# 第2章　思考题

## 1-1、判断题（线性表基本概念）

（1）线性表的链式存储结构优于顺序存储。（×）

（2）链表的每个结点都恰好包含一个指针域。（×）

（3）在线性表的链式存储结构中，逻辑上相邻的两个元素在物理位置上并不一定紧邻。（√）

（4）顺序存储方式的优点是存储密度大，插入、删除效率高。（×）

（5）线性链表的删除算法简单，因为当删除链中某个结点后，计算机会自动地将后续的各个单元向前移动。（×）

（6）顺序表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。（×）

（7）线性表链式存储的特点是可以用一组任意的存储单元存储表中的数据元素。（√）

（8）线性表采用顺序存储，必须占用一片连续的存储单元。（√）

（9）顺序表结构适宜于进行顺序存取，而链表适宜于进行随机存取。（×）

（10）插入和删除操作是数据结构中最基本的两种操作，所以这两种操作在数组中也经常使用。（×）

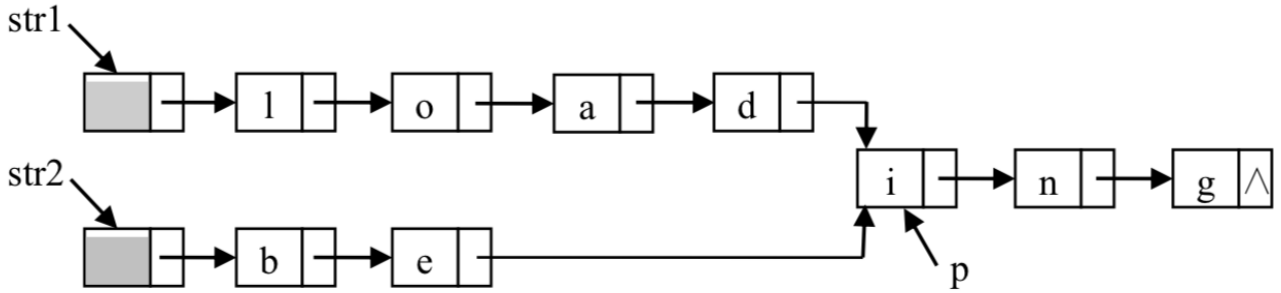
## 【算法设计1】

已知两个整型数组A和B，A中的元素从小到大排列，B中的元素从大到小排列，A和B不含相同的元素，并已知数组A的元素个数为na，数组B的元素个数为nb，设计一个算法（源代码）将A和B合并成为一个数组，合并后的结果放在A中，且A仍保持从小到大排列，不另设新的数组。

要求算法的时间复杂度控制在O(n)/O(na+nb)，辅组空间为O(1)。

## 【算法设计2】

假定采用带头结点的单链表保存单词，当两个单词有相同的后缀时，则可共享相同的后缀存储空间。例如，“loading”和“being”的存储映像如下图所示。




设 *str1* 和 *str2* 分别指向两个单词所在单链表的头结点，链表结点结构为（data,next），请设计一个时间上尽可能高效的算法，找出由 *str1* 和 *str2* 所指的两个链表共同后缀的起始位置(如图中字符 *i* 所在结点的位置 *p*)。要求:

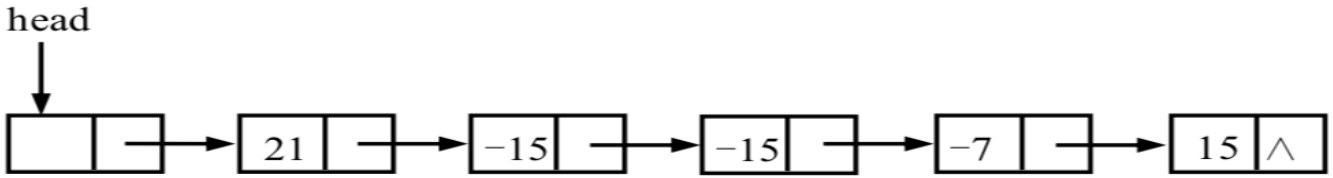
(1)给出算法的基本设计思想。

(2)根据设计思想，采用 C 或 C++或 JAVA 语言描述算法，关键之处给出注释。

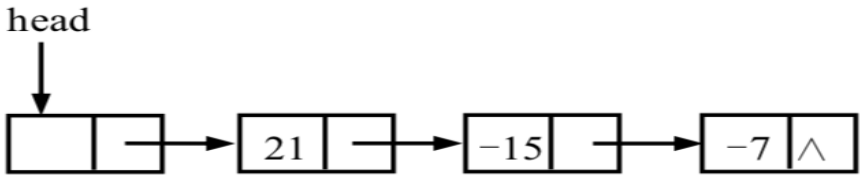
(3)说明你所设计算法的时间复杂度。

## 【算法设计3】

用单链表保存 $m$ 个整数，结点的结构为: data link ，且|data|≤$n$($n$ 为正整数)。现要求设计一个时间复杂度尽可能高效的算法，对于链表中 data 的绝对值相等的结点，仅保留第一次出现的结点而删除其余绝对值相等的结点。例如，若给定的单链表 head 如下:




