



逻辑教育  
Logic education

# Hello 数据结构与算法

数据结构与算法 — 线性表[专题]

数据结构与算法主题[2]

@HelloCoder\_CC

全力以赴·非同凡“想”

课程研发:CC老师

课程授课:CC老师

转载需注明出处,不得用于商业用途.已申请版权保护



## 线性表—顺序表[ 01 ]

对于非空的线性表和线性结构,其特点如下:

- 存在唯一的一个被称作“第一个”的数据元素;
- 存在唯一的一个被称作“最后一个”的数据元素
- 除了第一个之外,结构中的每个数据元素均有一个前驱
- 除了最后一个之外,结构中的每个数据元素都有一个后继.



## **ADT List {**

**Data:** 线性表的数据对象集合为 $\{a_1, a_2, \dots, a_n\}$ , 每个元素的类型均为 $DataType$ . 其中, 除了第一个元素 $a_1$ 外, 每一个元素有且只有一个直接前驱元素, 除了最后一个元素 $a_n$ 外, 每个元素有且只有一个直接后继元素. 数据元素之间的关系是一对一的关系.

## **Operation**

### **InitList(&L)**

操作结果: 初始化操作, 建立一个空的线性表 $L$ .

### **DestroyList(&L)**

初始条件: 线性表 $L$ 已存在

操作结果: 销毁线性表 $L$ .

### **ClearList(&L)**

初始条件: 线性表 $L$ 已存在

操作结果: 将 $L$ 重置为空表.

### **ListEmpty(L)**

初始条件: 线性表 $L$ 已存在

操作结果: 若 $L$ 为空表, 则返回 $true$ , 否则返回 $false$ .

### **ListLength(L)**

初始条件: 线性表 $L$ 已存在

操作结果: 返回 $L$ 中数据元素的个数

.....

课程研发: CC老师

课程授课: CC老师



.....

***GetElem(L,i,&e)***

初始条件: 线性表 $L$ 已存在,且 $1 \leq i < \text{ListLength}(L)$

操作结果: 用 $e$ 返回 $L$ 中第 $i$ 个数据元素的值;

***LocateElem(L,e)***

初始条件: 线性表 $L$ 已存在

操作结果: 返回 $L$ 中第1个值与 $e$ 相同的元素在 $L$ 中的位置. 若数据不存在则返回0;

***PriorElem(L, cur\_e,&pre\_e);***

初始条件: 线性表 $L$ 已存在

操作结果: 若 $cur\_e$ 是 $L$ 的数据元素,且不是第一个,则用 $pre\_e$ 返回其前驱,否则操作失败.

***NextElem(L, cur\_e,&next\_e);***

初始条件: 线性表 $L$ 已存在

操作结果: 若 $cur\_e$ 是 $L$ 的数据元素,且不是最后一个,则用 $next\_e$ 返回其后继,否则操作失败.

.....



.....

**ListInsert(L,i,e);**

初始条件: 线性表L已存在,且 $1 \leq i \leq \text{listLength}(L)$

操作结果: 在L中第i个位置之前插入新的数据元素e,L长度加1.

**ListDelete(L,i);**

初始条件: 线性表L已存在,且 $1 \leq i \leq \text{listLength}(L)$

操作结果: 删除L的第i个元素,L的长度减1.

**TraverseList(L);**

初始条件: 线性表L已存在

操作结果: 对线性表L进行遍历,在遍历的过程中对L的每个结点访问1次.

