项目说明文档

数据结构课程设计

——两个有序链表序列的交集

作 者 姓 名： 邓泉

学 号： 1953871

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1分析](#_Toc530859342)

[1.1 背景分析](#_Toc530859343)

[1.2 功能分析](#_Toc530859344)

[2 设计](#_Toc530859345)

[2.1 数据结构设计](#_Toc530859346)

[2.2 类结构设计](#_Toc530859347)

[2.3 成员与操作设计](#_Toc530859348)

[2.4 主程序设计](#_Toc530859349)

[3.功能实现](#_Toc530859350)

[3.1 建立链表功能](#_Toc530859351)

3.1.1 创建功能流程图.........................................................................................

[3.1.2 创建功能核心代码](#_Toc495668164)

[3.2 求交集功能](#_Toc530859352)

3.2.1 求交集功能流程图.....................................................................................

[3.2.2 求交集功能核心代码](#_Toc495668164)

3.3 重载=运算符.........................................................................................................

3.3.1 重载=运算符流程图..................................................................................

[3.3.2 重载=运算符核心代码](#_Toc495668164)

[3.4 输出功能](#_Toc530859353)

3.4.1 输出功能流程图.........................................................................................

[3.4.2 输出功能核心代码](#_Toc495668164)

[4 测试](#_Toc530859354)

[4.1 功能测试](#_Toc530859355)

[4.1.1 一般情况测试](#_Toc530859356)

[4.1.2 交集为空的情况测试](#_Toc530859357)

[4.1.3 完全相交测试](#_Toc530859358)

[4.1.4 其中一个序列完全属于交集的情况](#_Toc530859359)

[4.1.5 其中一个序列为空的情况](#_Toc530859360)

# 1分析

## 1.1 背景分析

求两个有序序列的交集在现实生活中有广泛和深远的意义。与无序序列求交集相比，有序序列求交集的操作复杂度更低，速度更快。可是在代码实现中如果不能充分利用输入序列有序的特点，程序将失去价值。

## 1.2 功能分析

该程序的输入是两个非降序序列，而其输出是这两个序列的交集。当两个序列的交集是空集时，输出为“NULL”。

# 2 设计

## 2.1 数据结构设计

明确要求代码实现的应是两个有序链表序列的交集。链表作为一种常用的顺序存储结构， 在插入、删除上较线性表有着很大的优势，而在遍历时较线性表慢一些。采用带有附加头结点的单链表。

## 2.2 类结构设计

本系统使用的链表由链表类（LinkedSet）和链表结点（SetNode）组成。为方便处理，采用struct描述结点类（SetNode），这样使得链表类（LinkedSet）可以直接访问链表结点。

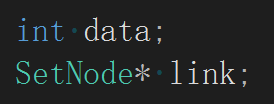
结点类含有两个成员，分别是该结点的数据域（data）和指向下一结点的指针域（link）。

链表类包含两个私有成员，分别是指向头结点的指针（first）和指向尾结点的指针（last）。

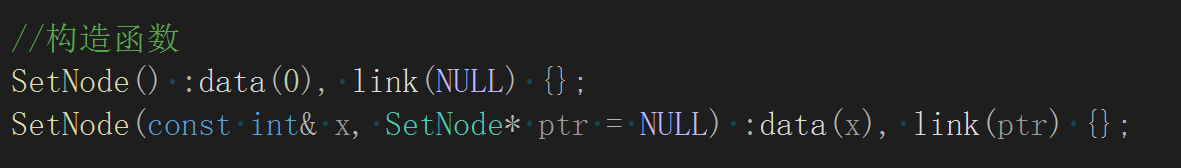
## 2.3 成员与操作设计

链表结点（SetNode）

公有成员：

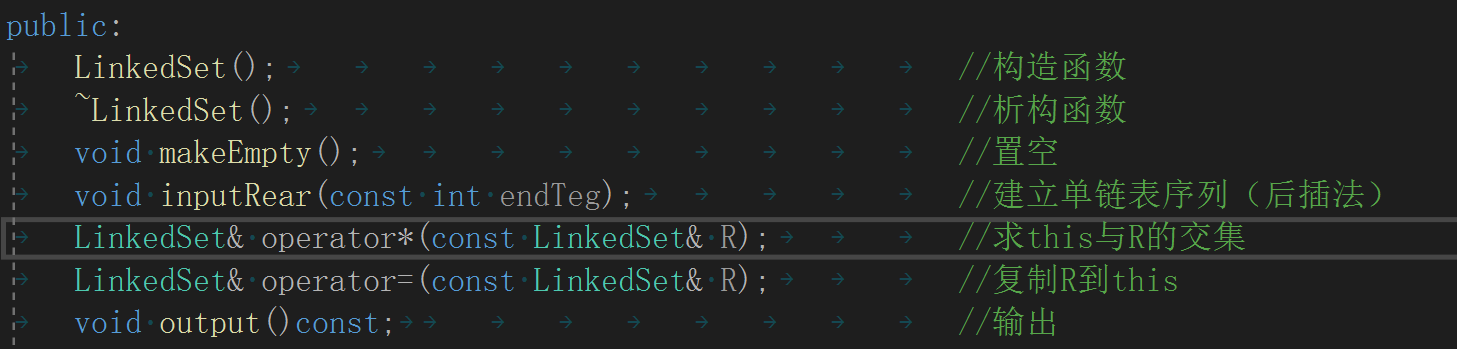


公有操作：



链表类（Linklist）

私有成员：



公有操作：

1609086320(1)

