项目说明文档

数据结构课程设计

**——**二叉排序树问题

作 者 姓 名： 罗吉皓

学 号： 1652792

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 4

1.1 项目名称 ：二叉排序树 4

1.2 项目背景 4

1.3 项目功能分析 4

II. 设计 5

2.1 数据结构设计 5

2.2 数据结构类的设计 5

2.3 系统设计 5

​2.3.1 二叉排序树节点类的设计 5

2.3.2 二叉排序树类的设计 6

2.4 系统设计 7

III 实现 8

3.1 建立二叉排序树的实现 8

3.1.0 建立二叉排序树的系统设计 8

3.1.1 建立二叉排序树流程图 9

3.1.2 思路分析 11

3.1.3 具体实现如下： 11

3.1.4 建立二叉排序树截屏示例 12

3.2 二叉排序树插入元素实现 13

3.2.0 二叉排序树插入元素的系统设计 13

3.2.1 二叉排序树插入元素流程图 14

3.2.2 思路分析 15

3.2.3 具体实现如下： 15

3.2.4 二叉排序树插入元素功能截屏示例 16

3.3 二叉排序树查询元素实现 16

3.3.0 二叉排序树查询元素的系统设计 16

3.3.1 二叉排序树查询元素流程图 17

3.3.2 思路分析 17

3.3.3 具体实现如下： 18

3.3.4 二叉排序树搜索元素功能截屏示例 18

3.4 总体系统截屏示例 18

IV 测试 19

4.1搜索功能测试 19

4.1.1 建立二叉排序树测试 19

4.1.2 插入功能测试 20

4.1.3 查询功能测试 20

4.2.健壮性实验 21

4.2.1 输入健壮性判断 21

4.2.2 插入/建立系统时元素重复 21

V 总结 22

VI 参考文献 22

1. 分析

1.1 项目名称 ：二叉排序树

1.2 项目背景

依次输入关键字并建立二叉排序树，实现二叉排序数的插入和查找功能。

二叉排序树就是指将原来已有的数据根据大小构成一棵二叉树，二叉树中的所有结点数据满足一定的大小关系，所有的左子树中的结点均比根结点小，所有的右子树的结点均比根结点大。

二叉排序树查找是指按照二叉排序树中结点的关系进行查找，查找关键自首先同根结点进行比较，如果相等则查找成功；如果比根节点小，则在左子树中查找；如果比根结点大，则在右子树中进行查找。这种查找方法可以快速缩小查找范围，大大减少查找关键的比较次数，从而提高查找的效率。

1.3 项目功能分析

​ 作为一个二叉树搜索树实现系统，首先应该有的功能就是输入关键字并建立二叉搜索树，并且可以予以显示。其次，二叉搜索树实现系统还应该具有插入元素、查询元素等功能。最后，考试报名系统软件还应该确保软件可以正常关闭。