**}ADT List.**



逻辑教育  
Logic education

## 线性表—单链表结点

结点



数据域

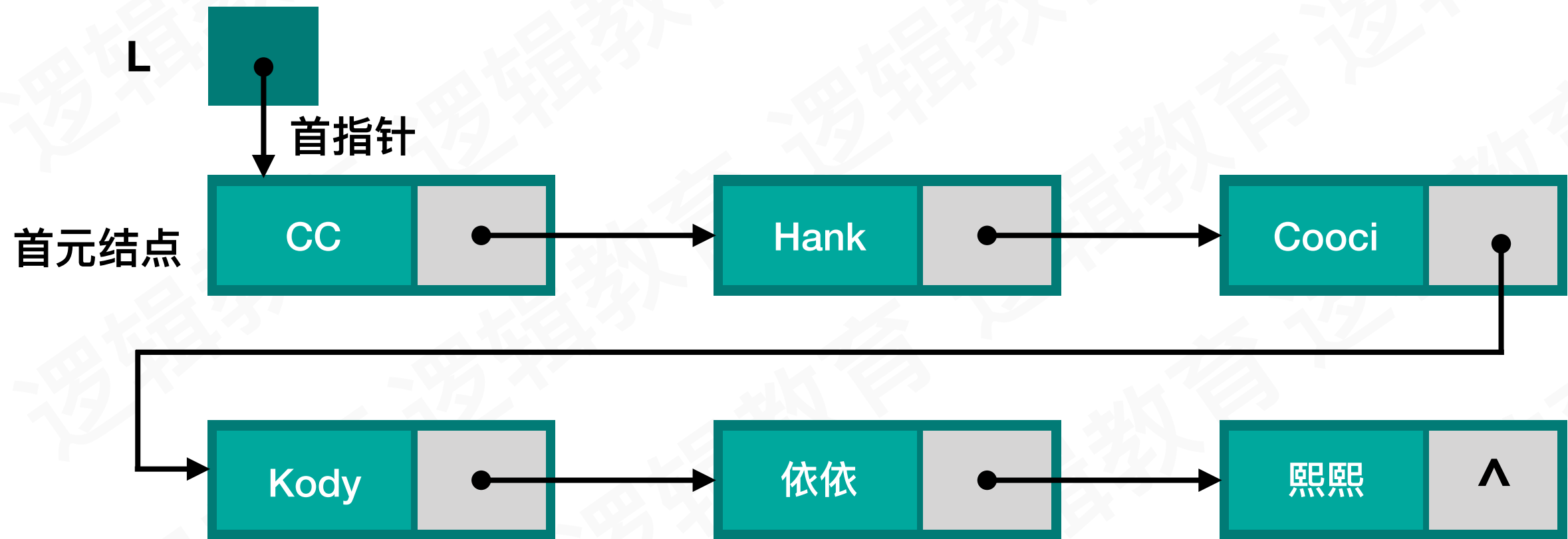
指针域

课程研发:CC老师  
课程授课:CC老师

转载需注明出处,不得用于商业用途.已申请版权保护



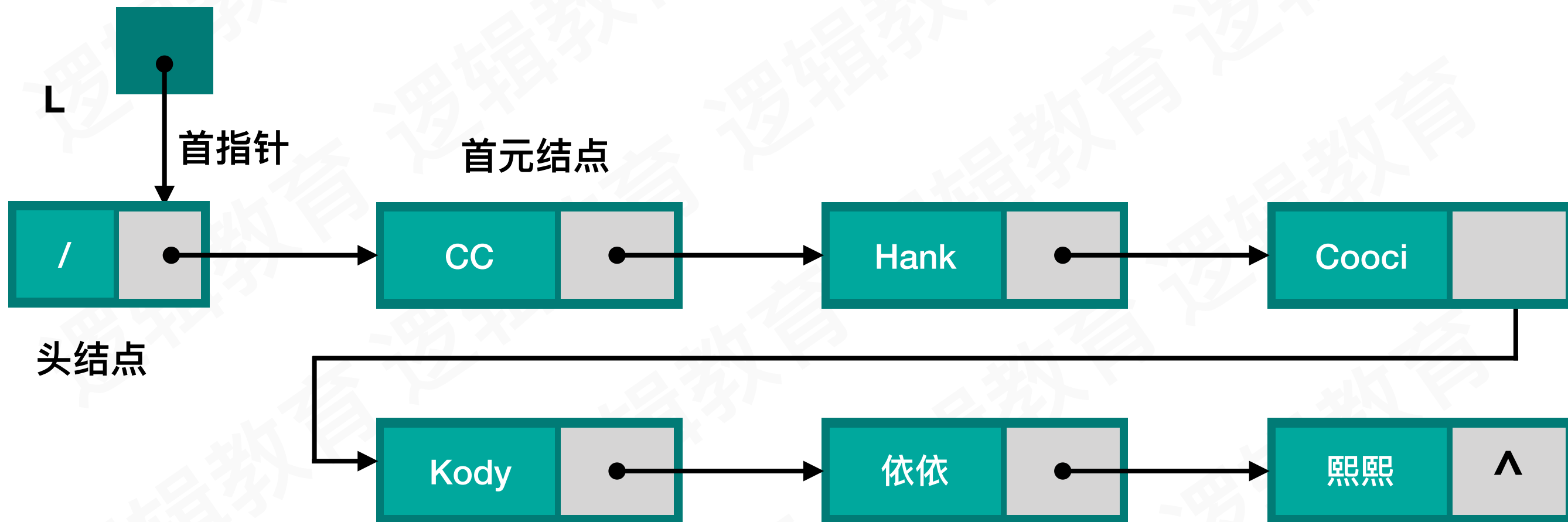
## 线性表—单链表逻辑状态







## 线性表—增加头结点的单链表逻辑状态

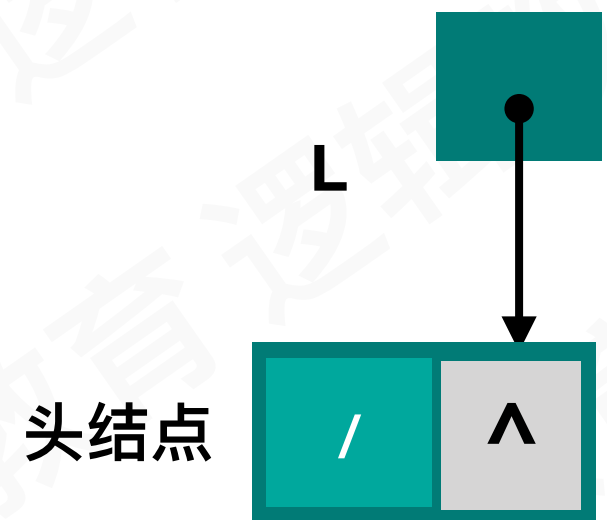




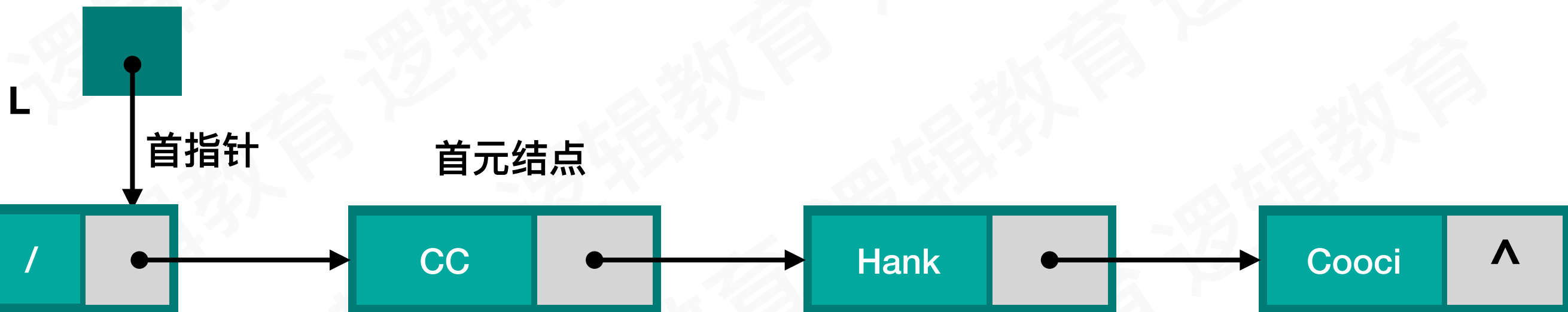


## 线性表—单链表为什么要增加头结点

1. 便于首元结点处理
2. 便于空表和非空表的统一处理.



(b) 空表



头结点

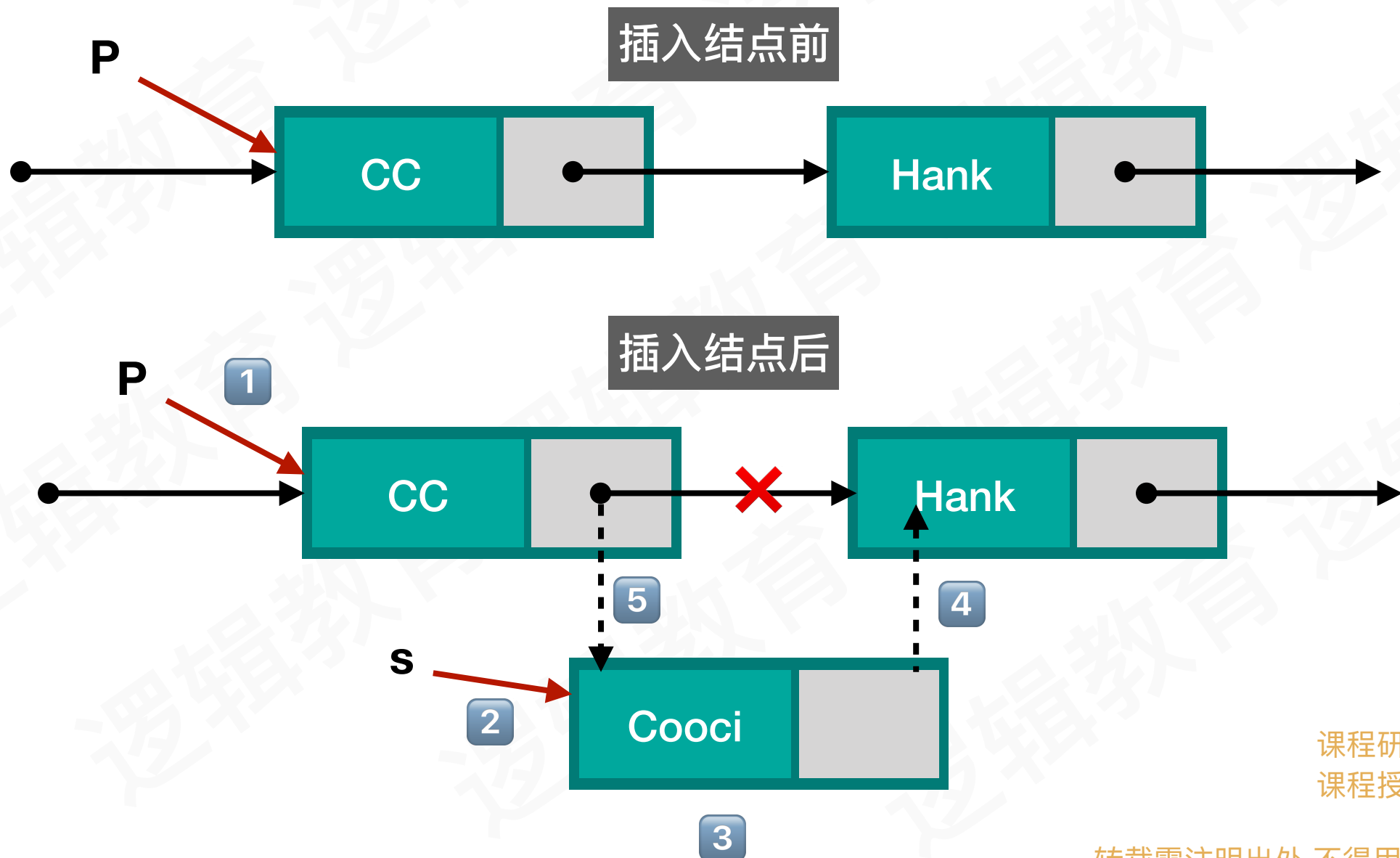
(a) 非空表

课程研发:CC老师  
课程授课:CC老师



## 线性表—单链表插入

假设要在单链表的两个数据元素a和b之间插入一个数据元素x,已知p为其单链表存储结构中指向结点a指针.如下图所示

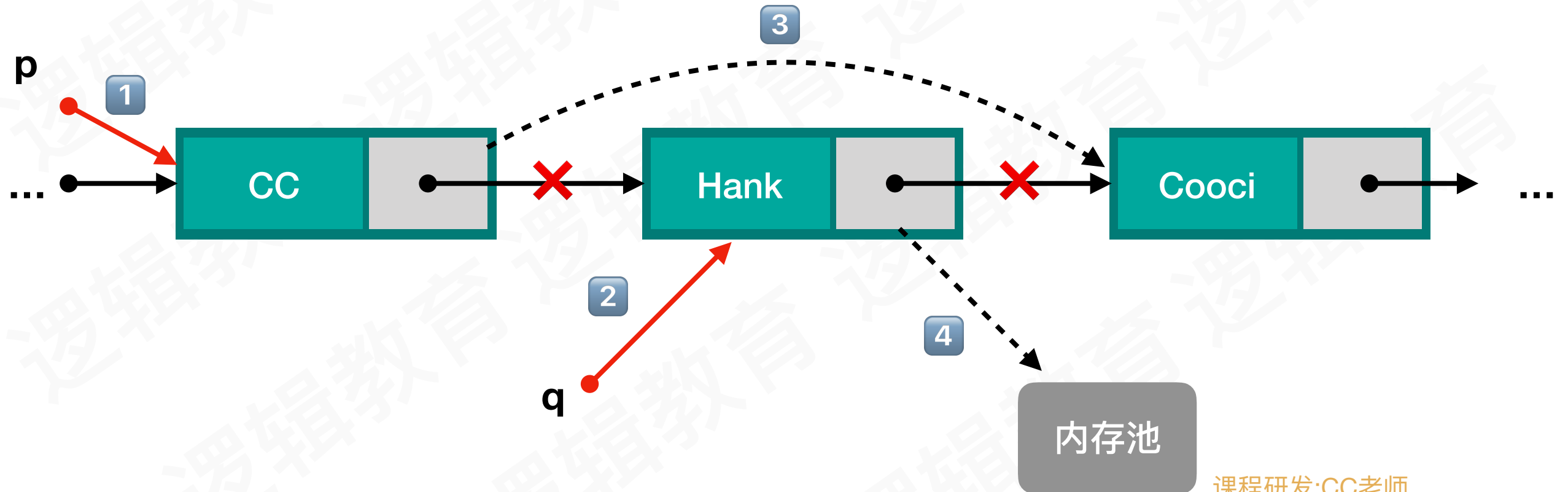


课程研发:CC老师  
课程授课:CC老师



## 线性表—单链表删除

要删除单链表中指定位置的元素,同插入元素一样; 首先应该找到该位置的前驱结点; 如下图所示在单链表中删除元素Hank时,应该首先找到其前驱结点CC. 为了在单链表中实现元素CC,Hank,Cooci 之间的逻辑关系的变化,仅需修改结点CC中的指针域即可. 假设p为指向结点CC的指针