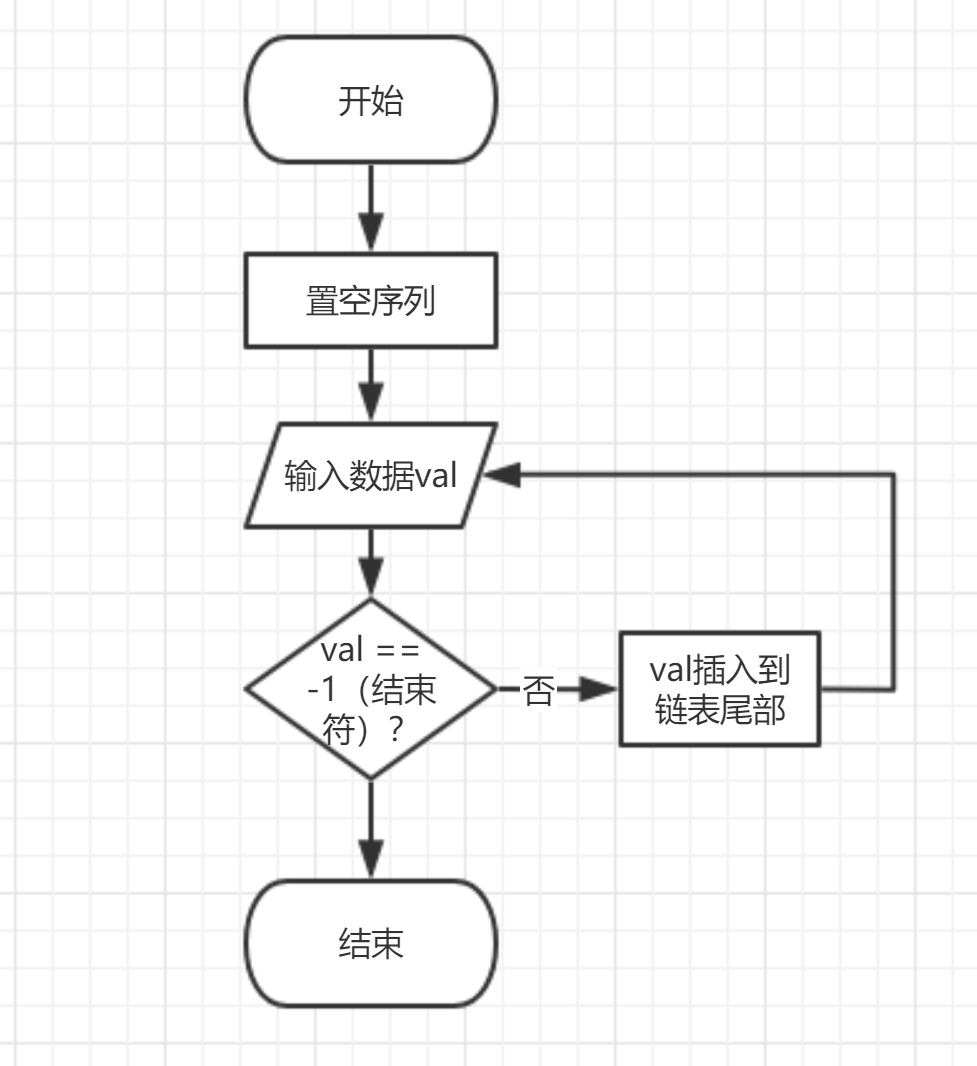
## 2.4 主程序设计

系统接受两个非降序序列作为输入，并给出这两个序列的交集作为输出。如果两个序列的交集为空集，则输出“NULL”。

# 3功能实现

## 3.1建立链表功能的实现

3.1.1 建立链表功能流程图