综上所述，一个二叉搜索树模拟系统至少应该具有输入、输出、插入、查询、退出的功能。

II. 设计

2.1 数据结构设计

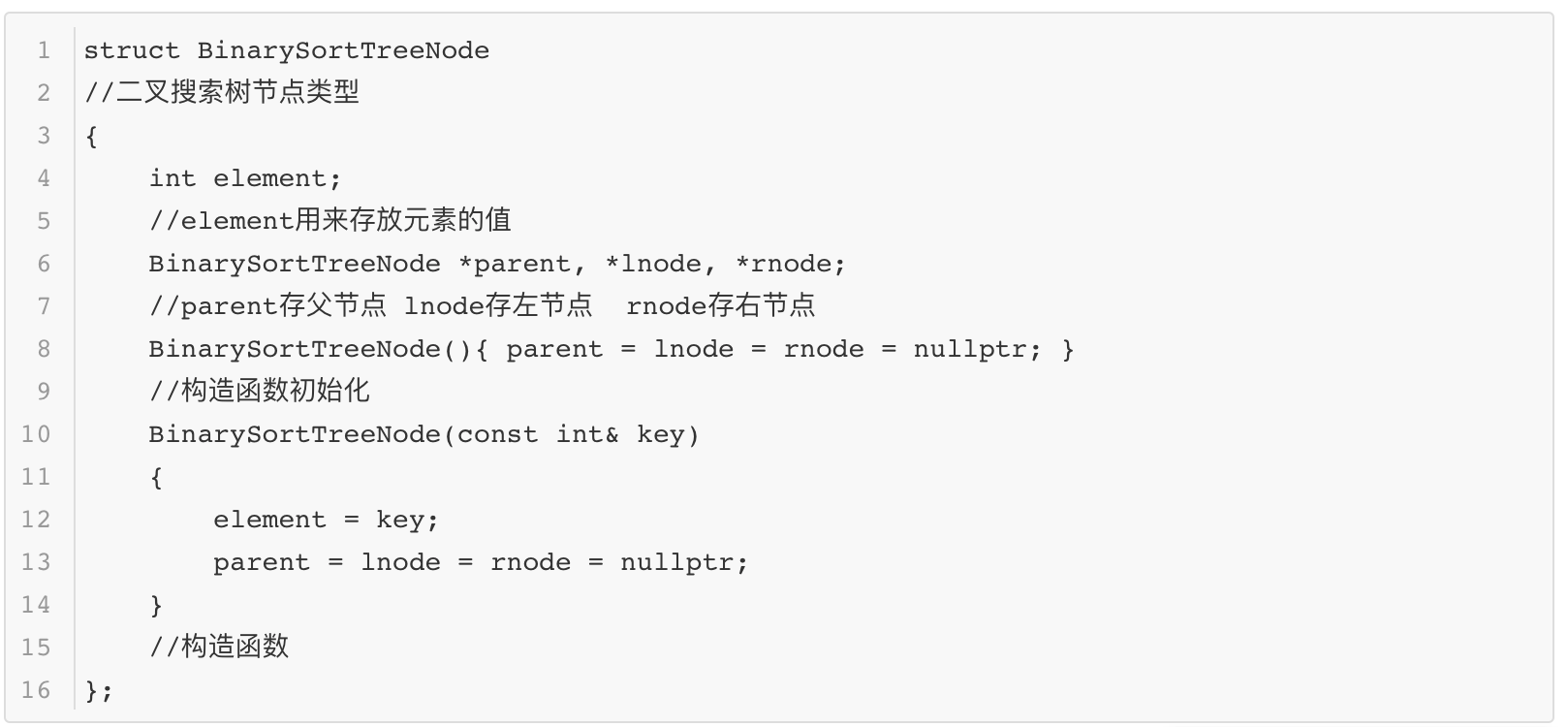
正如以上功能分析所述，该系统要求大量的添加查询操作，而链表进行增加、查询等操作十分简便，因此我们考虑使用链表数据结构来实现二叉搜索树的构建。在链表类的实现中，我们要考虑插入元素，查询元素等操作之外，还要求能够实现二叉排序树的基本要求，这在接下来的文档中会有所体现。

