

Lab 9 Part I 链表

Problem 1. 实现链表类LinkedList。节点类ListNode定义和链表类LinkedList声明已给出，需实现插入节点函数、删除节点函数、创建链表函数、查找函数、打印链表函数、反转链表函数、删除链表倒数第N个节点函数、查找链表中间节点、链表升序排序函数、删除排序链表中的重复元素函数。

```
class ListNode
{
public:
    ListNode() : val(0), next(0) {}
    ListNode(int x) : val(x), next(0) {}
    int val;
    ListNode *next;
}; //节点类定义

class LinkedList
{
public:
    LinkedList(); //默认构造函数，创建head节点
    ~LinkedList(); //析构函数，删除所有节点，包括head节点
    void insert(int position, int element); //向链表指定位置前添加新的节点
    void remove(int position); //移除链表指定位置的节点
    void createList(); //从键盘上读入若干个int数，基于这些数创建一个链表
    int find(int position); //查找指定位置的结点，返回结点的值
    void print(); //打印链表，例：[1,2,3,4]: 1 2 3 4
    void reverseList(); //反转链表，例：[1,2,3,4]->[4,3,2,1]
    void removeNthFromEnd(int n); //删除链表的倒数第N个结点，例：N=2, [1,2,3,4]->
    [1,2,4], 要求只能遍历一次链表
    int findMiddle(); //查找链表的中间节点，返回结点的值，要求只能遍历一次链表
    void sortList(); //链表升序排序，例：[5,4,9,1]->[1,4,5,9]，注：只允许在链表上进行操作，不允许将数据读到数组里进行排序
    void deleteDuplicates(); //删除排序链表中的重复元素，例：[1,1,1,2,3]->[1,2,3]
private:
    ListNode *head; //指向链表头节点
}; //链表类
```

Problem 2. 基于Problem 1定义的LinkedList类，创建两个有序的链表l1和l2，编写函数，将这两个有序的单链表合并成一个有序链表。

Problem 3. 基于Problem 1定义的ListNode和LinkedList类，编写函数创建一个带环的链表（将链表的最后一个节点的next指向中间的某个节点）；编写另外一个函数，检查任意给定的一个链表是否有环。

Problem 4. 给定下面的ListNode节点类，注意节点里面有两个指针next和random；编写函数，创建一个复杂链表，其中节点的next指针指向后面一个节点，random指针随机指向链表中的一个节点或者是NULL；编写另外一个函数，复制所创建的复杂链表（即创建另外一个复杂链表，其中节点之间的连接关系和给定的复杂链表一模一样）。

```
class ListNode
{
public:
    int val;
    ListNode *next;
    ListNode *random;
}; //节点类定义
```