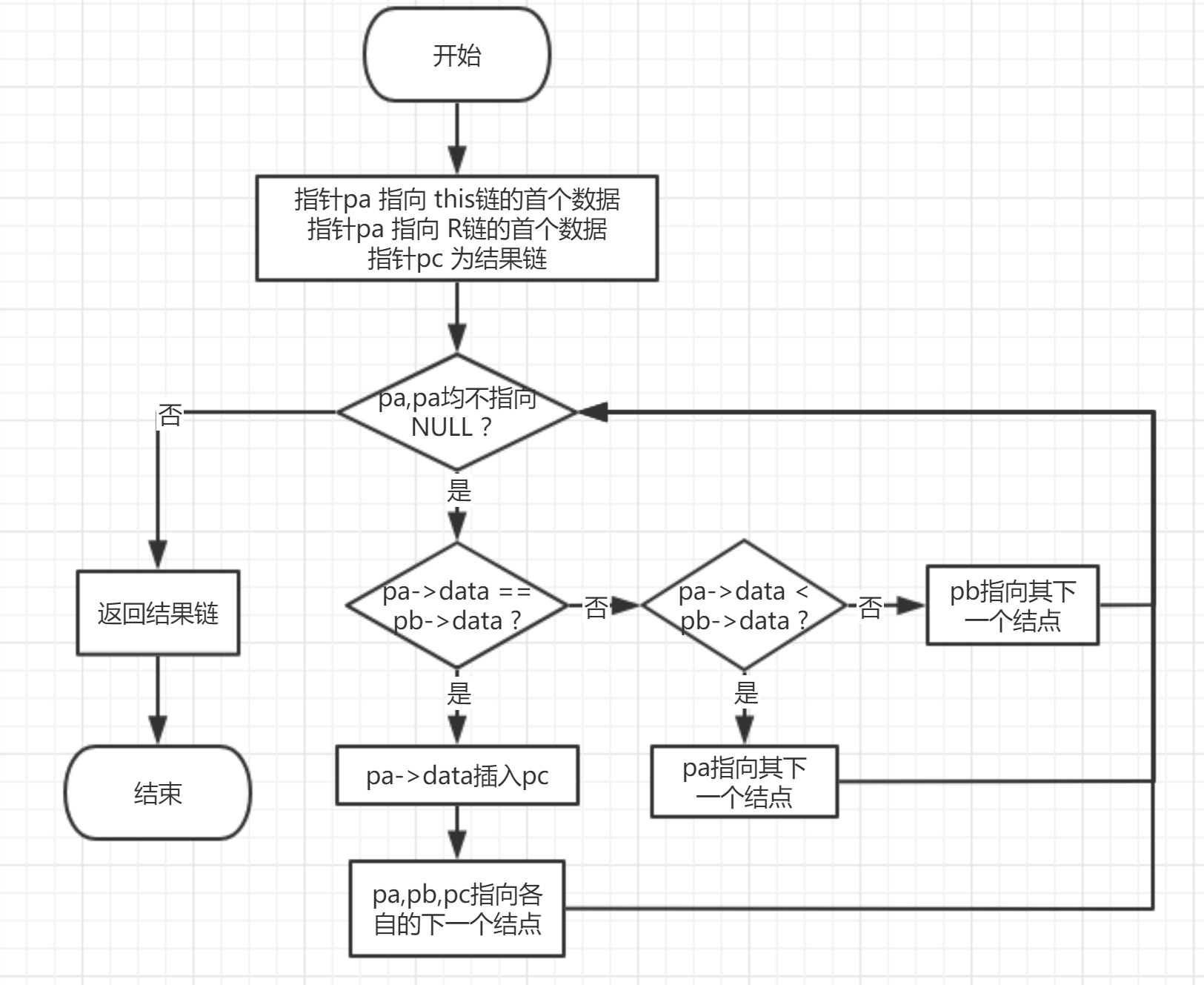


3.1.2 核心代码实现

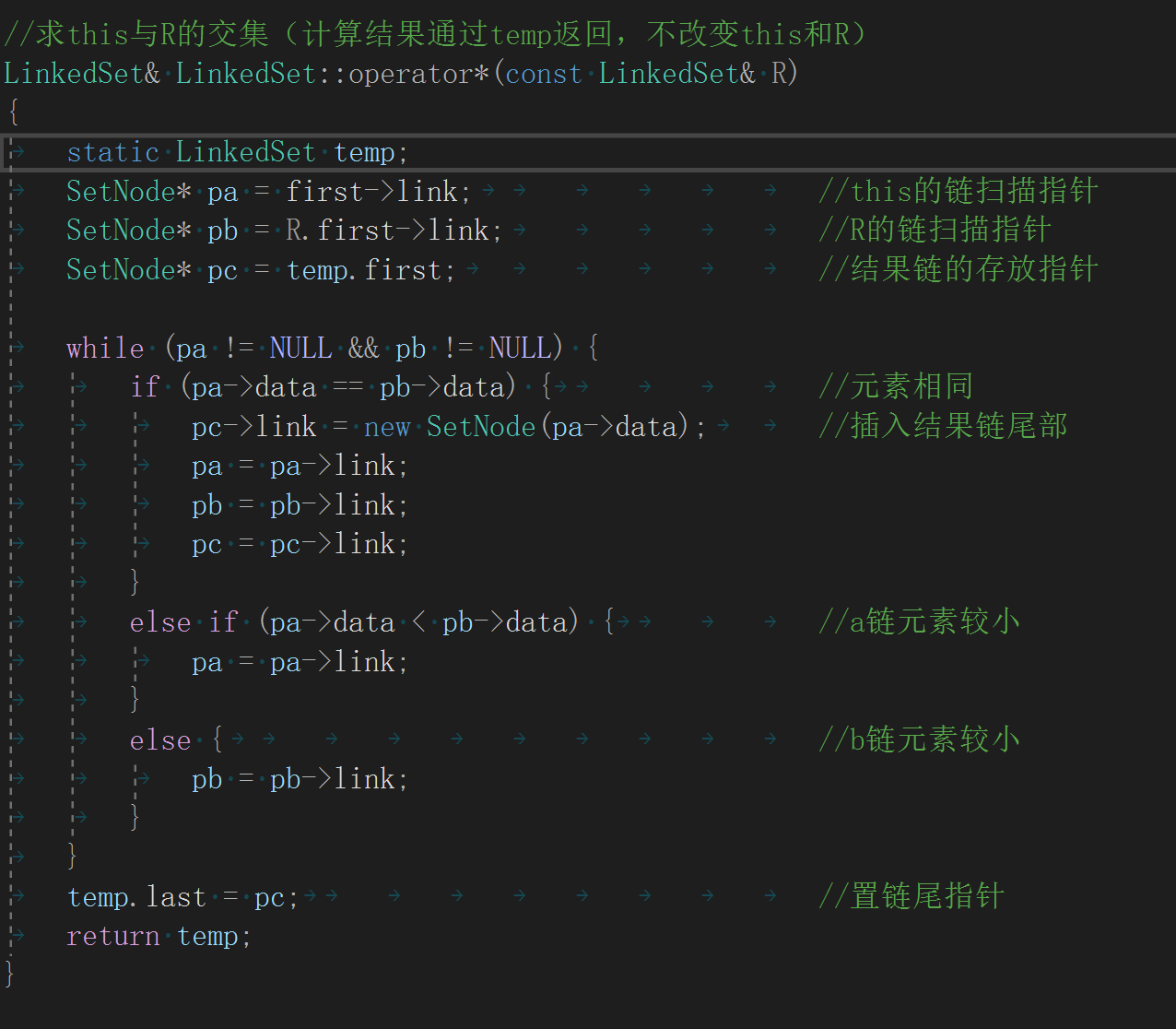


## 3.2 求交集功能

3.2.1求交集功能的流程图

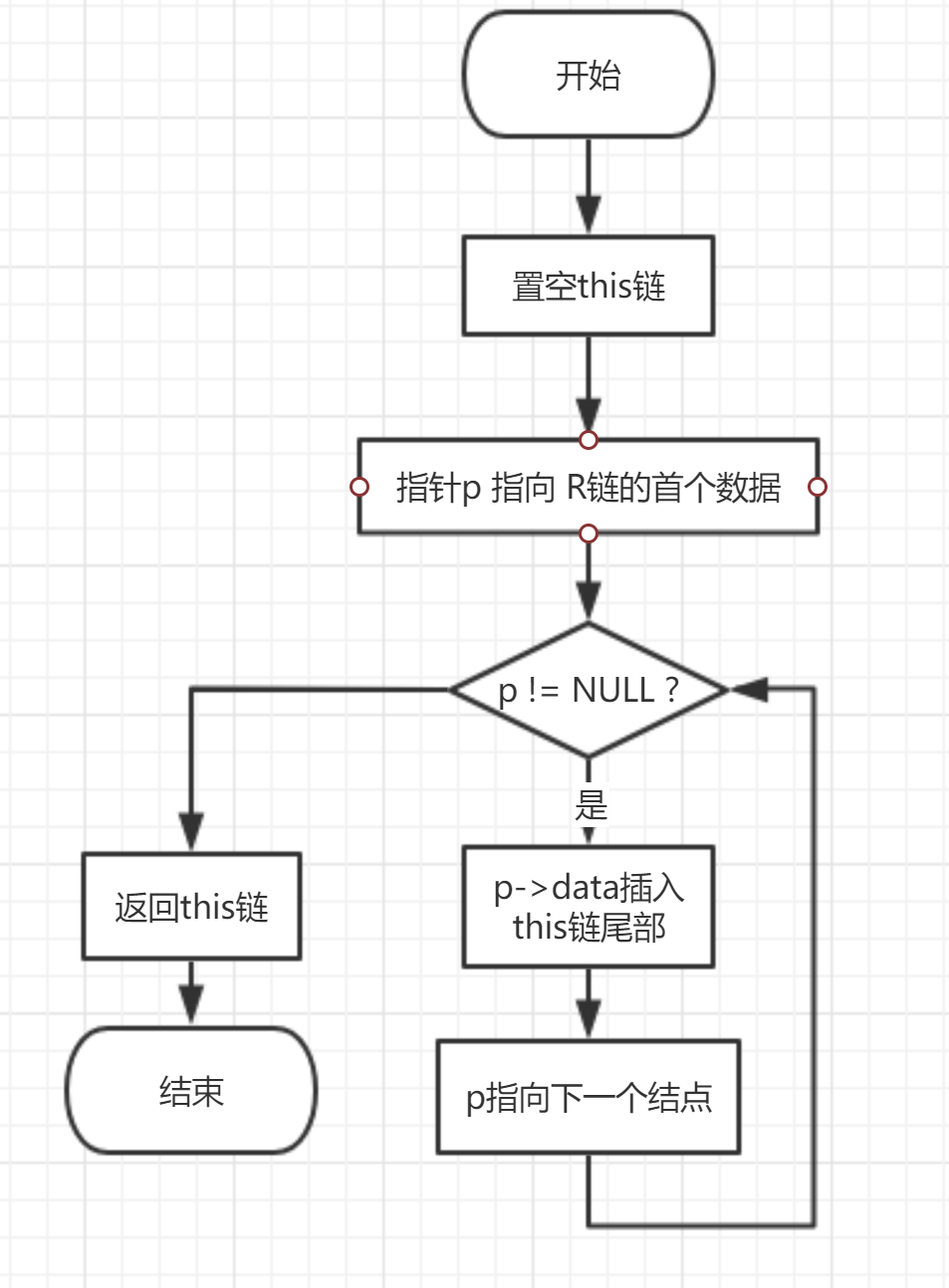


3.2.2 核心代码实现

