

003-数据结构与算法 [线性表算法练习篇]



一. 课程安排:

- 课程日期: 2020 年 4 月 8 日 第 4 次课程 (共9次课程)
- 授课老师: CC 老师
- 研发老师: CC 老师
- 课程时长: 2小时
- 课程主题: 线性表专题下链表相关算法实现
- 课程时间安排:
 - 上课: 20:00 – 21:00
 - 休息: 21:00 – 21:10
 - 上课: 21:10 – 22:00

• 课程作业:

- 博客,关于链表的面试题解析!

• 课程内容安排:

- 链表算法题分析与实现!

二. 课程内容笔记

- 题目1 将2个递增的有序链表合并为一个有序链表; 要求结果链表仍然使用两个链表的存储空间,不另外占用其他的存储空间. 表中不允许有重复的数据

La {1,2,3} , Lb {3, 6,9}

Lc {1, 2, 3, 6, 9}

关键字:

递增有序链表;

不允许有重复数据

保持递增关系(后插法)

不占用额外空间: 额外新建结点维持逻辑

思路:

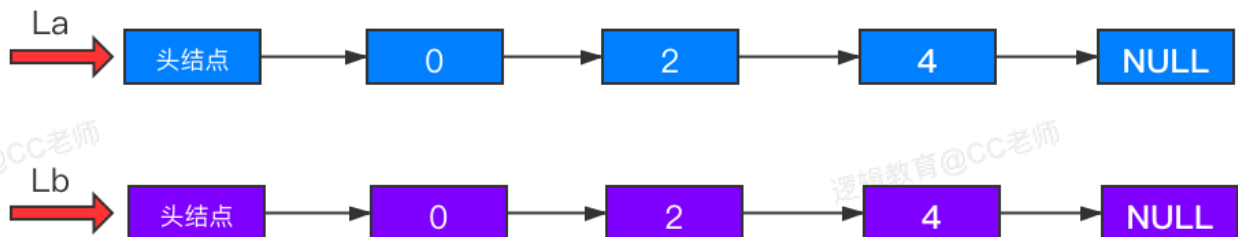
a. 待合并链表 La,Lb, 头指针Lc; (借用La的头结点)

b. pa,pb ,

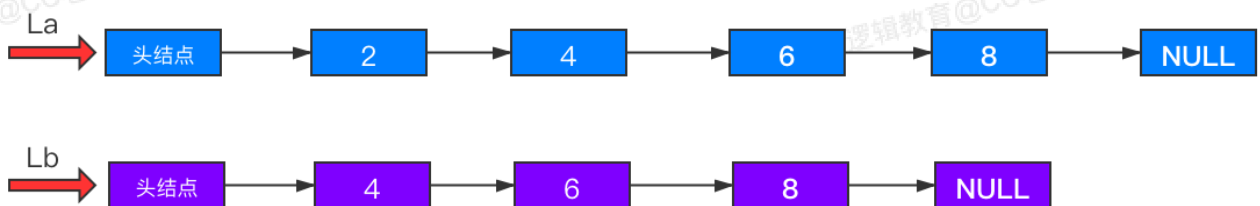
c. 循环(La/Lb 都没有达到尾结点位置)

d. 判断链表取较小者,存储Lc

e. 相等La元素插入Lc,Lb 释放掉



- 题目2: 已知两个链表A和B分别表示两个集合.其元素递增排列. 设计一个算法,用于求出A与B的交集,并存储在A链表中;



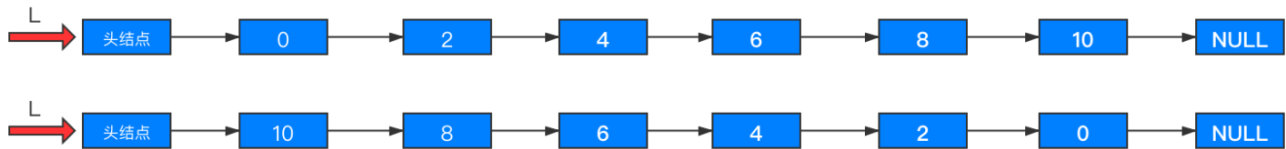
- La,Lb,Lc(La 的头结点)
- pa,pb,首元结点
- 循环条件pa,pb 不为空
- 相等, La 结点,链接Lc; Lb 结点释放
- 较小 干掉!
- 不存在交集多余释放

- 题目3: 设计一个算法,将链表中所有节点的链接方向原地旋转,即要求仅仅利用原表的存储空间. 换句话说,要求算法空间复杂度为 $O(1)$;

例如: $L=\{0,2,4,6,8,10\}$, 逆转后: $L = \{10,8,6,4,2,0\}$;

思路:

- *L 作为新的旋转后的链表首结点 p 首元结点;
- 从前往后遍历
- 临时p 指向待插入结点后继结点



- 题目4: 设计一个算法,删除递增有序链表中值大于等于minK且小于等于maxK(minK,maxK是给定的两个参数,其值可以和表中的元素相同,也可以不同)的所有元素;



思路:

- 遍历链表 第一个大于minK结点,pre 保存前驱;
- 遍历链表 第一个大于maxK结点, p指向;
- $pre \rightarrow next = p$;
- 介于pre和p之间结点删除掉

- 题目5: 设将 $n(n>1)$ 个整数存放到一维数组R中, 试设计一个在时间和空间两方面都尽可能高效的算法;将R中保存的序列循环左移p个位置($0<p<n$)个位置, 即将R中的数据由 $(x_0, x_1, \dots, x_{n-1})$ 变换为 $(x_p, x_{p+1}, \dots, x_{n-1}, x_0, x_1, \dots, x_{p-1})$.

思路:

- n个数据原地逆置 9 8 7 6 5 4 3 2 1 0
- 拆解[9 8 7 6 5 4 3][2 1 0]

c. n-p数据/p个数据 再一次逆置 [3 4 5 6 7 8 9][0 1 2]

- 题目6: 已知一个整数序列 $A = (a_0, a_1, a_2, \dots, a_{n-1})$, 其中 $(0 \leq a_i \leq n), (0 \leq i \leq n)$. 若存在 $a_{p_1} = a_{p_2} = \dots = a_{p_m} = x$, 且 $m > n/2$ ($0 \leq p_k < n, 1 \leq k \leq m$), 则称 x 为 A 的主元素. 例如, $A = (0, 5, 5, 3, 5, 7, 5, 5)$, 则5是主元素; 若 $B = (0, 5, 5, 3, 5, 1, 5, 7)$, 则 A 中没有主元素, 假设 A 中的 n 个元素保存在一个一维数组中, 请设计一个尽可能高效的算法, 找出数组元素中的主元素, 若存在主元素则输出该元素, 否则输出-1.
- 题目7: 用单链表保存 m 个整数, 结点的结构为 $(data, link)$, 且 $|data| \leq n$ (n 为正整数). 现在要去设计一个时间复杂度尽可能高效的算法. 对于链表中的 $data$ 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, 链表 $A = \{21, -15, 15, -7, 15\}$, 删除后的链表 $A = \{21, -15, -7\}$;