
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： 安江涛

学 号： 1952560

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.....	1
1.1 项目简介	1
2 设计.....	1
2.1 数据结构设计.....	1
2.2 类结构设计.....	1
2.3 成员与操作设计.....	1
3 实现.....	4
3.1 总体系统的实现.....	4
3.1.1 总体系统流程图	4
3.1.2 总体系统核心代码	5
3.1.3 总体系统截屏示例	5

1 分析

1.1 项目简介

已知两个非降序链表序列 S1 和 S2，设计函数构造出 S1 和 S2 的交集新链表 S3。

1. 输入说明：输入分 2 行，分别在每行给出由若干个正整数构成的非降序序列，用-1 表示序列的结尾（-1 不属于这个序列）。数字用空格间隔。
2. 输出说明：在一行中输出两个输入序列的交集序列，数字间用空格分开，结尾不能有多余空格；若新链表为空，输出 NULL。

2 设计

2.1 数据结构设计

如上所述，本题应该用链表实现。

2.2 类结构设计

链表包括两个抽象数据类型（ADT）——链表结点类（ListNode）与链表类（List），本题采用 class 描述链表结点类（ListNode）与链表类（List）。该链表封装成了一个头文件。可以直接调用。

2.3 成员与操作设计

链表结点类（ListNode）

```
1. template<class T> class List;
2.
3. template<class T>
4. class ListNode
5. {
6. public:
7.     friend List<T>;
8.     ListNode():_data(NULL),_next(nullptr){} //构造函数
9.     ListNode(const T& item,ListNode<T>* n=nullptr)
10.         :_data(item),_next(n){}
11.     ~ListNode() {} //析构函数
12.     _next = nullptr;
13. }
14. ListNode<T>* next()const { return this->_next; } //返回后置指针
15. T data()const { return this->_data; } //返回数据
16. void setData(T& d) { _data = d; } //设置数据
17. private:
18.     T _data; //存数据
19.     ListNode* _next; //后置指针
20. };
```

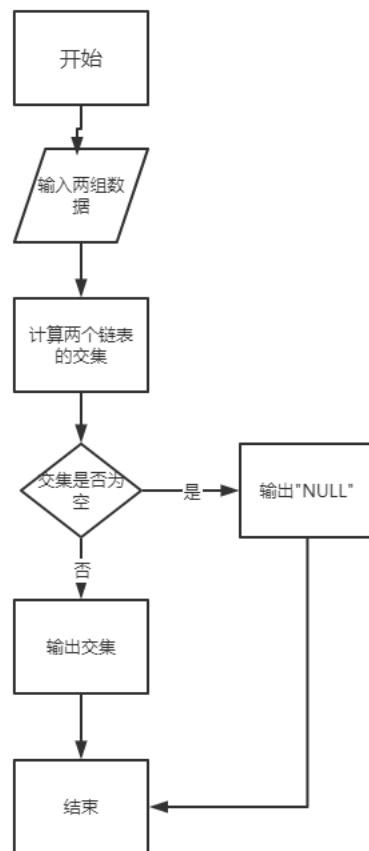
链表类 (List)

```
1. template<class T>
2. class List
3. {
4. public:
5.     List() :_len(0),_head(new ListNode<T>()),_last(nullptr) { //构造函数
6.         _head->_next= _head;
7.         _last = _head;
8.     }
9.     ~List() { //析构函数
10.         this->clear();
11.         delete _head;
12.     }
13. public:
14.     ListNode<T>* find(const T& item)const;
15.     void clear();//清除
16.     void push(const T& item);//后插
17.     bool insert(const T& item, int ind = 0);//插入
18.     bool erase(const T& item);//删除
19.     ListNode<T>* begin()const { return this->_head->_next; }
20.     ListNode<T>* end()const { return this->_head; }
21.     bool change(const T& newitem, ListNode<T>* itor);//修改
22.     int size()const { return this->_len; }
23.     bool empty()const { return !(this->_len); }
24.     void print()const;
25. private:
26.     int _len;
27.     ListNode<T>* _head;
28.     //head is a pointer to the first list node
29.     ListNode<T>* _last;
30.     //last is a pointer to the last node
31. };
```

3 实现

3.1 总体系统的实现

3.1.1 总体系统流程图



3.1.2 总体系统核心代码

```
1. #include "../H/cl_List.h"
2. int main()
3. {
4.     List<int> list1, list2, list3;
5.     int item = 0;
6.     while (true) {
7.         std::cin >> item;
8.         if (item == -1)
9.             break;
10.        list1.push(item);
11.    }
12.    while (true) {
13.        std::cin >> item;
14.        if (item == -1)
15.            break;
16.        list2.push(item);
17.    }
18.    ListNode<int>* itor1 = list1.begin();
19.    ListNode<int>* itor2 = list2.begin();
20.    while (itor1 != list1.end() && itor2 != list2.end()) {
21.        if (itor1->data() == itor2->data()) {
22.            list3.push(itor1->data());
23.            itor1 = itor1->next(); itor2 = itor2->next();
24.        }
25.        else if (itor1->data() < itor2->data()) {
26.            itor1 = itor1->next();
27.        }
28.        else if (itor1->data() > itor2->data()) {
29.            itor2 = itor2->next();
30.        }
31.    }
32.    list3.print();
33. }
```

3.1.3 总体系统截屏示例

```
Microsoft Visual Studio 调试控制台
1 3 5 -1
2 4 6 8 10 -1
NULL
D:\c++作业\做题\Debug\做题.exe (进程 12352) 已退出，代码为 0。
按任意键关闭此窗口. . .
```