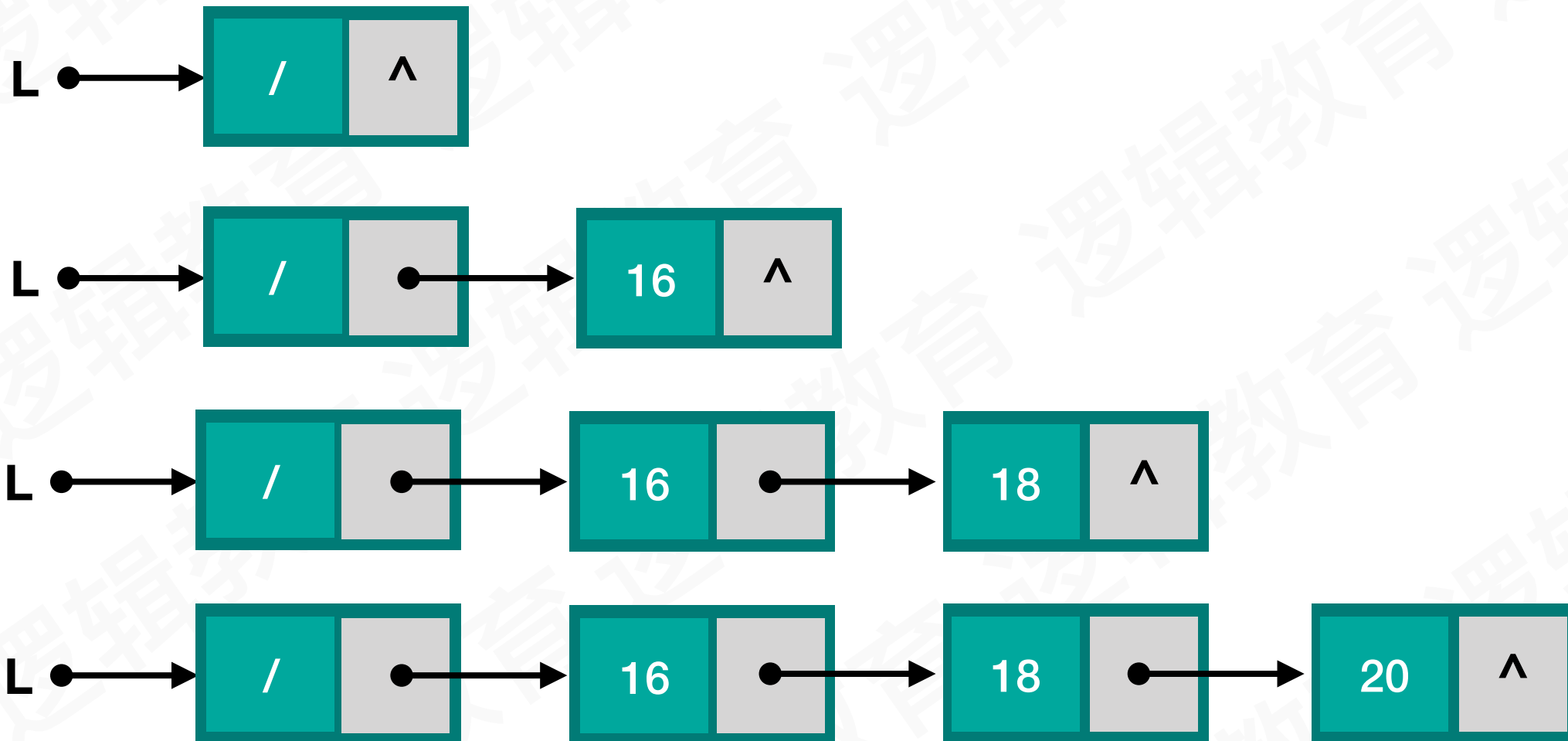


课程研发:CC老师  
课程授课:CC老师



## 线性表—单链表前插法

假设插入随机数



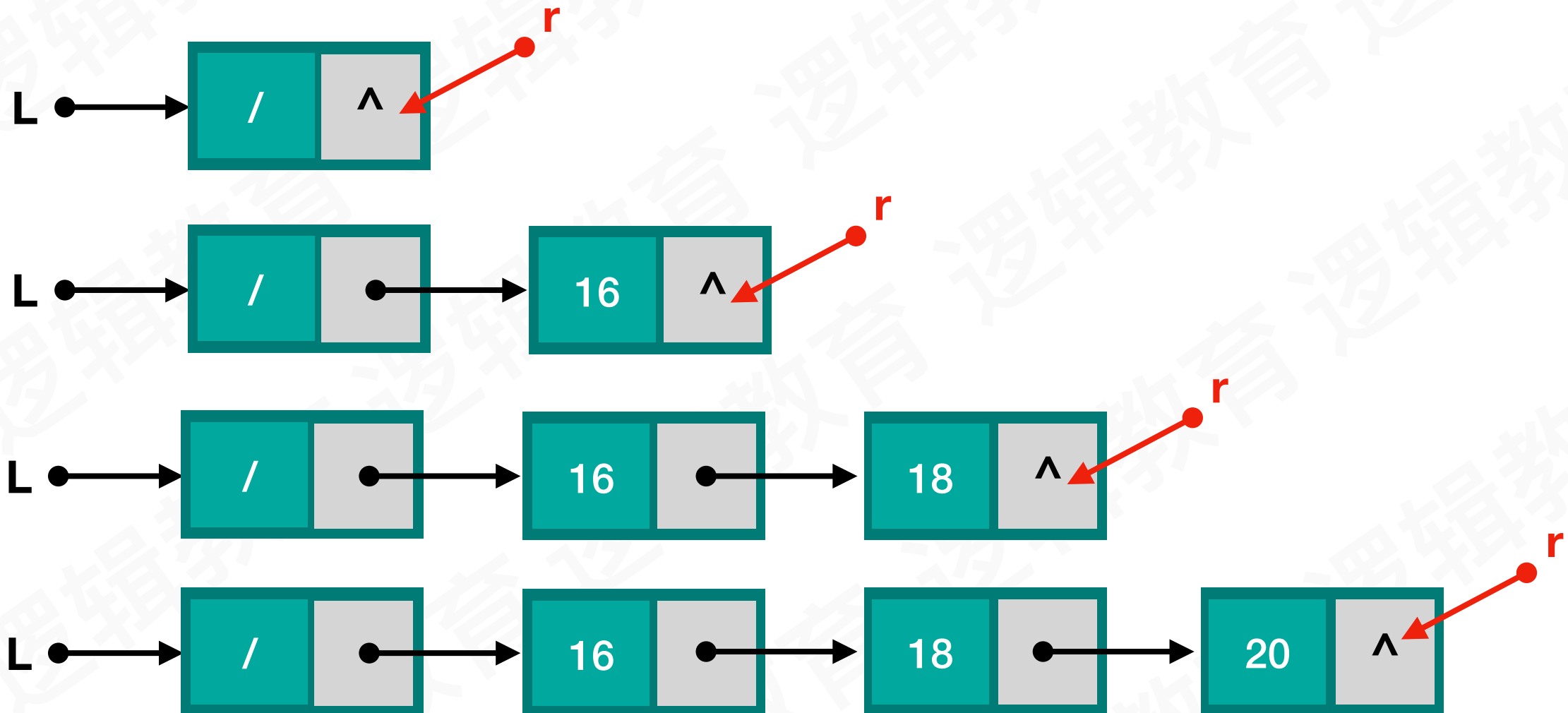
前插法创建单链表

课程研发:CC老师  
课程授课:CC老师



## 线性表—单链表后插法

假设插入随机数

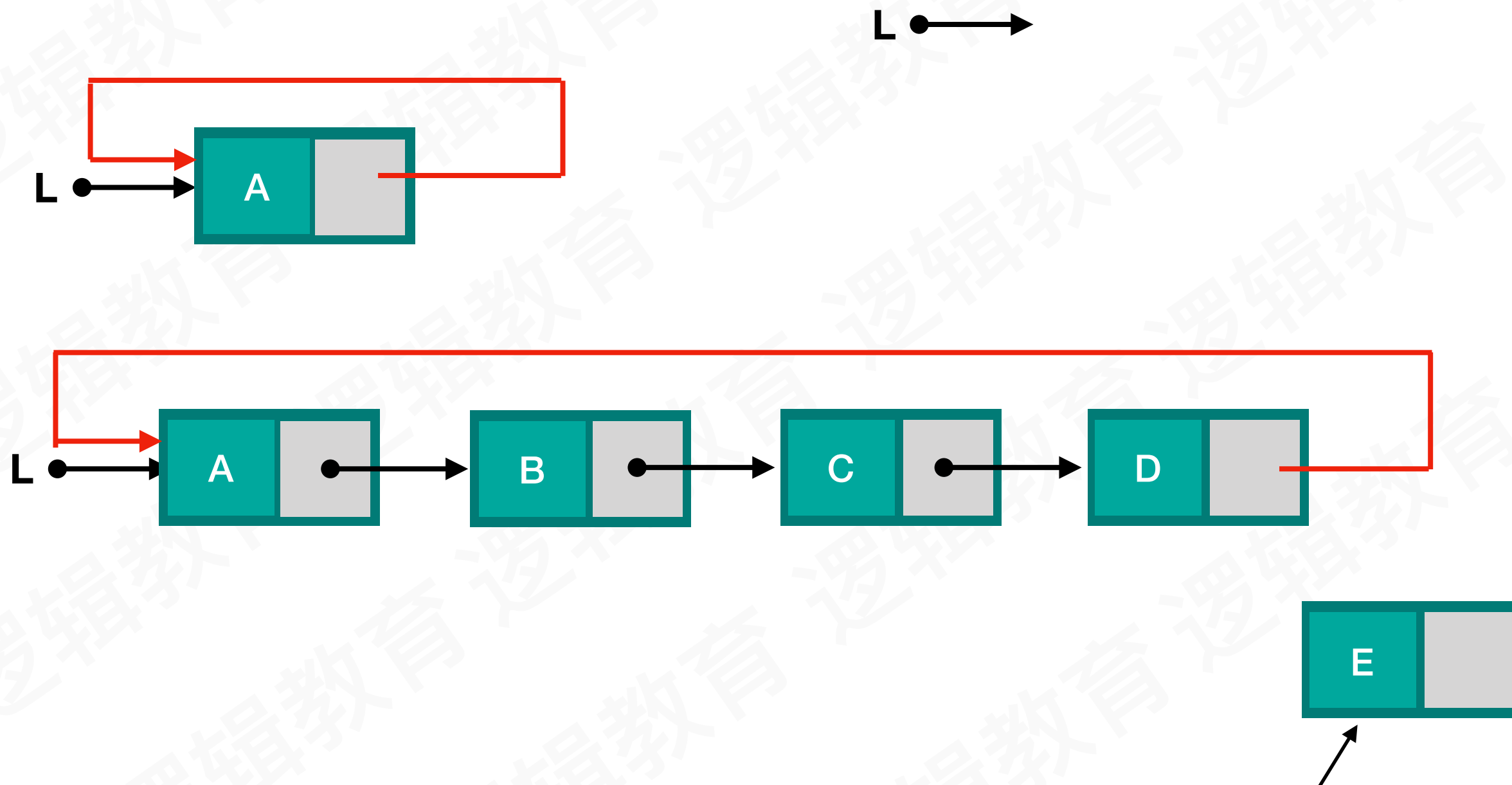


后插法创建单链表

课程研发:CC老师  
课程授课:CC老师

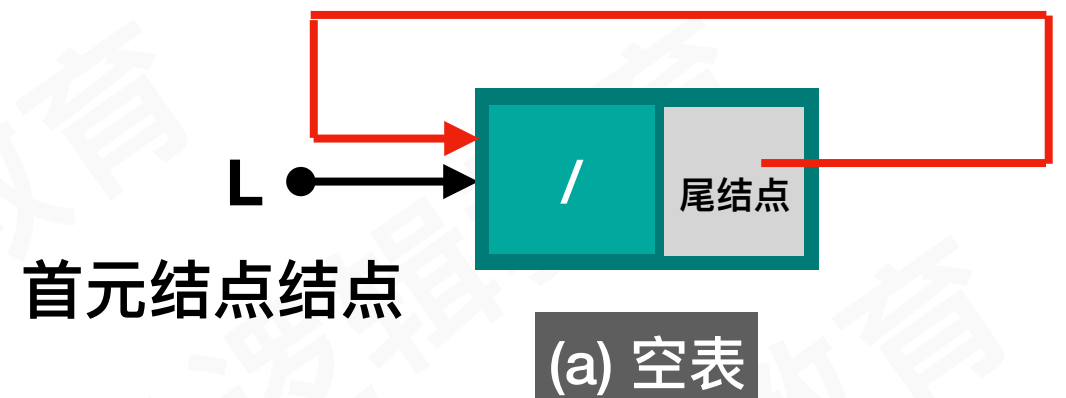