则删除结点后的 head 为:




要求: (1) 给出算法的基本设计思想。

(2) 使用 C 或 C++语言，给出单链表结点的数据类型定义。

(3) 根据设计思想，采用 C 或 C++语言描述算法，关键之处给出注释。

(4) 说明你所设计算法的时间复杂度和空间复杂度。

## 【算法设计4】

查找倒数第k个元素(尾结点记为倒数第0个)

## 【算法设计5】查找中间结点

设两个初始化指向头结点的指针p, q。p每次前进两个结点, q每次前进一个结点, 这样当p到达链表尾巴的时候, q到达了中间.复杂度O(n)。

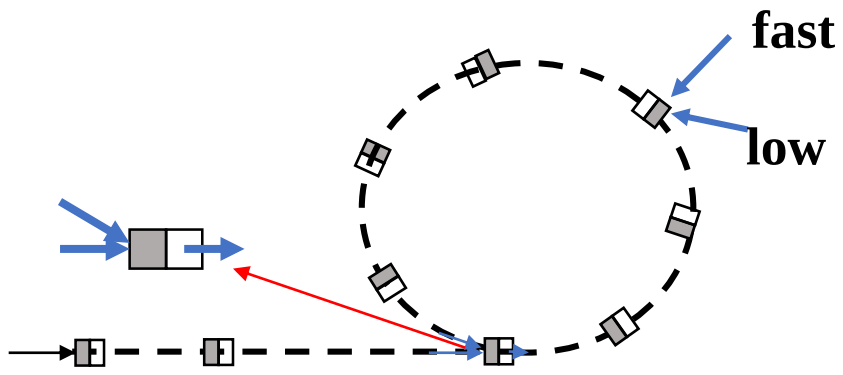
## 【算法设计6】逆序打印链表

已知链表的头结点， 逆序打印这个链表. 使用递归(即让系统使用栈)， 时间复杂度O(n)