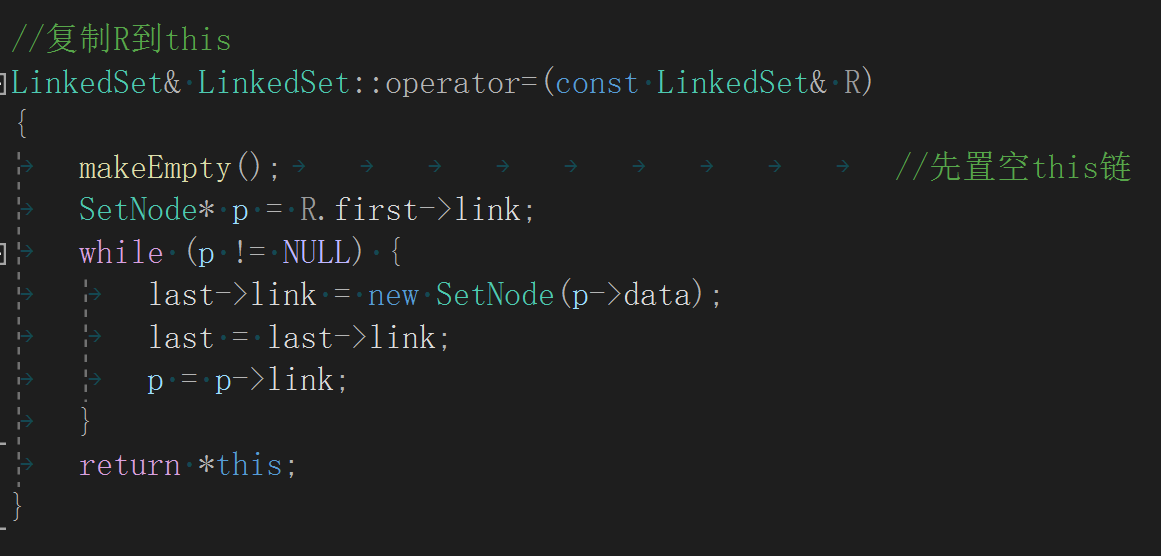


## 3.3 重载=运算符

3.3.1 重载=运算符流程图

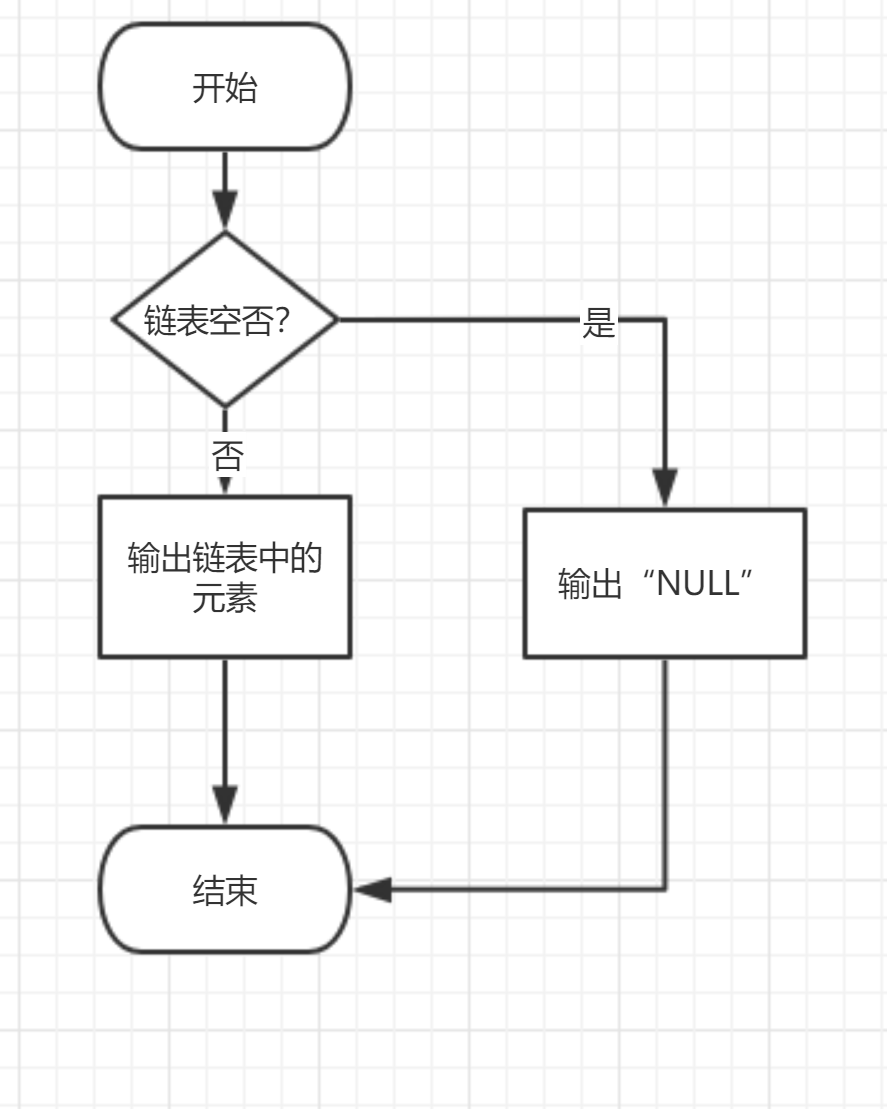


3.3.2 核心代码实现

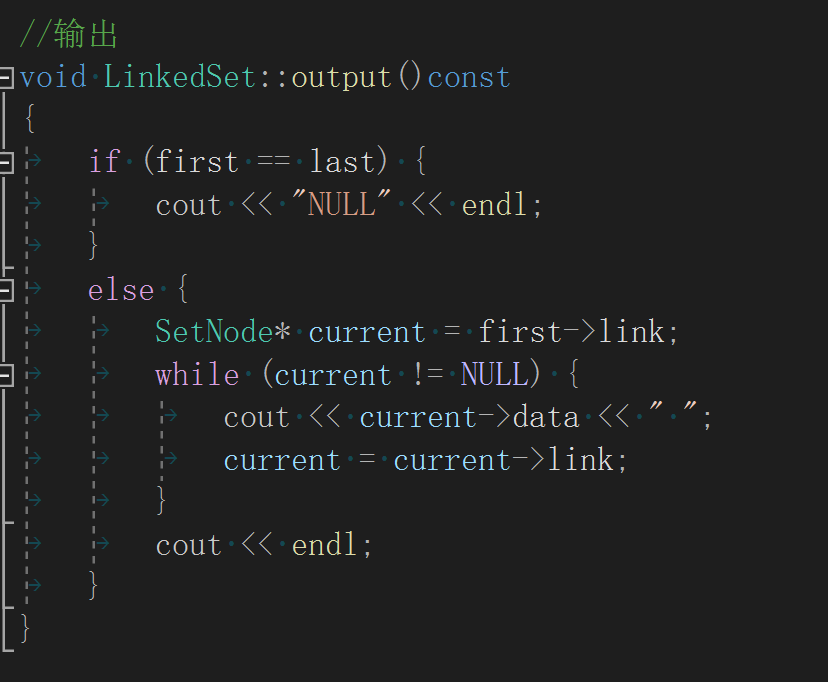