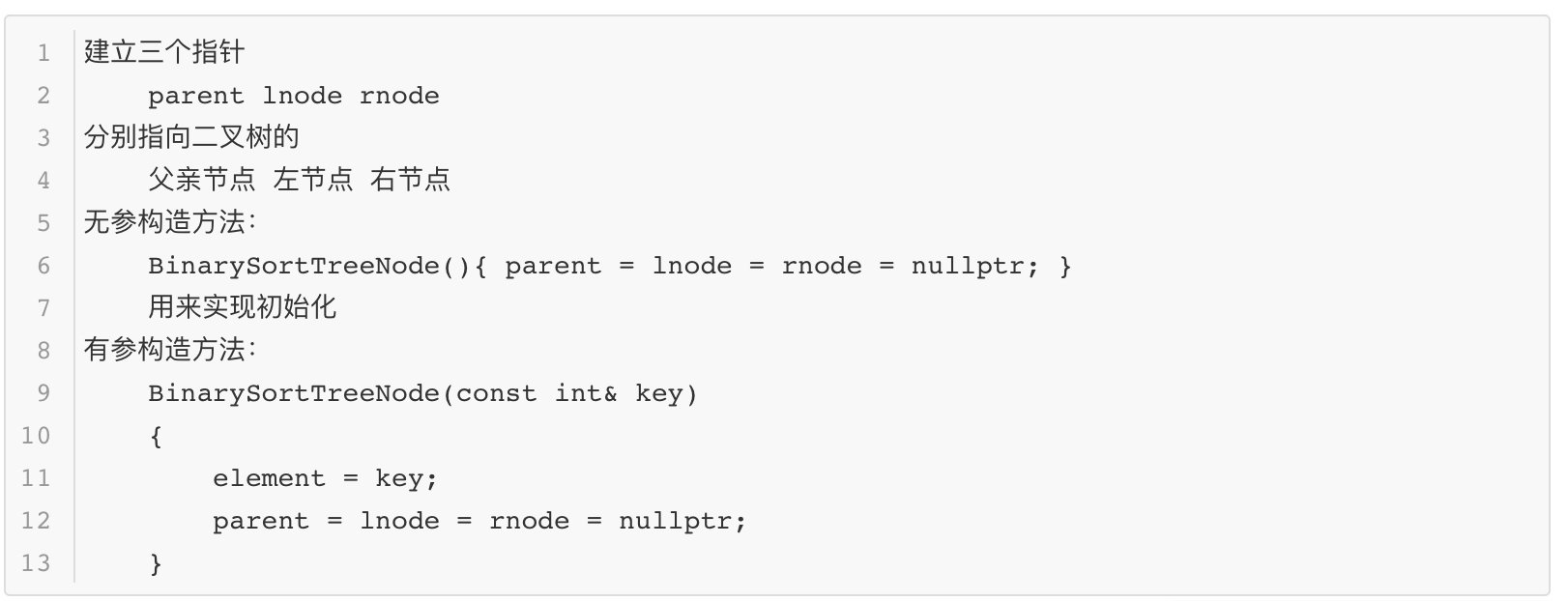
2.2 数据结构类的设计

经典的链表一般包括两个抽象数据类型（ADT）——链表结点类（BinarySortTreeNode）与链表类（BinarySortTree），而两个类之间的耦合关系可以采用嵌套、继承等多种关系。为方便处理，本系统采用struct描述链表结点类（BinarySortTreeNode）。

