


2.3.4 循环链表

循环链表是另一种形式的链式存储结构形式。

- 循环单链表：将表中尾节点的指针域改为指向表头节点，整个链表形成一个环。由此从表中任一节点出发均可找到链表中其他节点。
- 循环双链表：形成两个环。

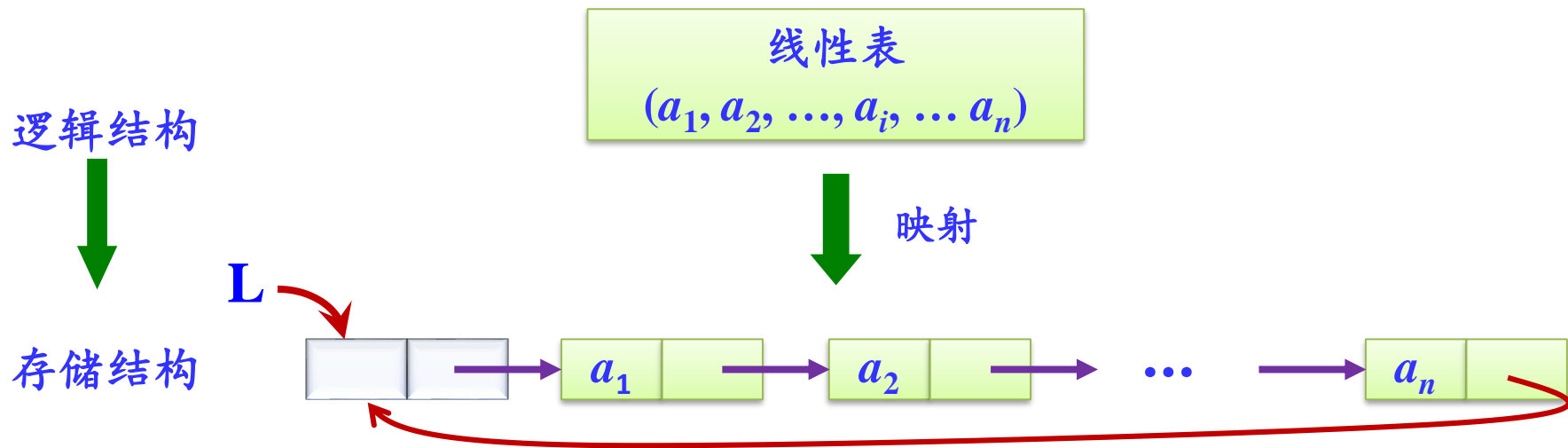


节点类型与非
循环单链表的
相同



节点类型与非
循环双链表的
相同

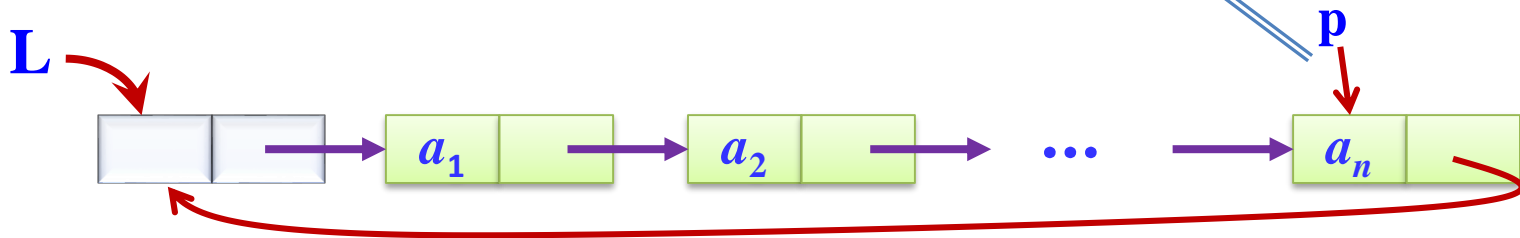
1、循环单链表



带头节点循环单链表示意图

与非循环单链表相比，循环单链表：

- 链表中没有空指针域
- p所指节点为尾节点的条件： $p \rightarrow next == L$



【例2-8】 某线性表最常用的操作是在尾元素之后插入一个元素和删除第一个元素，故采用_____存储方式最节省运算时间。

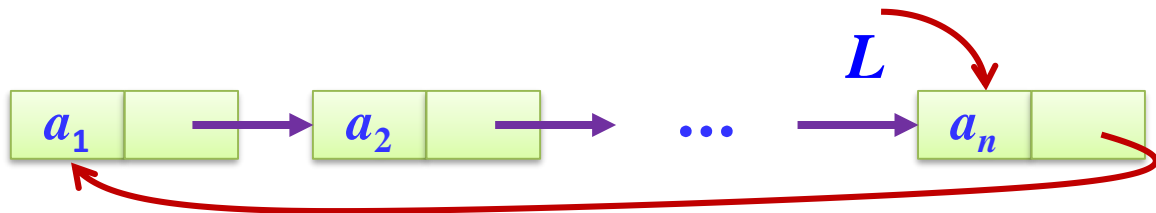
A.单链表

B.仅有头结点指针的循环单链表

C.双链表

D.仅有尾结点指针的循环单链表

D. 仅有尾结点指针的循环单链表



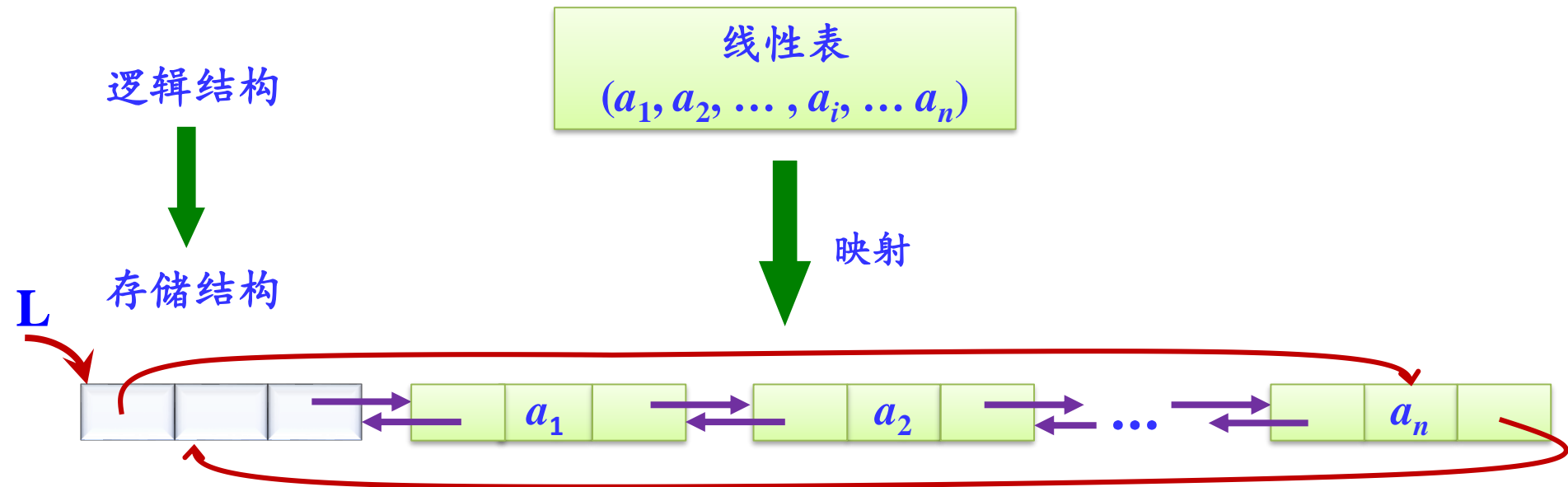
- 在尾元素之后插入一个元素
- 删除第一个元素



时间复杂度
均为 $O(1)$

选择D