## 线性表—循环链表

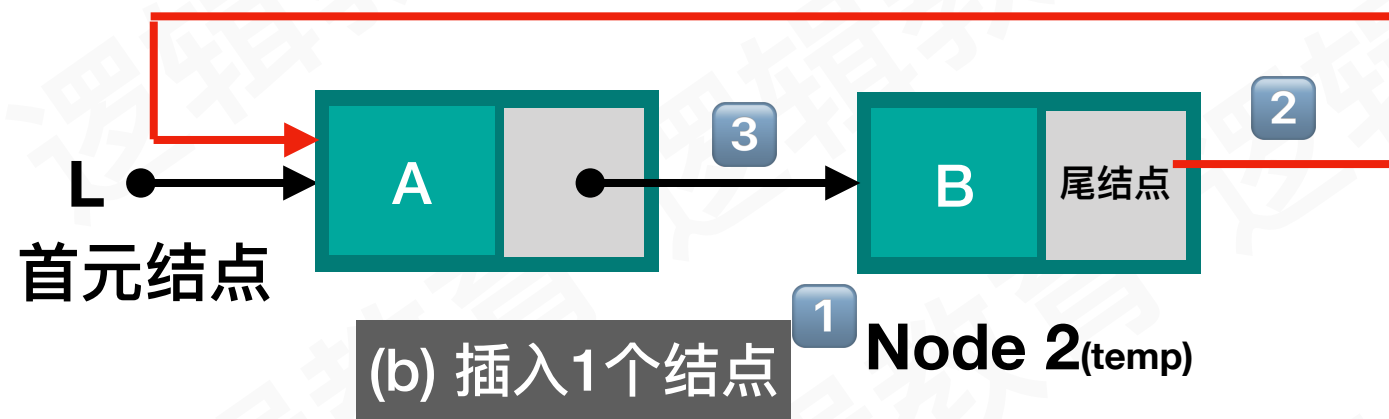


课程研发:CC老师  
课程授课:CC老师

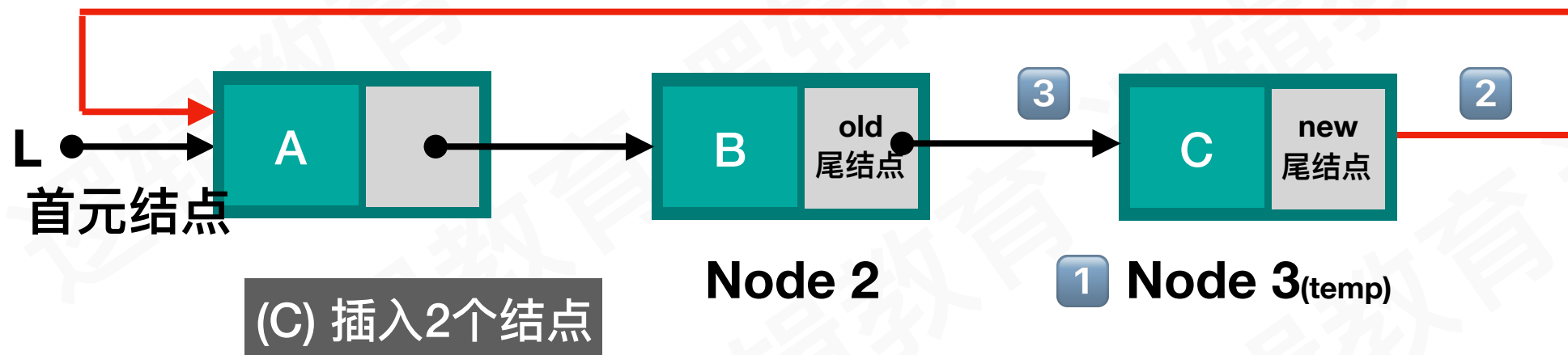
## 线性表—循环链表 初始化以及赋值



```
*p = (LinkList *)malloc(sizeof(Node));
(*p)->data = item;
(*p)->next = *p;
```



```
temp->data = item;
temp->next = *p;
target->next = temp
```



```
temp->data = item;
temp->next = *p;
target->next = temp
```