2.3 系统设计

​2.3.1 二叉排序树节点类的设计





2.3.2 二叉排序树类的设计



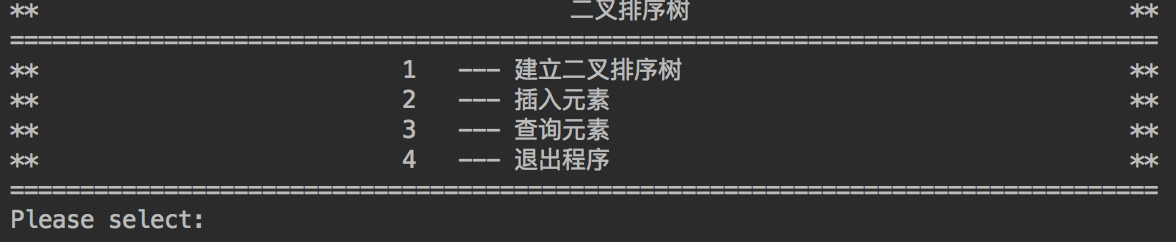


2.4 系统设计

系统首先调用Init()函数实现对屏幕的初始化，根据用户所输入的操作码（choice）执行二叉排序树对应的构造，插入，查询，输出等操作，具体实现如下：



输入功能截屏示例：



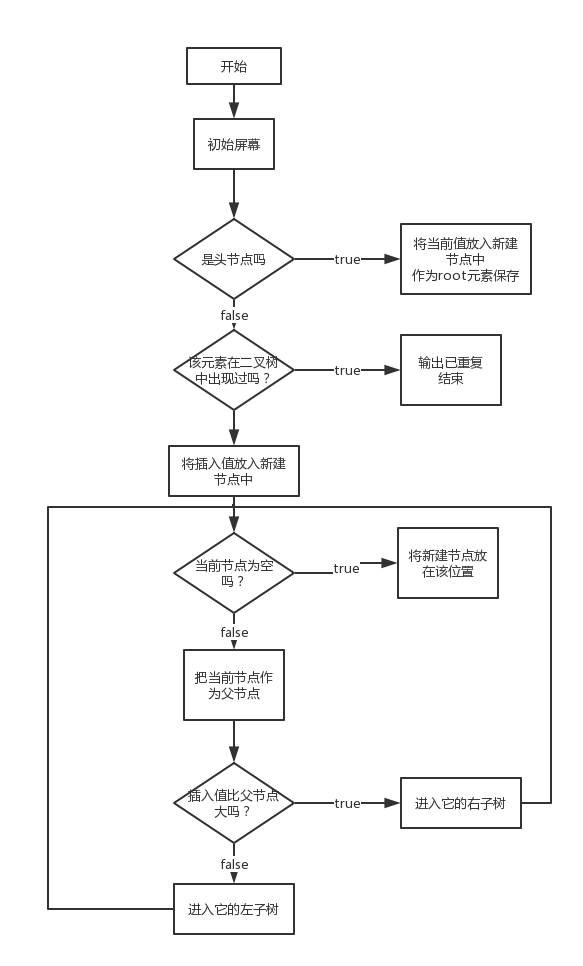
III 实现

3.1 建立二叉排序树的实现

3.1.0 建立二叉排序树的系统设计



3.1.1 建立二叉排序树流程图



3.1.2 思路分析

该链表存储主要有两个指针：lnode rnode ，当新插入元素值比当前节点大时，就去它的右子树查找，反之去它的左子树寻找，直到寻找到第一个空指针的位置，将该元素插入。