## 3.4 输出功能

3.4.1 输出功能流程图：



3.4.2 核心代码实现



# 

# 4 测试

## 4.1 功能测试

### 4.1.1 一般情况测试

**测试用例**：

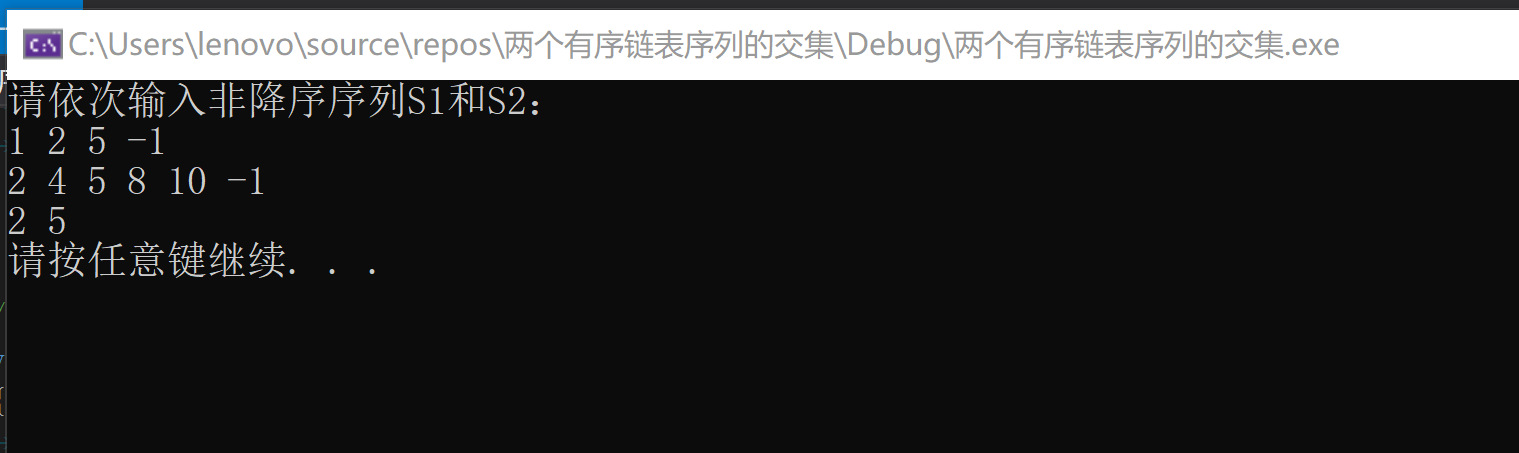
1 2 5 -1

2 4 5 8 10 -1

