
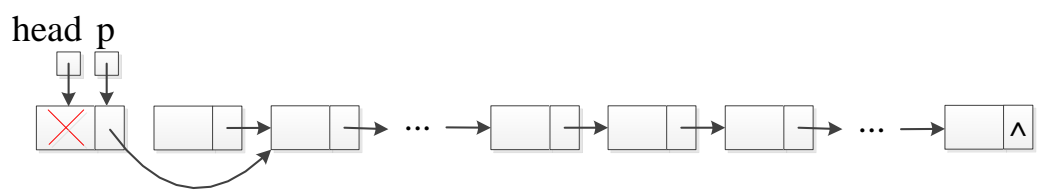
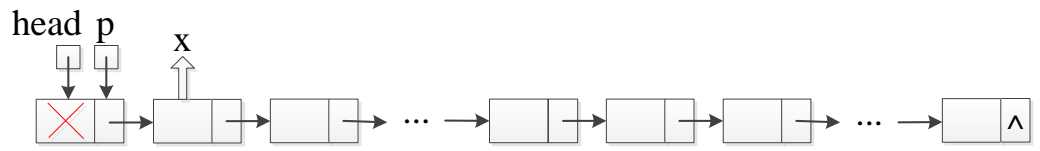
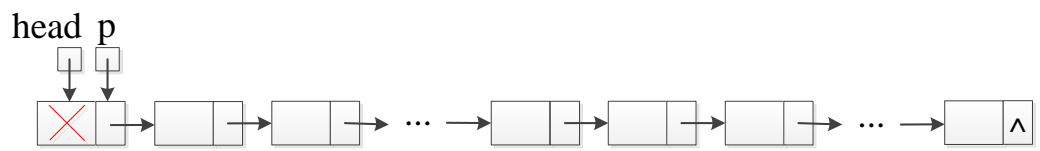


第i个



对于带头结点的单链表：

删除第 0 个元素

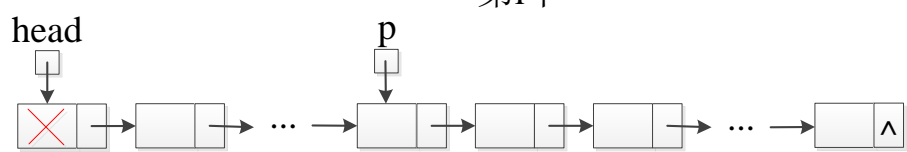


删除非第 0 个元素

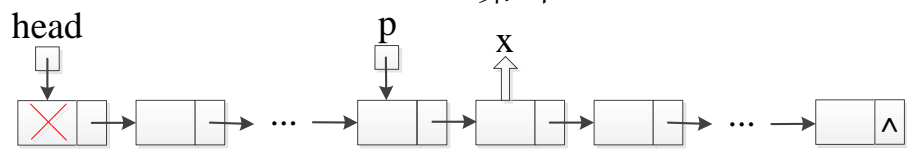
第i个

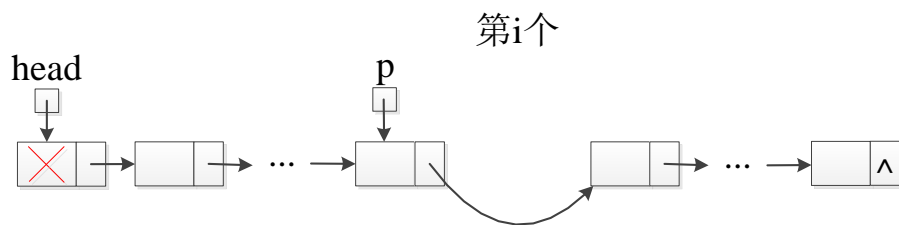
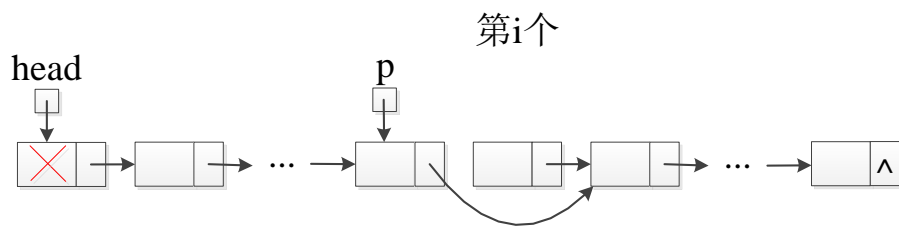


第i个



第i个

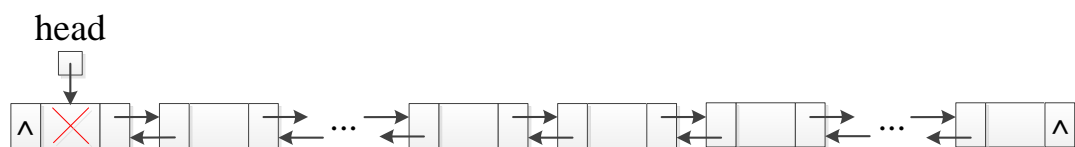




pp. 44-45 算法 2.8

4. 双链表

(1) 示意图

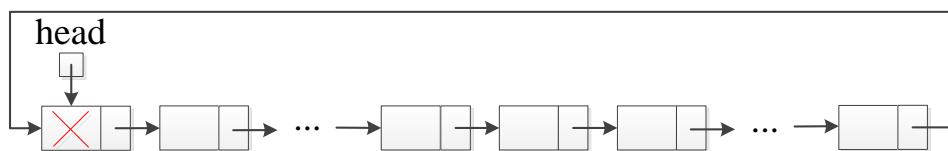


(2) 结点类定义

p. 52

5. 单循环链表

(1) 示意图

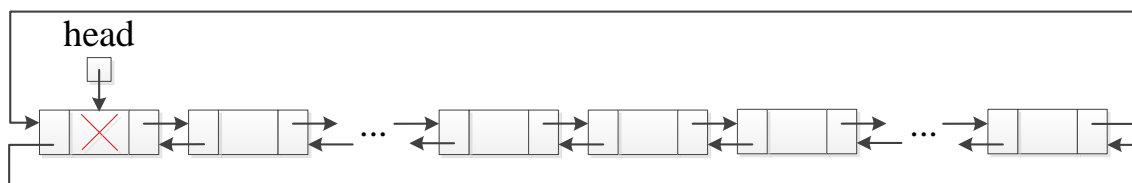


(2) 结点类定义

形式上与单链表的相同。

6. 双循环链表

(1) 示意图



(2) 结点类定义

形式上与双链表的相同。