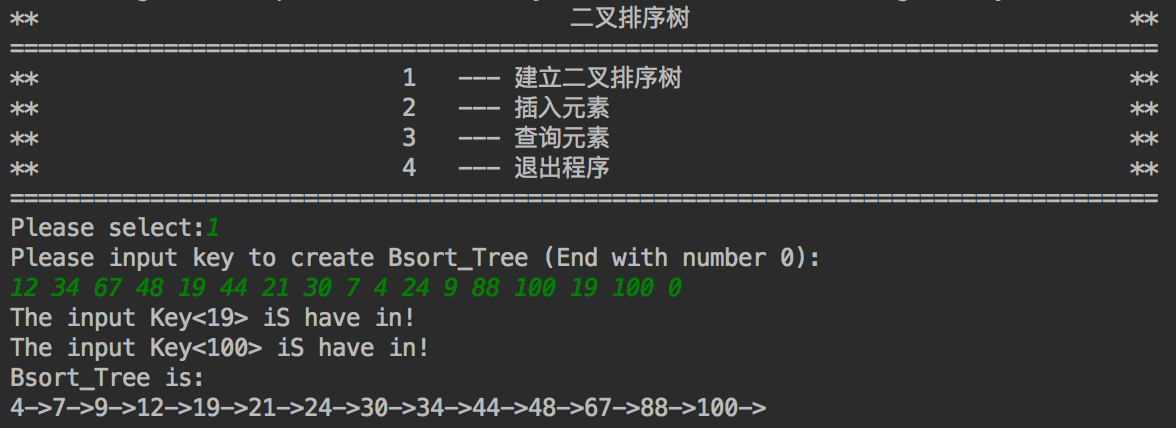
3.1.3 具体实现如下：



其中判断重复的TreeCheck函数实现如下：



3.1.4 建立二叉排序树截屏示例

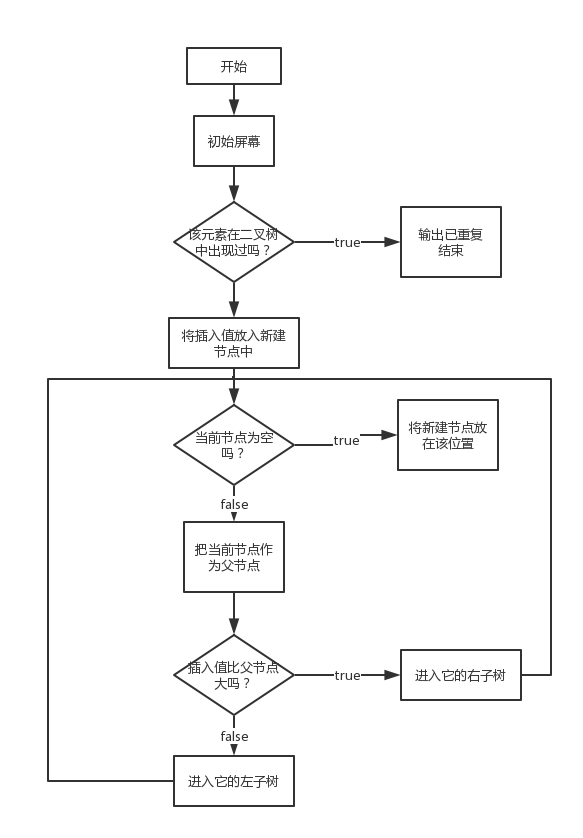


3.2 二叉排序树插入元素实现

3.2.0 二叉排序树插入元素的系统设计



3.2.1 二叉排序树插入元素流程图



3.2.2 思路分析

插入元素的功能实现和建立系统的实现较为类似。

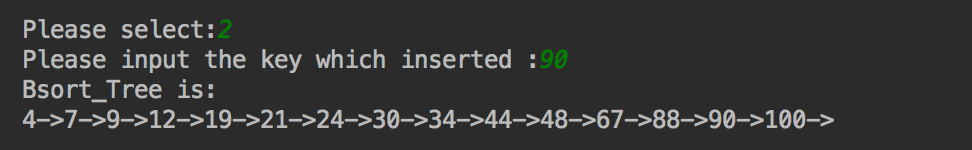
3.2.3 具体实现如下：



其中判断重复的TreeCheck函数实现如下：



3.2.4 二叉排序树插入元素功能截屏示例



3.3 二叉排序树查询元素实现

3.3.0 二叉排序树查询元素的系统设计



3.3.1 二叉排序树查询元素流程图

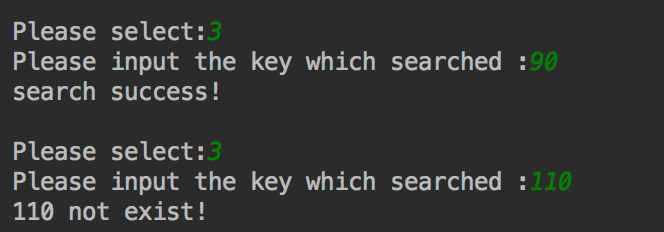
3.3.2 思路分析

当新插入元素值比当前节点大时，就去它的右子树查找，反之去它的左子树寻找，如果找到空指针的位置，那么该查询的元素在整棵树中不存在。

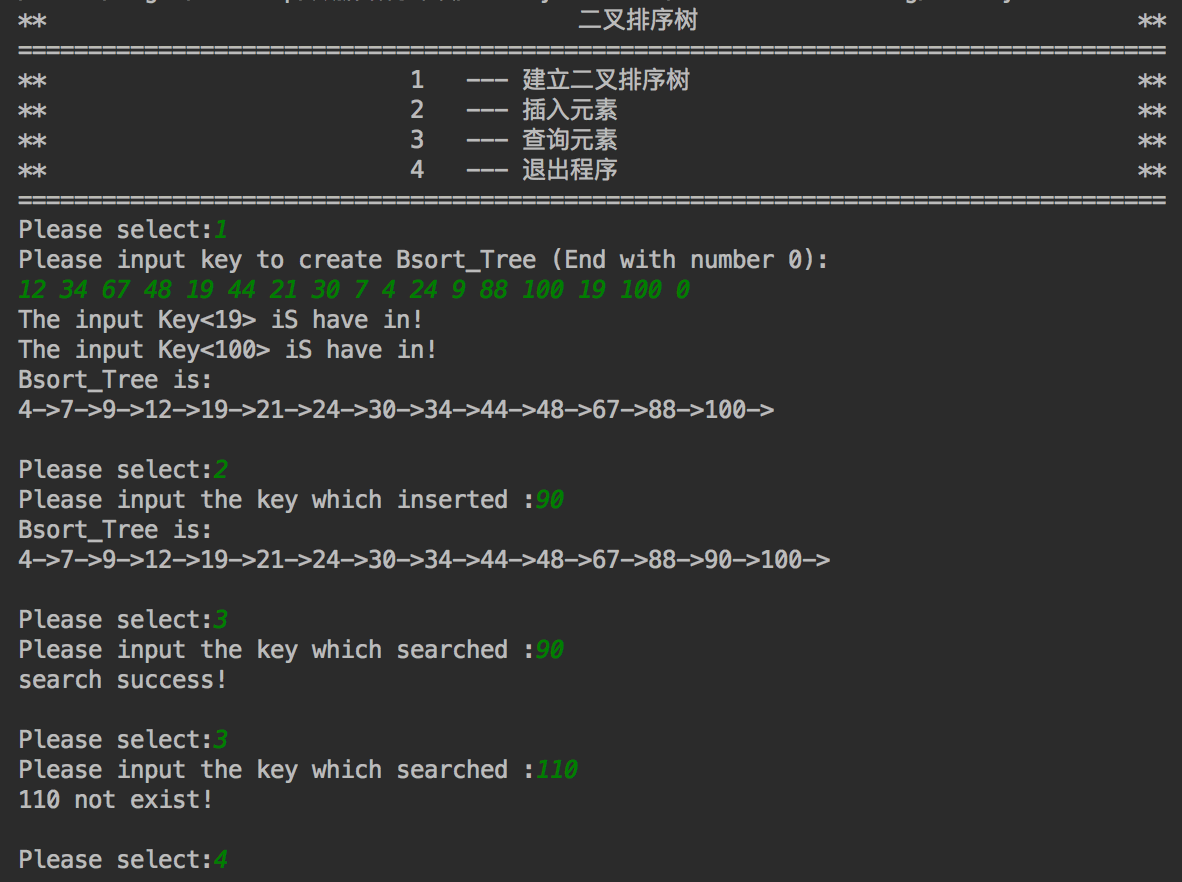
