王道计算机考研强化课

算法题备考

链表

链表的算法题备考

基本功训练：按位序查找

Key：循环遍历+计数器

```c
//定义单链表结点
typedef struct LNode{
    int data;
    struct LNode *next;
}LNode, *LinkList;

//求单链表的长度
int listLen(LinkList L){
    int length=0;     //计数器
    LNode * p= L->next;
    while (p!=NULL){
        length++;
        p=p->next;
    }
    printf("链表长度 = %d\n", length);
    return length;
}
```

```c
//返回单链表的中间结点
LNode * findMidNode(LinkList L){
    int length=0;     //计数器
    LNode * p= L->next;
    while (p!=NULL){
        length++;    //累加结点总数
        p=p->next;
    }
    int count=0;      //计数器
    p= L->next;       //从头遍历
    while (p!=NULL){
        count++;
        if (count==length/2)
            break;   //找到中间结点即可跳出循环
        p=p->next;
    }
    return p;
}
```

2009年真题

算法题

【2009 统考真题】已知一个带有表头结点的单链表，结点结构为

data	link

假设该链表只给出了头指针 list。在不改变链表的前提下，请设计一个尽可能高效的算法，查找链表中倒数第 k 个位置上的结点（k 为正整数）。若查找成功，算法输出该结点的 data 域的值，并返回 1；否则，只返回 0。要求：

1）描述算法的基本设计思想。

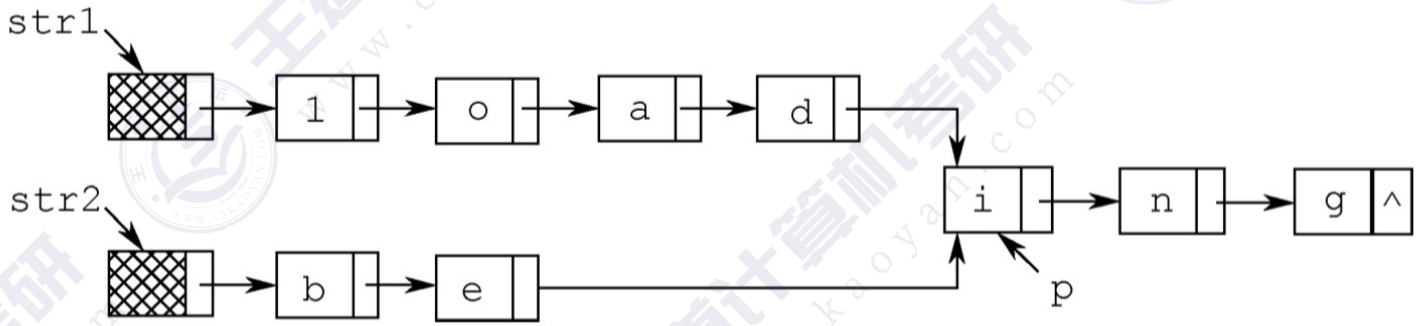
2）描述算法的详细实现步骤。

3）根据设计思想和实现步骤，采用程序设计语言描述算法（使用 C、C++或 Java 语言实现），关键之处请给出简要注释。

2012年真题

算法题

【2012 统考真题】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，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 所指向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。

基本功训练：按关键字条件查找+删除

Key：如果要插入或删除结点，可考虑前后指针

在带头结点的单链表 L 中，删除所有值为 x 的结点，并释放其空间，假设值为 x 的结点不唯一，试编写算法以实现上述操作。

基本功训练：按关键字条件查找+删除

Key：如果要插入或删除结点，可考虑前后指针

在带头结点的单链表 L 中，删除所有值为 x 的结点，并释放其空间，假设值为 x 的结点不唯一，试编写算法以实现上述操作。

```
//删除值为 x 的结点
void deletX(LinkList L, int x){
    LNode * pre = L;     //pre指向p的前驱结点
    LNode * p= pre->next; //p指向下一个要处理的结点
    while (p!=NULL){
        //对当前结点p进行处理
        if(p->data == x){
            LNode *q = p; //删除并释放值为x的结点
            p = p->next;  //p指向后一个结点
            pre->next = p;//修改前驱结点的next指针
            free(q);
        } else {
            pre = p;     // pre、p 后移
            p=p->next;
        }
    }
}
```