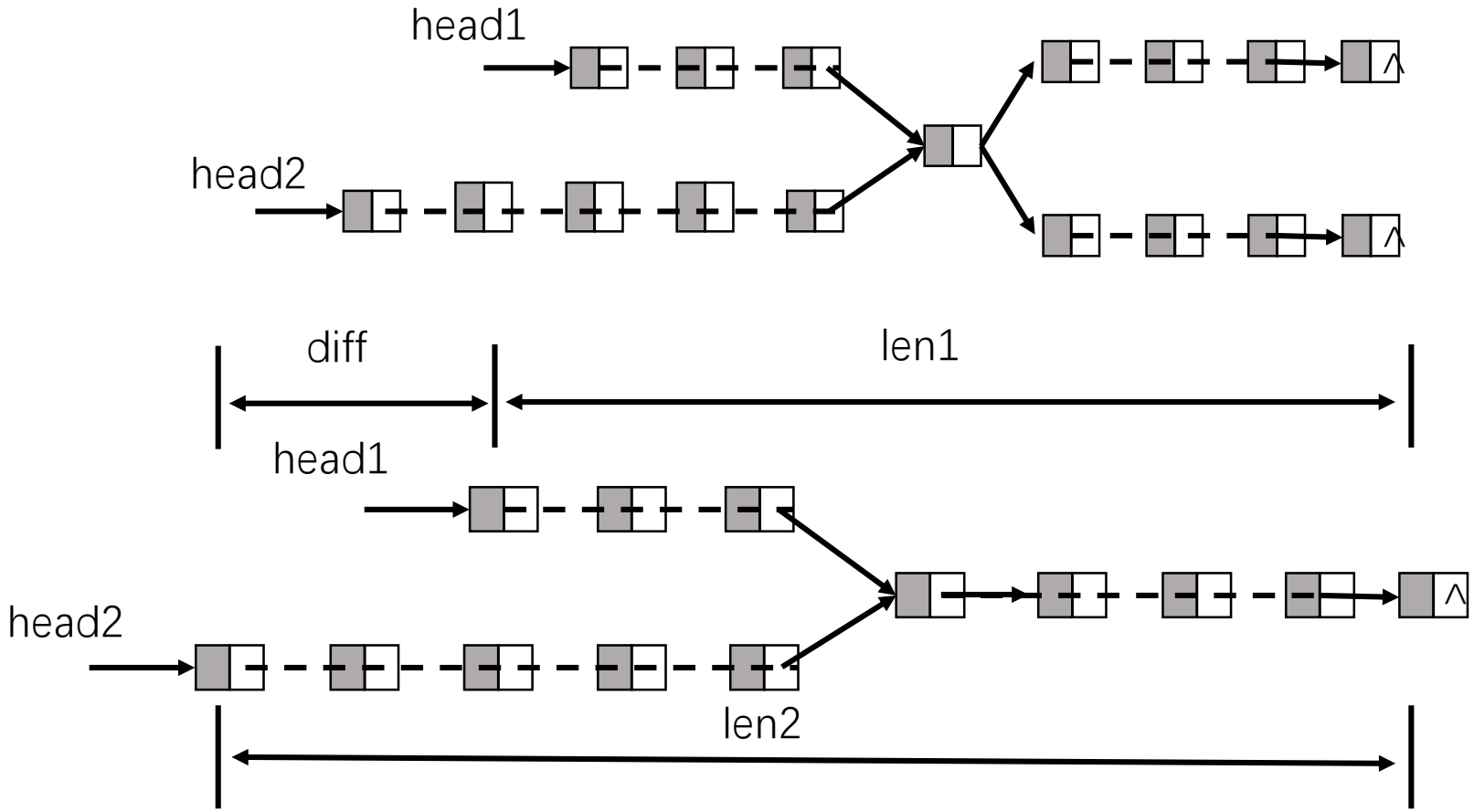
# 【算法设计7】线性链表环路问题

问题1：如何判断单链表中是否存在环？

问题2：若存在环，如何找到环的入口点？

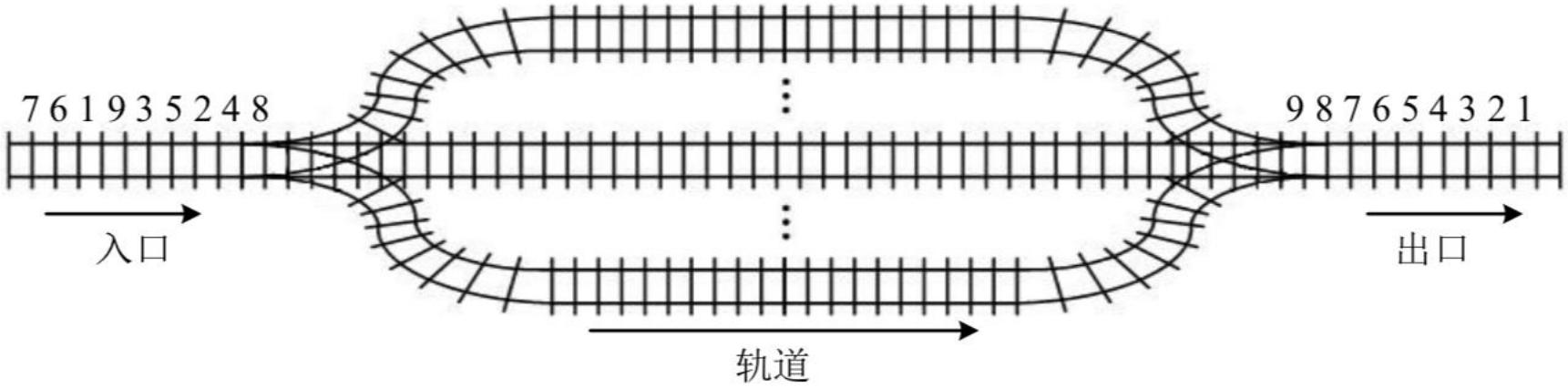


# 【算法设计8】线性链表交叉问题



**讨论题1：**

设有如下图所示的火车车轨，入口到出口之间有 $n$ 条轨道，列车的行进方向均为从左至右，列车可驶入任意一条轨道。现有编号为 1~9 的 9 列列车，驶入的次序依次是 8, 4, 2, 5, 3, 9, 1, 6, 7。若期望驶出的次序依次为 1 至 9，则 $n$ 至少是？




要点：1）考虑“至少”

2）每个队列中后边的元素要大于前边的元素

## 讨论题2：

在如下数组A中链接存储了一个线性表，表头指针为A [0].next，试写出该线性表。

| A | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| data | | 60 | 50 | 78 | 90 | 34 | | 40 |
| next | 3 | 5 | 7 | 2 | 0 | 4 | | 1 |

```
Struct {
        int data;
        int next;}   A[Max}
```