3.3.3 具体实现如下：



3.3.4 二叉排序树搜索元素功能截屏示例



3.4 总体系统截屏示例



IV 测试

4.1搜索功能测试

4.1.1 建立二叉排序树测试

测试用例：

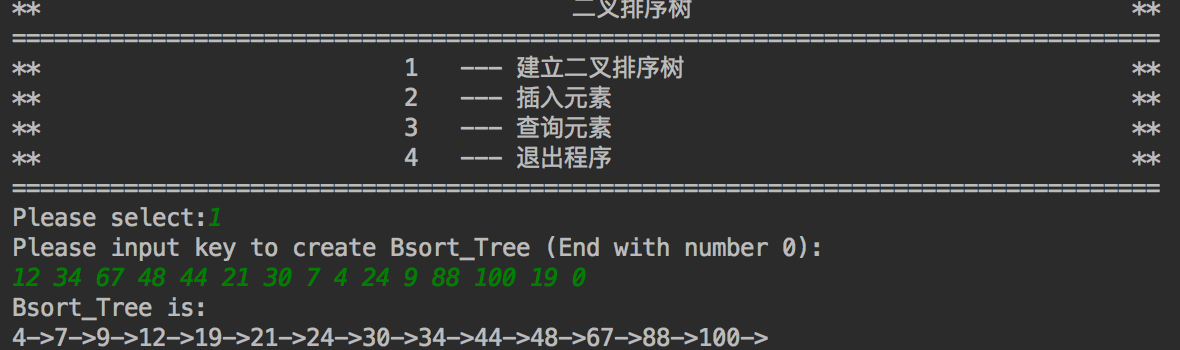
12 34 67 48 44 21 30 7 4 24 9 88 100 19 0

预期结果：

Bsort\_Tree is:

4->7->9->12->19->21->24->30->34->44->48->67->88->100->

实际结果



4.1.2 插入功能测试

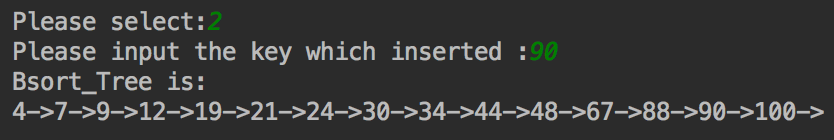
测试用例：

90

预期结果：

4->7->9->12->19->21->24->30->34->44->48->67->88->90->100->

实际结果：



4.1.3 查询功能测试

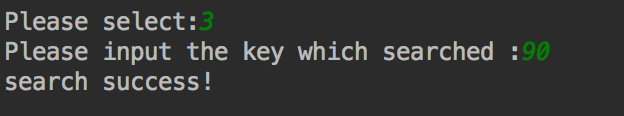
测试用例：

90

预期结果

search success!

实际结果



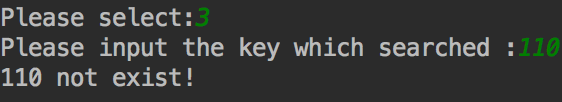
测试用例：

110

预期结果

110 not exist!