课程研发:CC老师  
课程授课:CC老师

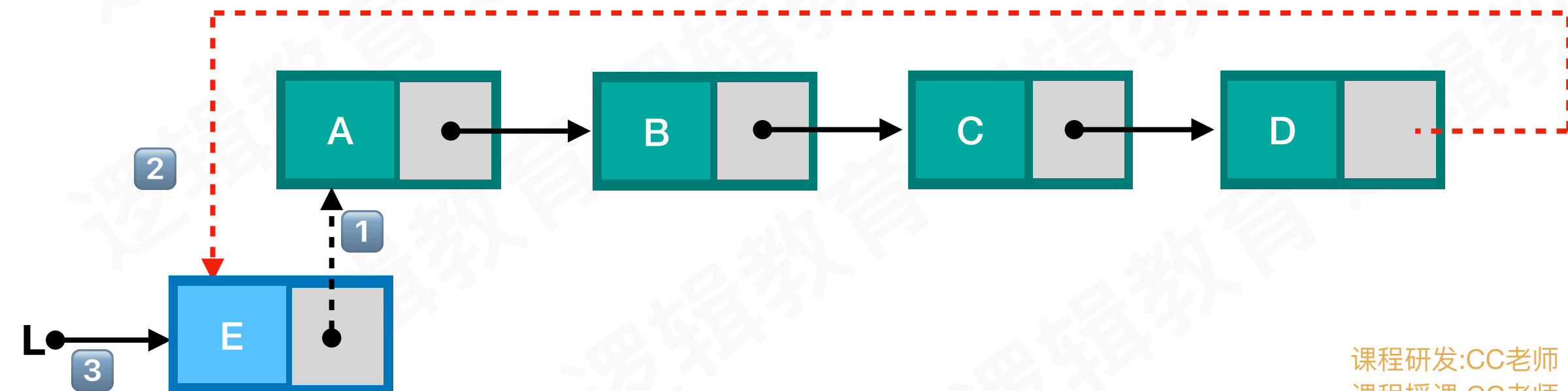




## 线性表—循环链插入数据

### 情况1 在插入位置在首节点上

- 1 判断插入位置是否在首元结点上
- 2 创建新结点,并赋值给新结点

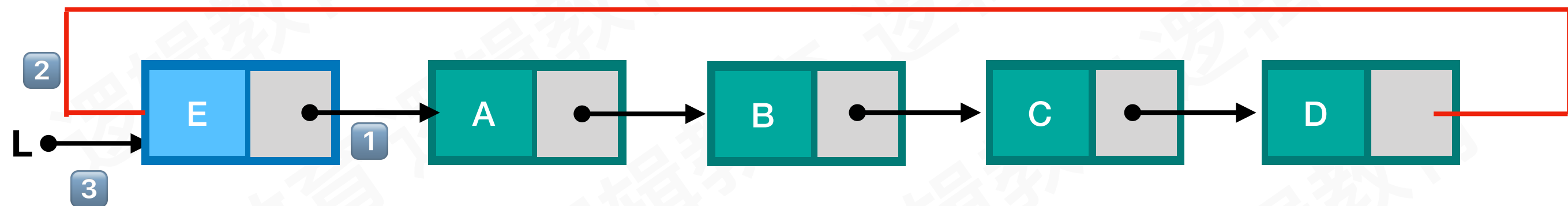


课程研发:CC老师  
课程授课:CC老师



## 线性表—循环链插入数据

情况1 在插入位置在首节点上





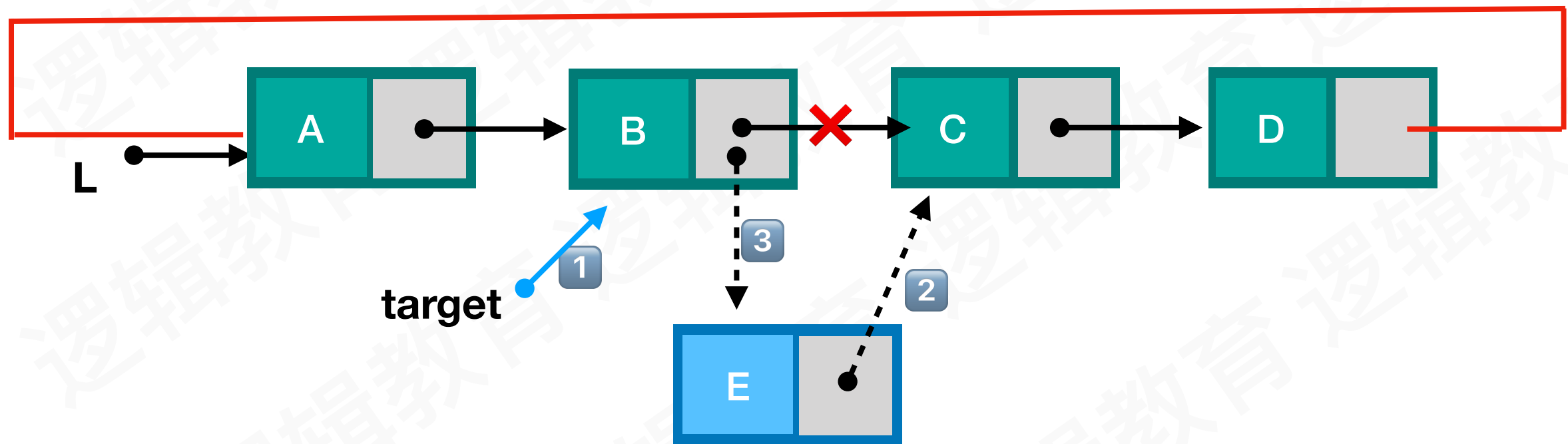
## 线性表—循环链插入数据

### 情况2 其他位置上

1 创建新结点,并赋值给新结点



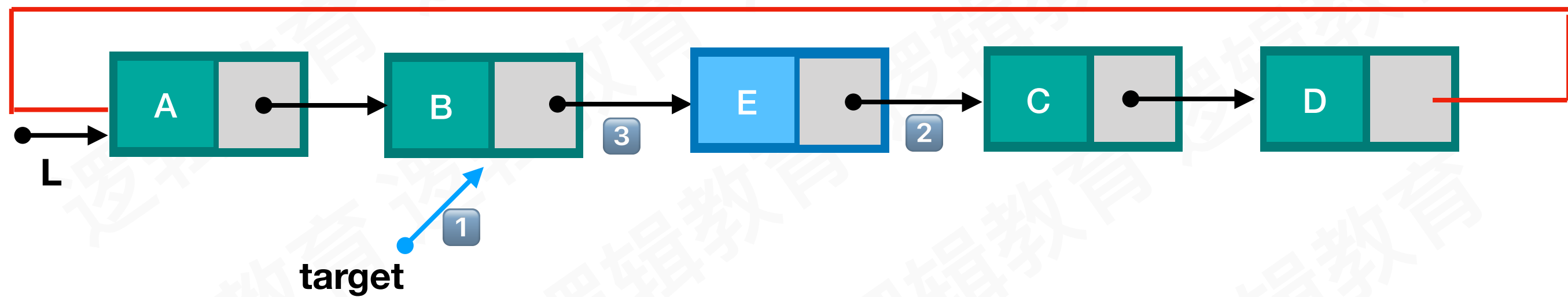
2 找到插入的位置的前一个结点 **target**





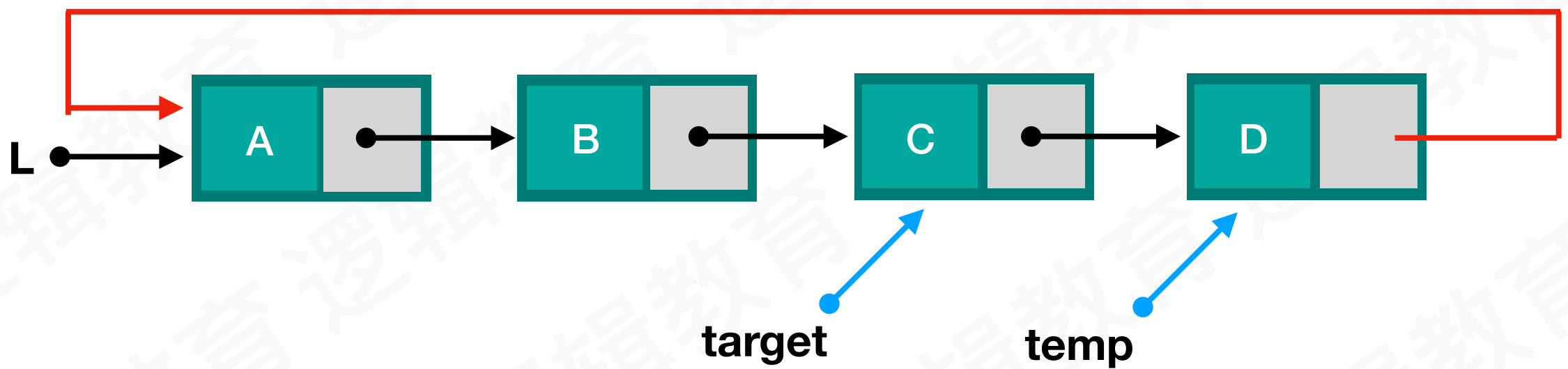
## 线性表—循环链插入数据

### 情况2 其他位置上



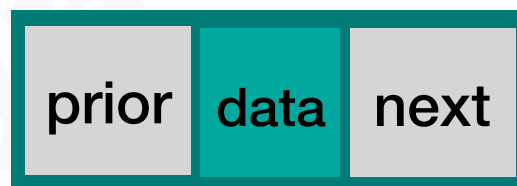


## 线性表—循环链删除数据





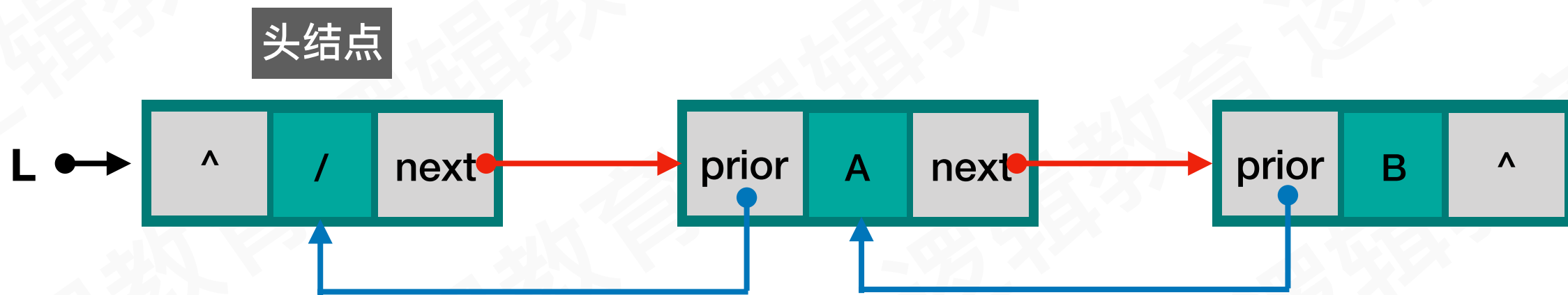
## 线性表—双向链表



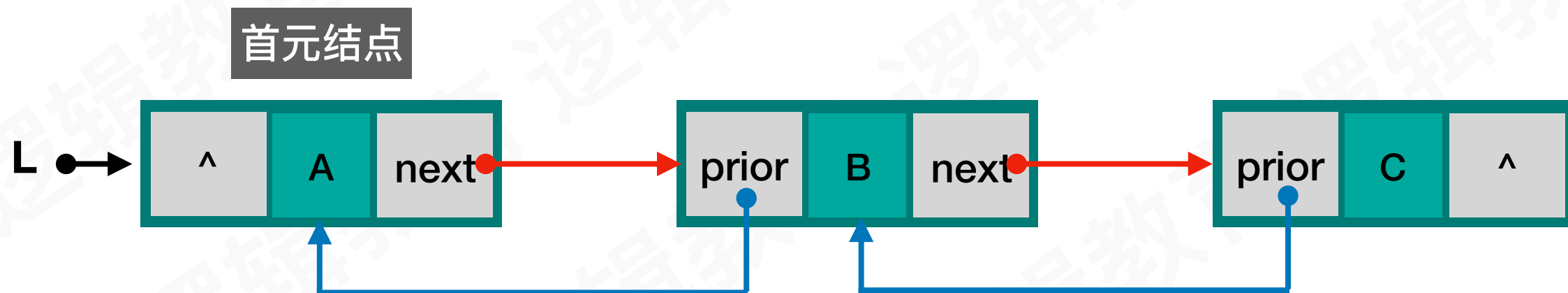
(a) 结点结构



## 线性表—双向链表



非空的双向链表-带头结点



非空的双向链表

