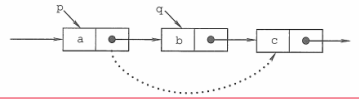
2.3线性表的链式表示(中)

单链表上基本操作的实现

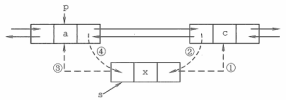
- 对某一结点进行前插操作
  - 即寻找到要插入结点的前一个结点，然后按照正常插入方法执行即可
  - 时间复杂度为O(n)
- 删除结点操作
  - 删除结点操作是将单链表的第1个结点删除。先检查删除位置的合法性,后查找表中第1-1 个结点,即被删结点的前驱结点,再将其删除
  - 示意图



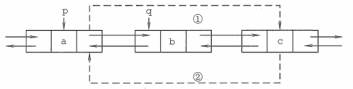
  - 该算法的主要时间也耗费在查找操作上,时间复杂度为O(n)
- 删除结点*p
  - 方法一
    - 要删除某个给定结点*p,通常的做法是先从链表的头结点开始顺序找到其前驱结点,然后再执行删除操作
    - 算法的时间复杂度为 O(n)
  - 方法二
    - 删除结点*p的操作可用删除*p的后继结点操作来实现,实质就是将其后继结点的值赋予其自身,然后删除后继结点
    - 时间复杂度为O(1)。
- 求表长操作
  - 从第一个结点开始顺序依次访问表中的每个结点，为此需要设置一个计数器变量，每访问一个结点，计数器加1,直到访问到空结点为止
  - 算法的时间复杂度为 O(n)

双链表

- 单链表的缺点
  - 单链表只能从头结点依次顺序地向后遍历
  - 不能够快速的对其前驱结点进行操作
- 概念　双链表结点中有两个指针prior和next,分 别指向其前驱结点和后继结点
- 双链表的插入操作
  - 示意图



  - 原则就是在进行双链表插入的时候要保证不会造成断链
- 双链表的删除操作
  - 示意图



  - 原则仍然是在进行操作的时候一定要考虑是否会造成断链

操作的时候一定要考虑，本操作顺序是否会造成后继结点的丢失

循环链表

- 循环单链表
  - 循环单链表和单链表的区别在于,表中最后一个结点的指针不是NULL,而改为指向头结点，形成了一个环
  - 判空条件：头结点的指针是否等于头指针
  - 仅设尾指针
    - 如果是要在链首之前插入结点，此时效率明显更高
    - 时间复杂度O（1）
- 循环双链表
  - 在循环双链表中,头结点的 prior指针还要指向表尾结点
  - 在循环双链表L中,某结点*p为尾结点时,p->next==L
  - 当循环双链表为空表时,其头结点的 prior 域和 next域都等于L

静态链表

- 基本结构
  - 静态链表借助数组来描述线性表的链式存储结构　静态链表也要预先分配一块连续的内存空间
  - 结点也有数据域data和指针域next　这里的指针是结点的相对地址(数组下标)，又称游标
- 特点
  - 静态链表以 next==-1作为其结束的标志
  - 静态链表的插入、删除操作与动态链表的相同, 只需要修改指针,而不需要移动元素
- 优点：增、删 操作不需要大量移动元素
- 缺点：不能随机存取，只能从头结点开始依次往后查找；容量固定不可变
- 适用场景：①不支持指针的低级语言；②数据元素数量固定不变的场景（如操作系统的文件分配表FAT）