实际结果



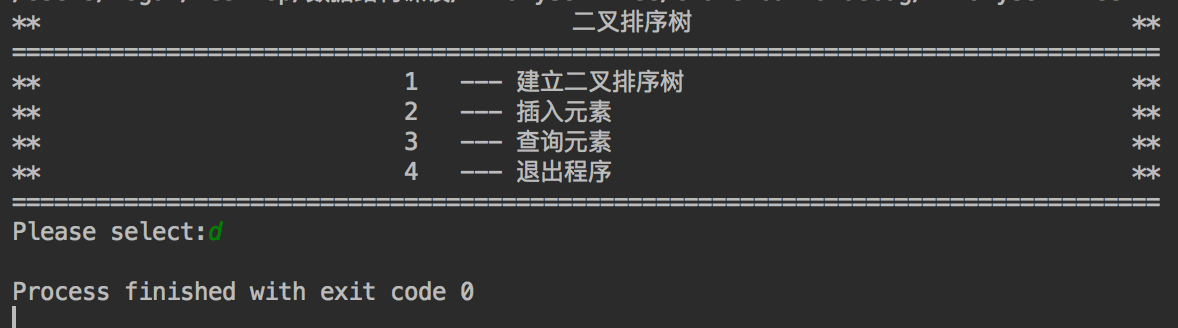
4.2.健壮性实验

4.2.1 输入健壮性判断

测试用例：随意输入**choice**

预期结果：程序直接退出

实际结果



4.2.2 插入/建立系统时元素重复

测试用例：

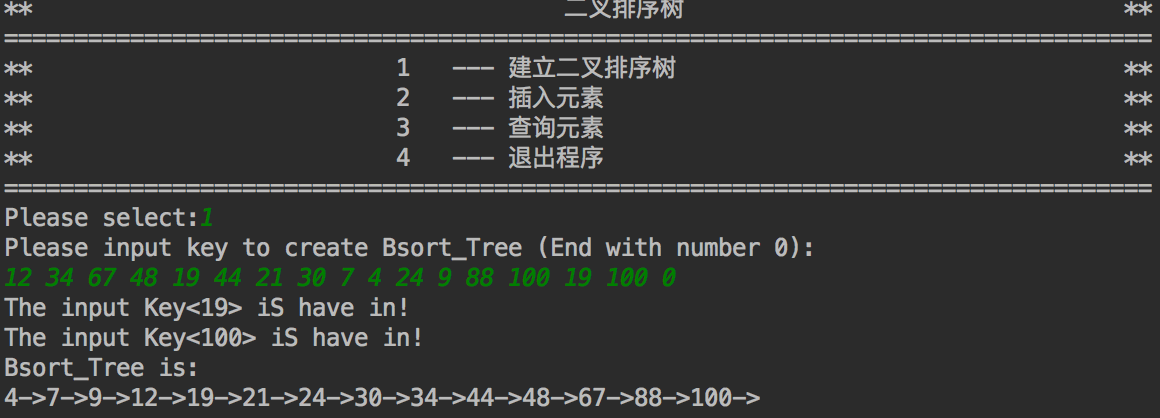
12 34 67 48 19 44 21 30 7 4 24 9 88 100 19 100 0

预期结果：

The input Key<19> iS have in!

The input Key<100> iS have in!

实际结果



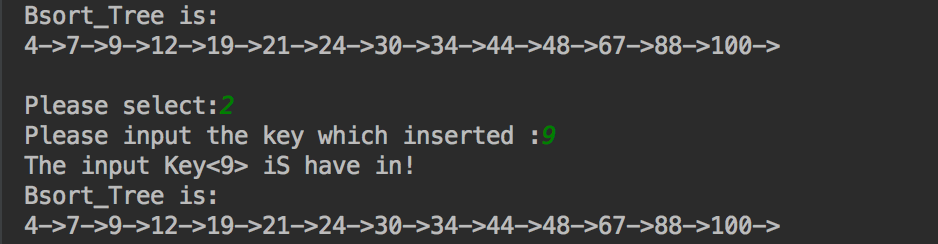
测试用例：

9

预期结果：

The input Key<9> iS have in!

实际结果



V 总结

通过对于二叉排序树类的设计，我近一步理解了二叉排序树的原理。

VI 参考文献

数据结构课本