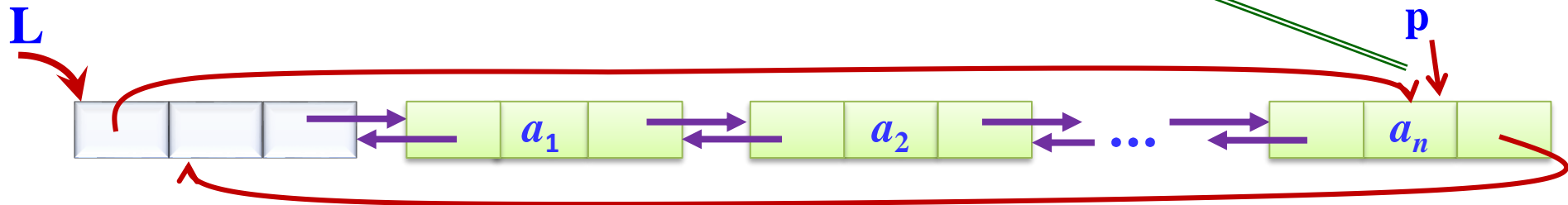
2、循环双链表



带头节点循环双链表示意图

与非循环双链表相比，循环双链表：

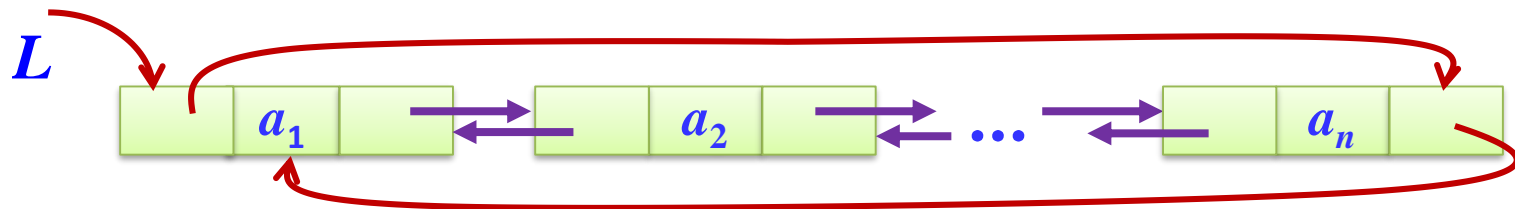
- 链表中没有空指针域
- p 所指节点为尾节点的条件： $p \rightarrow \text{next} == L$
- 一步操作即 $L \rightarrow \text{prior}$ 可以找到尾节点



【例2-9】如果对含有 n ($n>1$) 个元素的线性表的运算只有4种, 即删除第一个元素、删除尾元素、在第一个元素前面插入新元素、在尾元素的后面插入新元素, 则最好使用_____。

- A.只有尾结点指针没有头结点的循环单链表
- B.只有尾结点指针没有头结点的非循环双链表
- C.只有首结点指针没有尾结点指针的循环双链表
- D.既有头指针也有尾指针的循环单链表

C.只有首结点指针没有尾结点指针的循环双链表



- 删除第一个元素
- 删除尾元素
- 在第一个元素前面插入新元素
- 在尾元素的后面插入新元素



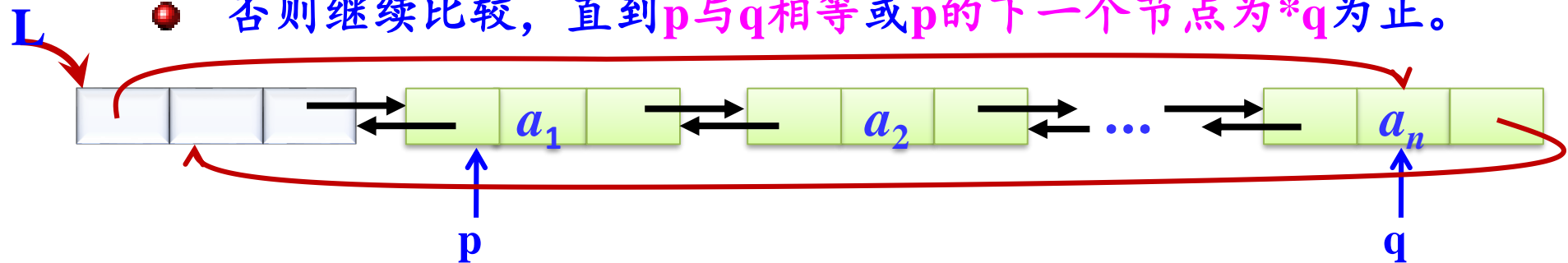
时间复杂度
均为 $O(1)$

选择C

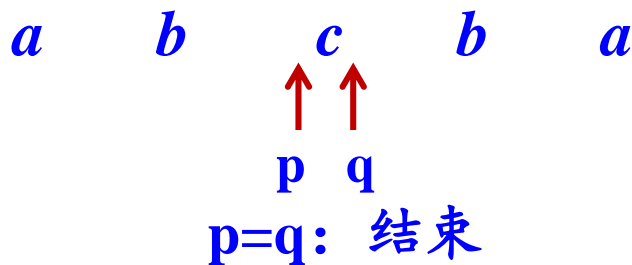