**预期结果**：

2 5

**实验结果**



### 

### 4.1.2 交集为空的情况测试

**测试用例：**

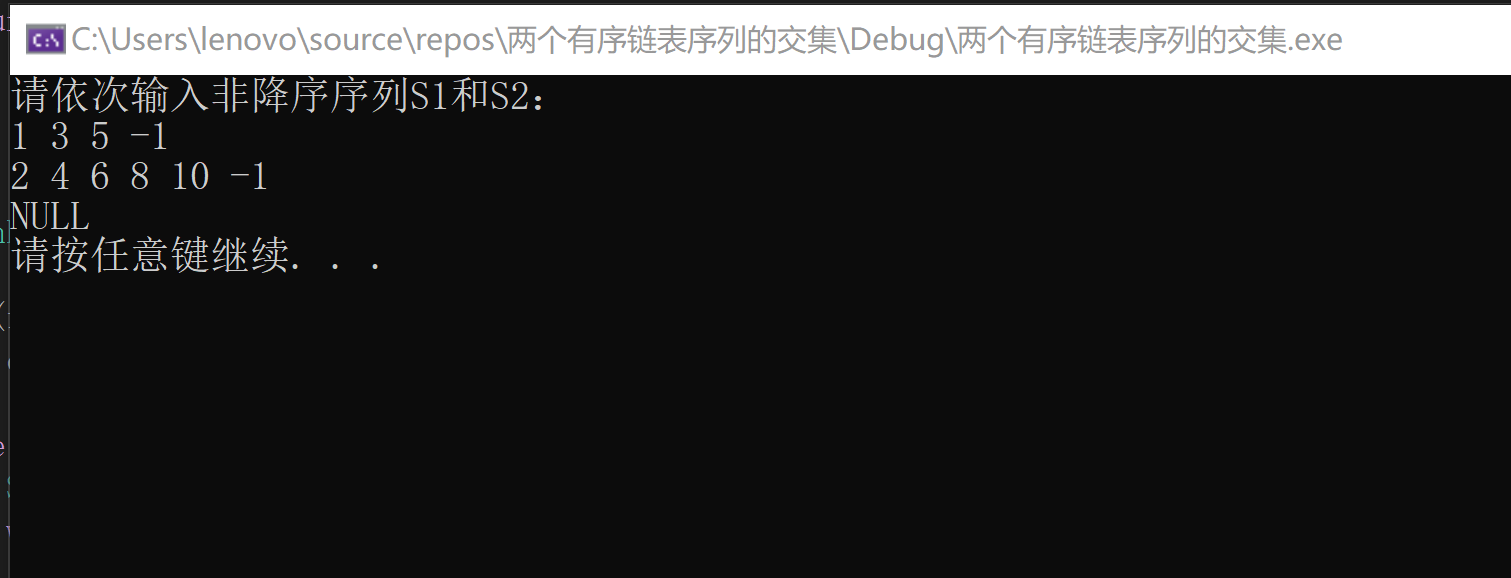
1 3 5 -1

2 4 6 8 10 -1

**预期结果：**

NULL

**实验结果：**



### 4.1.3 