基本功训练：按关键字条件查找+插入

Key：如果要插入或删除结点，可考虑前后指针

在一个关键字递增有序的单链表中插入新关键字x，需确保插入后单链表保持递增有序。

基本功训练：按关键字条件查找+插入

Key：如果要插入或删除结点，可考虑前后指针

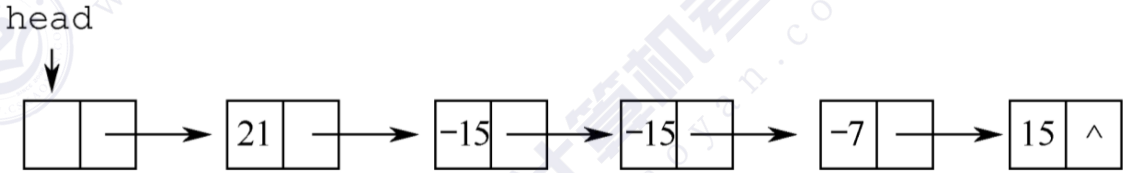
在一个关键字递增有序的单链表中插入新关键字x，需确保插入后单链表保持递增有序。

```
//在递增有序链表中插入值为 x 的结点
void InsertX(LinkList L, int x){
    LNode * pre = L;     //pre指向p的前驱结点
    LNode * p= pre->next; //p指向下一个要处理的结点
    while (p!=NULL){
        //对当前结点p进行处理
        if(p->data > x){
            break;  //x应插入到p的前面
        } else {
            pre = p;     // pre、p 后移
            p=p->next;
        }
    }
    LNode *q = (LNode *) malloc(sizeof(LNode));
    q->data = x;
    q->next = p;
    pre->next = q;
}
```

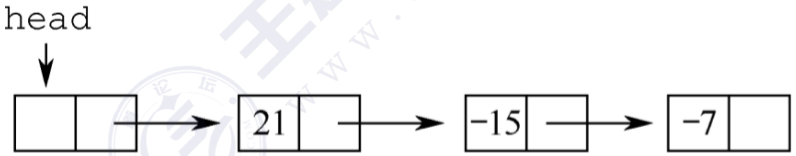
2015年真题

算法题

【2015 统考真题】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。

基本功训练：头插法（原地逆置）

试编写算法将带头结点的单链表就地逆置，所谓“就地”是指辅助空间复杂度为 $O(1)$。

```
//头插法实现链表的原地逆置
void ListReserve(LinkList L){
    //分配一个辅助头结点
    LinkList head = (LNode *) malloc(sizeof(LNode));
    head->next = NULL;

    while (L->next != NULL){
        LNode * p = L->next;    //按顺序拆下每个结点
        L->next = L->next->next;
        p->next = head->next;   //头插法
        head->next = p;
    }
    L->next = head->next;
    free(head);      //释放辅助头结点
}
```

基本功训练：尾插法（保持原序）

设 $C=\{a_1, b_1, a_2, b_2, \cdots, a_n, b_n\}$ 为线性表，采用带头结点的单链表存放，设计一个就地算法，将其拆分为两个线性表，使得 $A=\{a_1, a_2, \cdots, a_n\}$，$B=\{b_n, \cdots, b_2, b_1\}$。

基本功训练：尾插法（保持原序）

设 $C=\{a_1, b_1, a_2, b_2, \cdots, a_n, b_n\}$ 为线性表，采用带头结点的单链表存放，设计一个就地算法，将其拆分为两个线性表，使得 $A=\{a_1, a_2, \cdots, a_n\}$，$B=\{b_n, \cdots, b_2, b_1\}$。

```
LinkList A=NULL;
LinkList B=NULL;
```

```
void func(LinkList C){
    //分配A、B两个头结点
    A = (LNode *) malloc(sizeof(LNode));
    A->next = NULL;
    LNode * tailA = A;  //tailA 指向A的链尾
    B = (LNode *) malloc(sizeof(LNode));
    B->next = NULL;
    int count=1;    //计数器
    while (C->next != NULL){
        LNode * p = C->next;    //按顺序拆下每个结点
        C->next = C->next->next;
        if(count%2==1){ //奇数结点，插入A的链尾
            tailA->next = p;    //尾插法
            p->next = NULL;
            tailA = p;
        } else {        //偶数结点，插入B的链头
            p->next = B->next;  //头插法
            B->next = p;
        }
        count++;    //计数器+1
    }
}
```

2019年真题

算法题

【2019 统考真题】设线性表 $L=(a_1,a_2,a_3,\cdots,a_{n-2},a_{n-1},a_n)$ 采用带头结点的单链表保存，链表中的结点定义如下：

```
typedef struct node
{   int data;
    struct node*next;
}NODE;
```

请设计一个空间复杂度为 $O(1)$且时间上尽可能高效的算法，重新排列 L 中的各结点，得到线性表 $L'=(a_1,a_n,a_2,a_{n-1},a_3,a_{n-2},\cdots)$。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++语言描述算法，关键之处给出注释。

3）说明你所设计的算法的时间复杂度。