课程研发:CC老师  
课程授课:CC老师



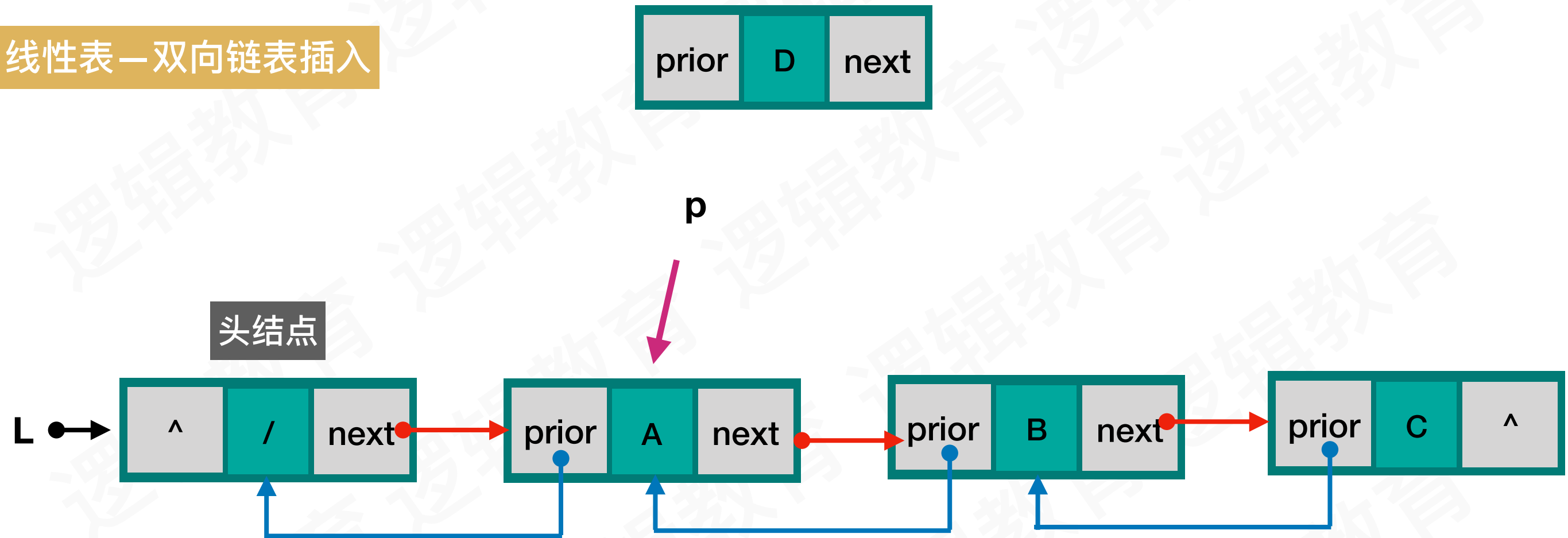


## 线性表—双向链表创建



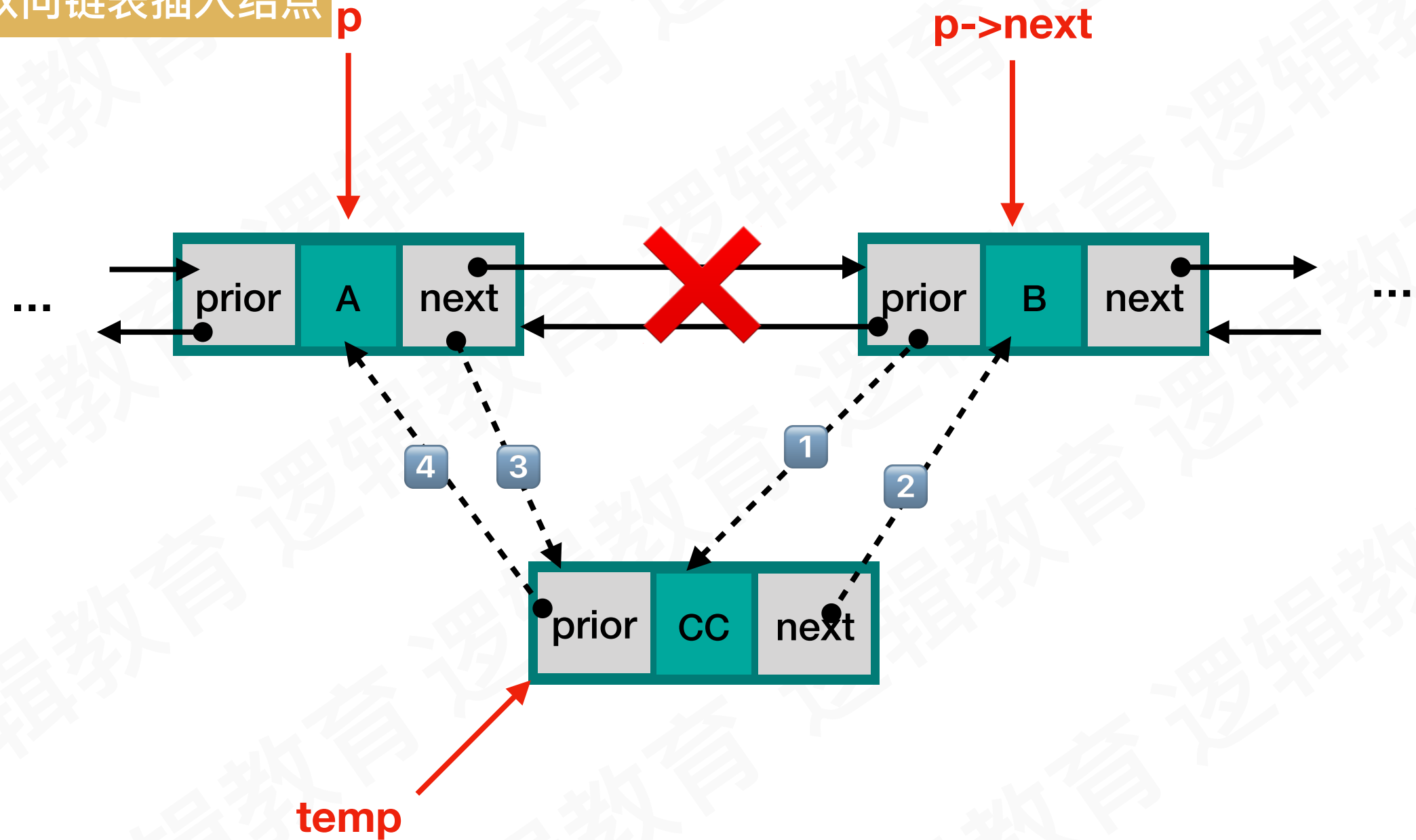


## 线性表—双向链表插入





## 线性表—双向链表插入结点

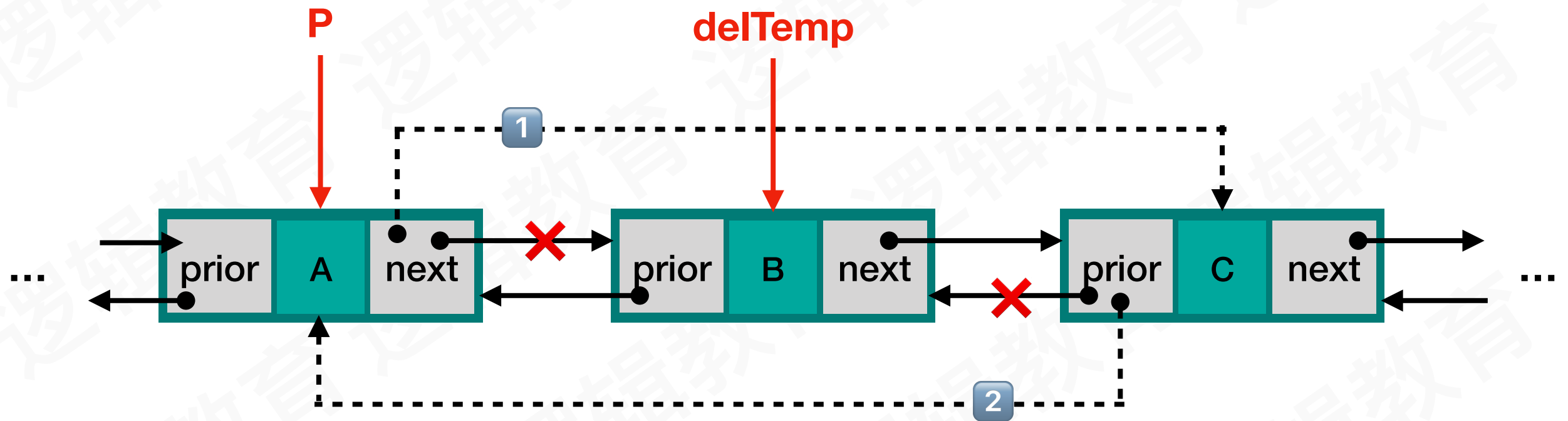


在双向链表中插入结点时指针变换情况图

课程研发:CC老师  
课程授课:CC老师



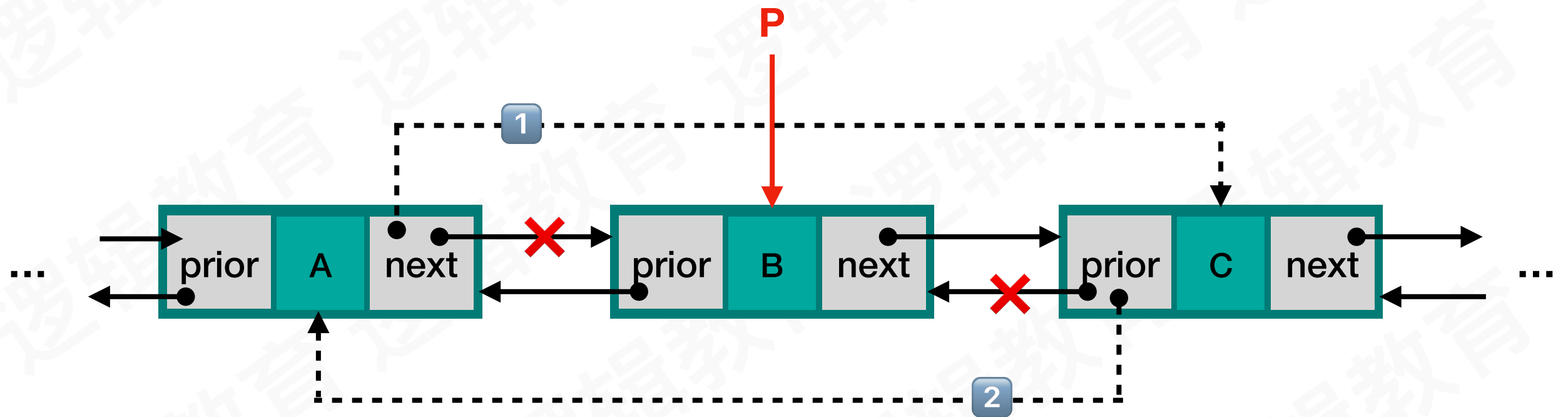
## 线性表—双向链表删除结点



在双向链表中删除结点时指针变换情况图



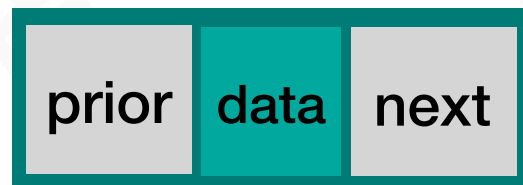
## 线性表—双向链表删除结点



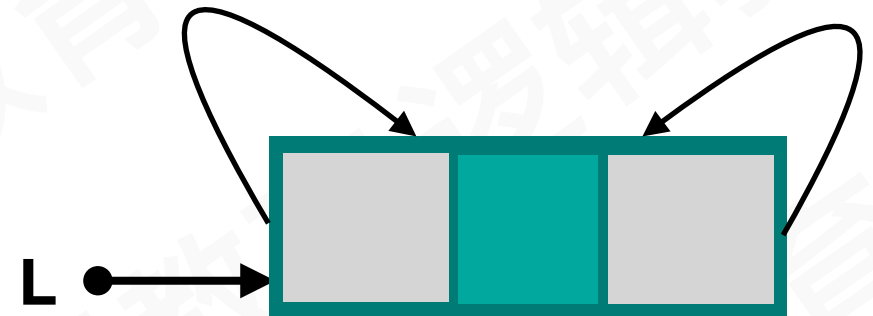
在双向链表中删除结点时指针变换情况图



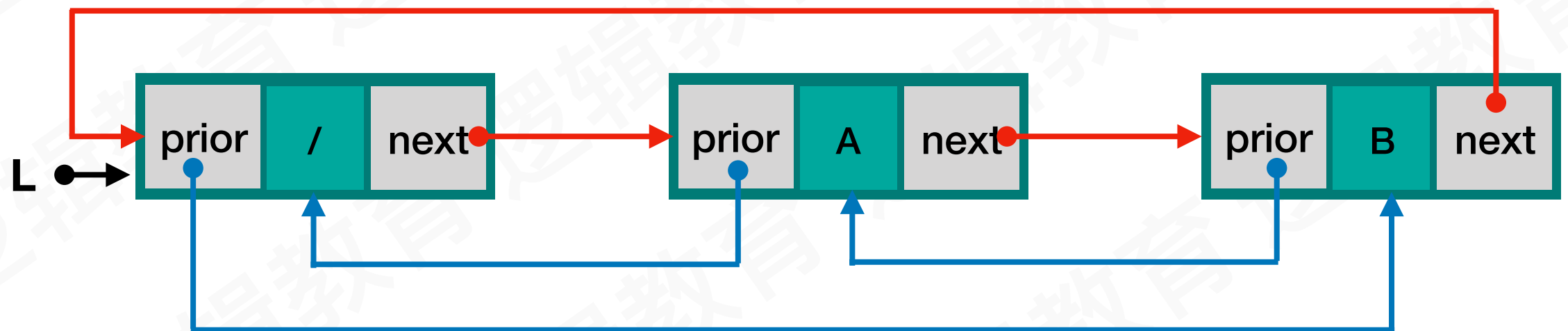
## 线性表—双向循环链表



(a) 结点结构



(b) 空的双向循环链表

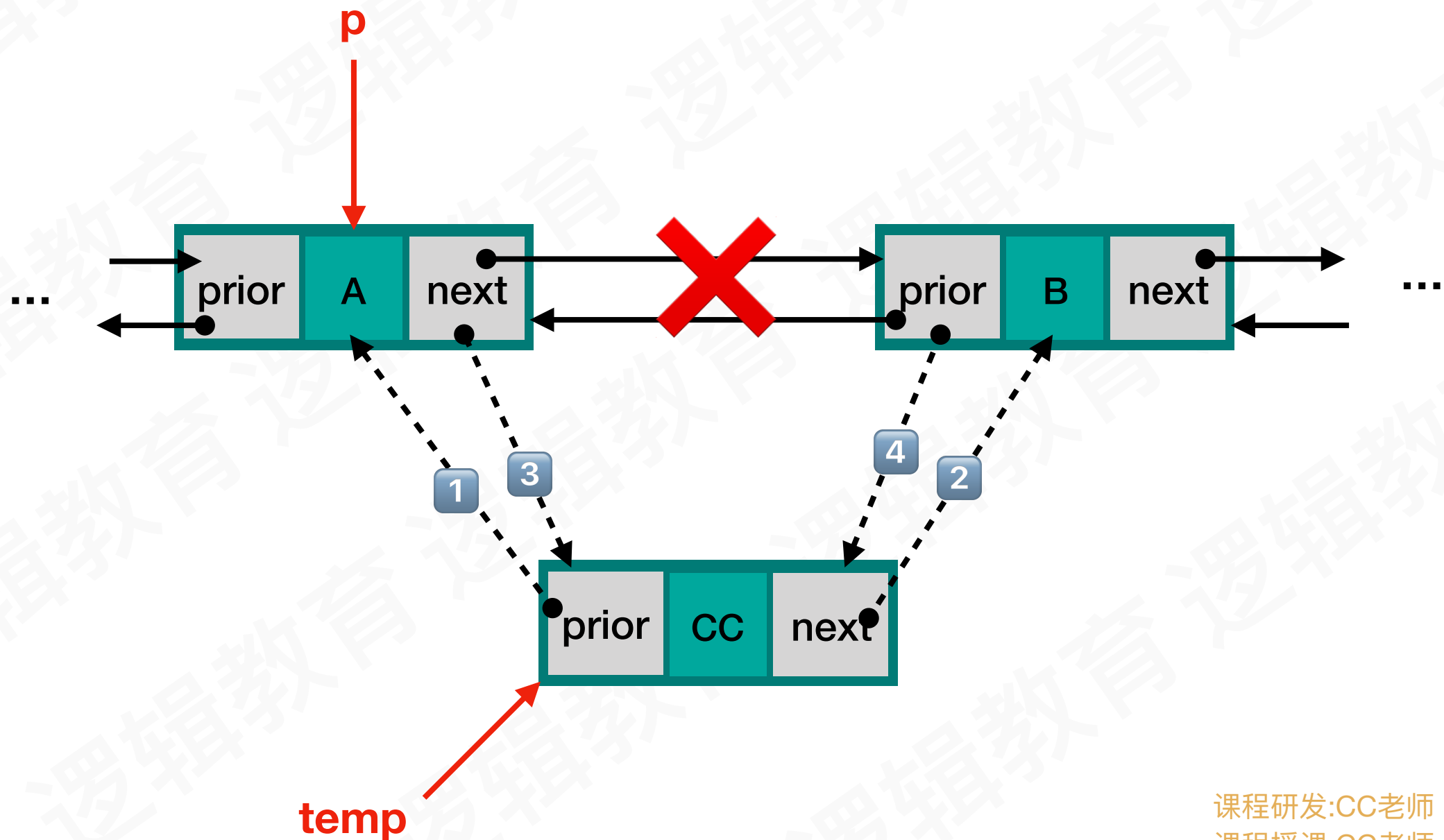


(c) 非空的双向循环链表

课程研发:CC老师  
课程授课:CC老师



## 线性表—双向循环链表插入结点



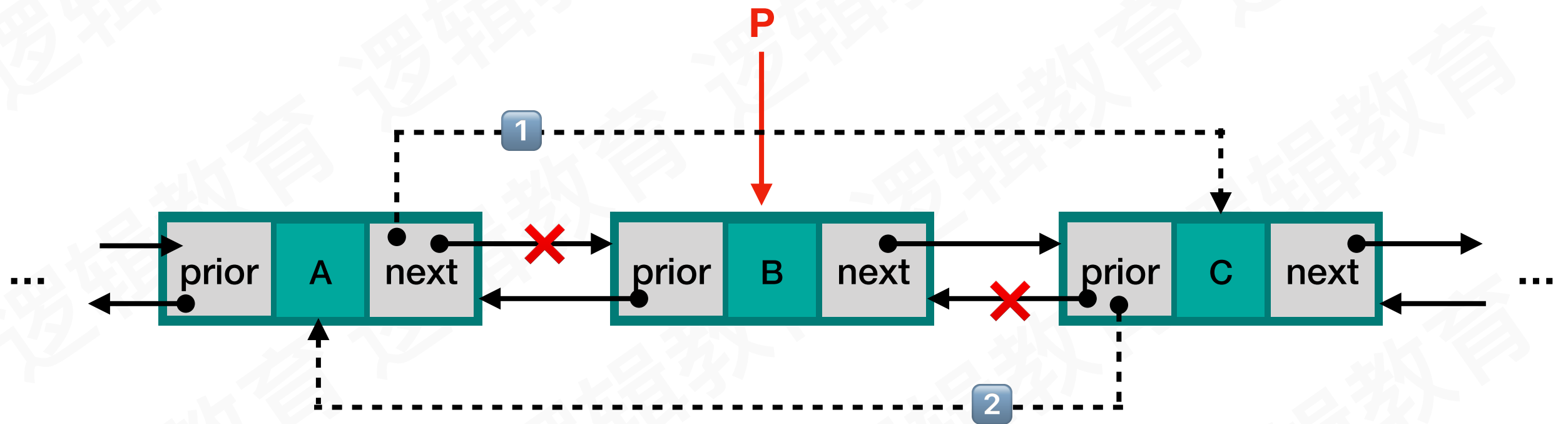
课程研发:CC老师  
课程授课:CC老师