完全相交的情况测试

**测试用例：**

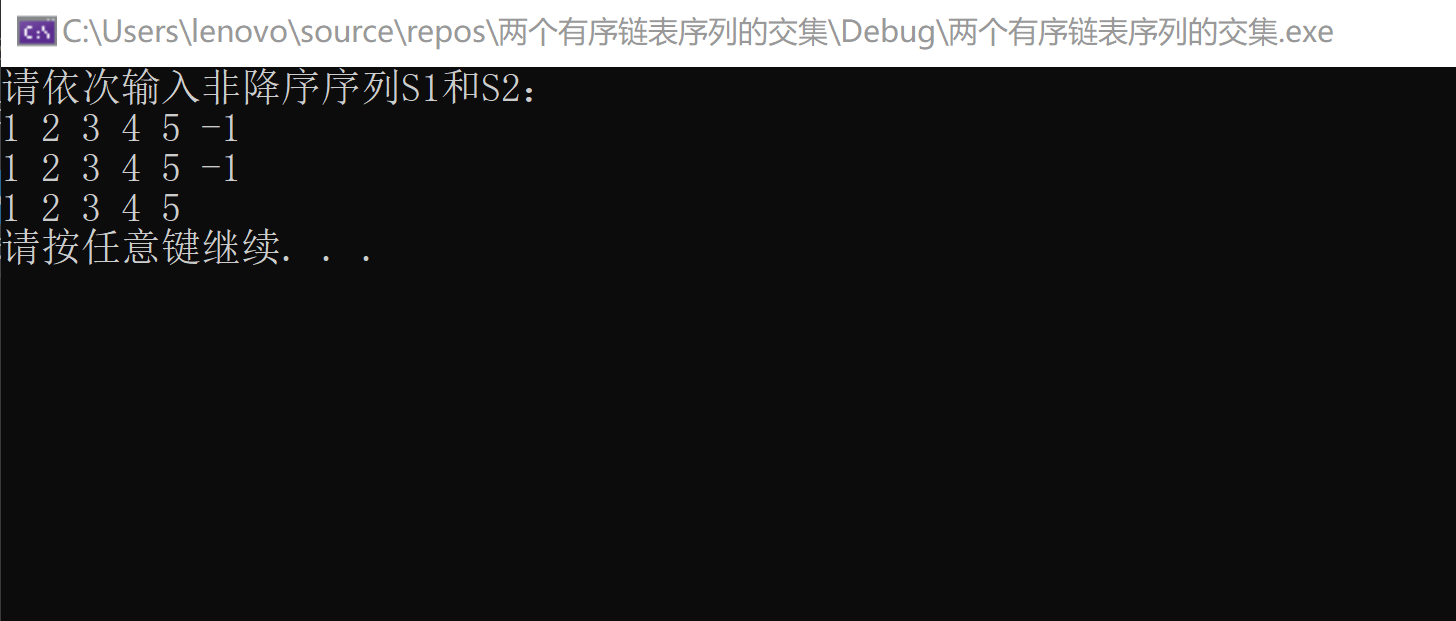
1 2 3 4 5 -1

1 2 3 4 5 -1

**预期结果：**

1 2 3 4 5

**实验结果：**



### 4.1.4 其中一个序列完全属于交集的情况测试

**测试用例：**