【例2-10】设计判断带头节点的循环双链表L（含两个以上的节点）是否对称相等的算法。

算法思路

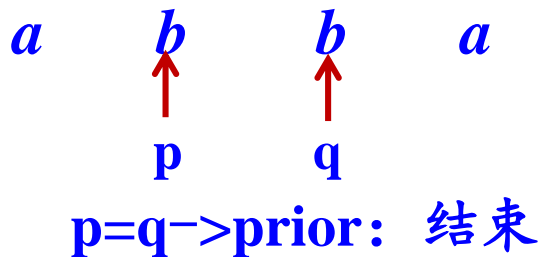
- p从左向右扫描L，q从右向左扫描L
- 若对应数据节点的data域不相等，则退出循环
- 否则继续比较，直到p与q相等或p的下一个节点为*q为止。



① 数据节点为奇数的情况:



② 数据节点为偶数的情况:



```
int Equal(DLinkedList *L)
{   int same=1;
    DLinkedList *p=L->next;        //p指向第一个数据节点
    DLinkedList *q=L->prior;        //q指向最后数据节点
    while (same==1)
    {   if (p->data!=q->data)
        same=0;
        else
        {   if (p==q || p==q->prior) break;
            q=q->prior;                //q前移
            p=p->next;                //p后移
        }
    }
    return same;
}
```

思考题

循环链表的作用？在什么情况下使用循环链表？

——本讲完——