在双向链表中插入结点时指针变换情况图





## 线性表—双向循环链表删除结点



在双向链表中删除结点时指针变换情况图



## 顺序表和链表的比较

### 空间性能比较

- (1). 存储空间分配
- (2). 存储密度的大小

$$\text{存储密度} = \frac{\text{数据元素本身占用存储量}}{\text{结点结构占用的存储量}}$$

### 时间性能比较

- (1). 存储元素的效率
- (2). 插入和删除操作的效率



## 线性表总结





逻辑教育  
Logic education

## 线性表算法设计—练习篇章

### 题目1：

将2个递增的有序链表合并为一个链表的有序链表；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

课程研发:CC老师  
课程授课:CC老师



## 线性表算法设计—练习篇章

### 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; 例如： $L_a = \{2, 4, 6, 8\}$ ;  $L_b = \{4, 6, 8, 10\}$ ;  $L_c = \{4, 6, 8\}$



## 线性表算法设计—练习篇章

### 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:  $L = \{10,8,6,4,2,0\}$ ;



逻辑教育  
Logic education

## 线性表算法设计—练习篇章

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;

课程研发:CC老师  
课程授课:CC老师

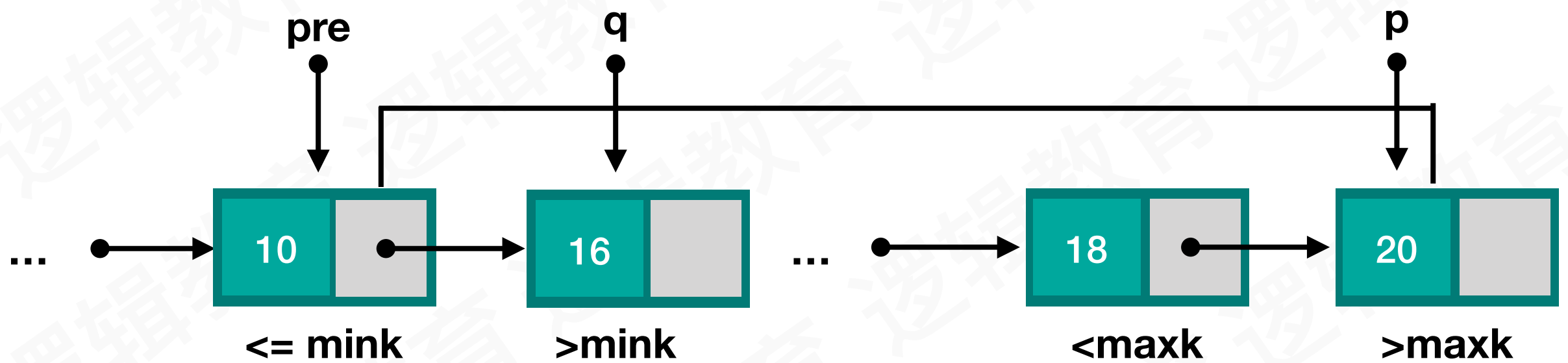
转载需注明出处,不得用于商业用途.已申请版权保护





## 线性表算法设计—练习篇章

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





## 线性表算法设计—练习篇章

### 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 $(x_p, x_{p+1}, \dots, x_{n-1}, x_0, x_1, \dots, x_{p-1})$ .

例如:  $\text{pre}[10] = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ,  $n = 10, p = 3$ ;  $\text{pre}[10] = \{3, 4, 5, 6, 7, 8, 9, 0, 1, 2\}$



## 线性表算法设计—练习篇章

### 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.



## 线性表算法设计—练习篇章

### 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 $A = \{21, -15, -7\}$ ;