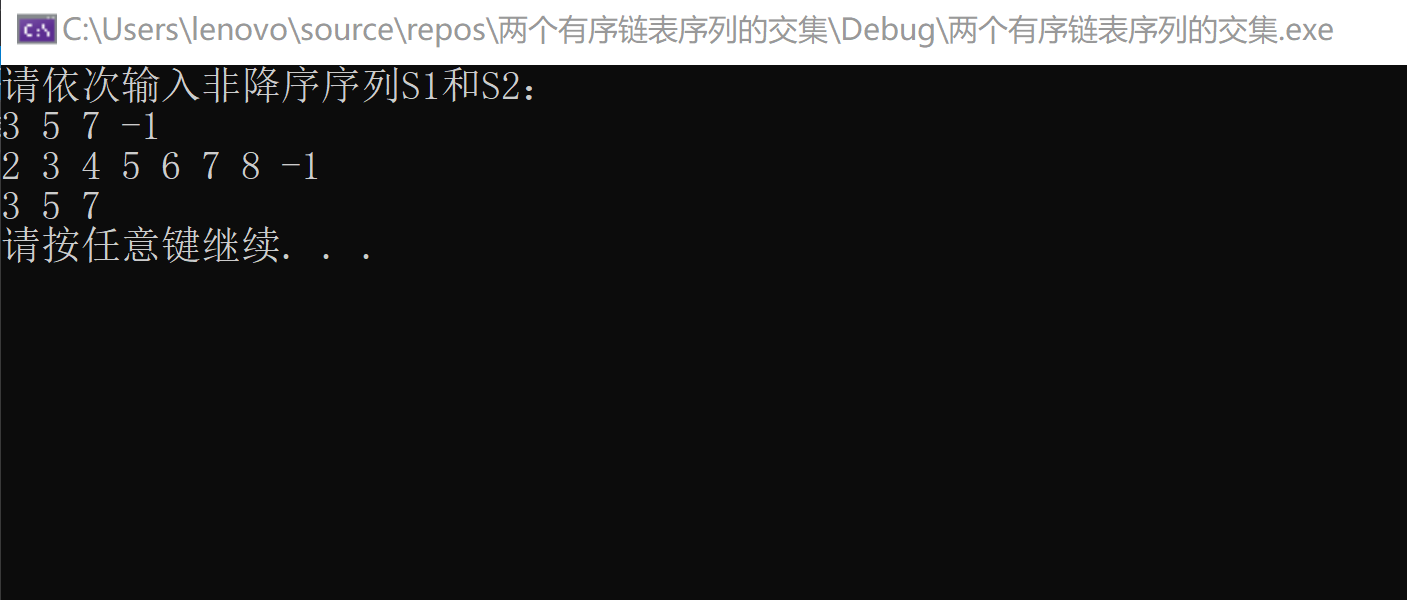
3 5 7 -1

2 3 4 5 6 7 8 -1

**预期结果：**

3 5 7

**实验结果：**



### 4.1.5 其中一个序列为空的情况测试

**测试用例：**

-1

10 100 1000 -1

**预期结果：**

NULL

**实验结果：**

