项目说明文档

数据结构课程设计

——二叉排序树

作 者 姓 名： 陈君

学 号： 2250420

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji Universit

# 1 分析

## 1.1 背景分析

## 二叉排序树（Binary Search Tree，BST）是一种常用的数据结构，它具有快速的查找、插入和删除操作。在计算机科学领域，BST被广泛应用于数据库系统、编译器设计、图形学等多个领域。设计和实现一个二叉排序树，提供基本的树操作，并为用户提供一个可视化的界面，能够直观地理解和操作二叉排序树。

## 1.2 功能分析

# 本项目旨在设计并实现一个可靠、高效的二叉排序树。通过使用适当的数据结构和算法，确保树的平衡性，避免出现极端情况下的性能退化。为了使用户更好地理解和使用二叉排序树，本项目将提供一个直观的可视化界面。用户可以通过界面进行插入、删除和查找操作，同时能够观察树的结构变化。优化算法和数据结构，以提高树的查找、插入和删除性能。在大规模数据集下，确保系统依然能够快速响应用户操作。

# 2 设计

## 2.1 数据结构设计

如上功能分析所述，这是一个树形结构，所以需要设计两个类，分别是结点类和树类，整个二叉排序树由一个根节点开始构建。同时结点类是树类的友元，这样可以方便树中的方法访问结点中的数据。

## 2.2 类结构设计

本系统采用设计树BST类来存储二叉平衡树，用nodelist类来存储结点的信息，同时将BST类定义为nodelist的友元，方便访问结点信息。

## 2.3 成员与操作设计

## 结点类（node）

class node

{

public:

node();

node(int n);

int get\_key();

private:

friend BST;

node\* leftchild;

node\* rightchild;

int key;

};

BST类

class BST

{

public:

BST() :root(NULL) {};//默认构造函数

void build\_BST();//构建一个BST

void insert(int num);//插入功能

node\* \_search(int num,node\* &p,node\*root);//查找功能的递归调用函数

node\* search(int num,node\* &p); //查找功能

void inorder\_traversal(void); //中序遍历递归调用函数

void \_inorder\_traversal(node\* root); //中序遍历递归调用

private:

node\* root;

};

## 2.4 系统设计

主程序先调用initialzeio函数初始化io界面然后再声明一个BST对象创建二叉搜索树，然后使用一个switch语句，跳转到建立二叉排序树、插入元素、查询元素、退出程序，四个操作。

# 3 实现

3.1void BST::build\_BST()//构建二叉搜索树

具体实现：通过不断插入元素实现二叉搜索树的构建。提示用户输入数值，判断该数值的合法性，如果输入错误，直接结束，进入下一次选择。输入正确直接调用插入函数，插入新的值。

3.2.3 构造函数功能截屏示例

## 3.2 void initializeio()//初始化io界面

具体实现：使用一连串的cout函数，打印题目所展示的界面。

## 3.2.1 void initializeio()功能截屏示例



## 3.3 void insert(int num);//插入功能

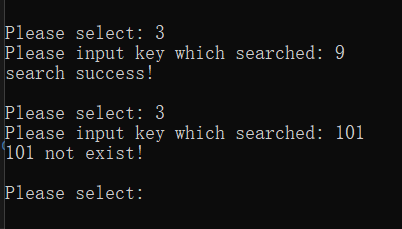
具体实现：插入的元素位置应该位于搜索失败的位置。所以插入函数应该先调用搜索函数，获取搜索的失败的位置，为此搜索函数额外增加一个引用参数用于记录搜索终止的位置的双亲结点。调用搜索函数，如果搜索函数返回值不为空，说明已经含有这个元素，此时返回搜索失败。如果为空，则将n和双亲结点的值比较，如果比双亲小就将这个结点插入在双亲的左孩子位置上，如果比双亲大，则插入在双亲的右孩子位置上。

### 3.3.2 void insert(int num);//插入功能截图示例

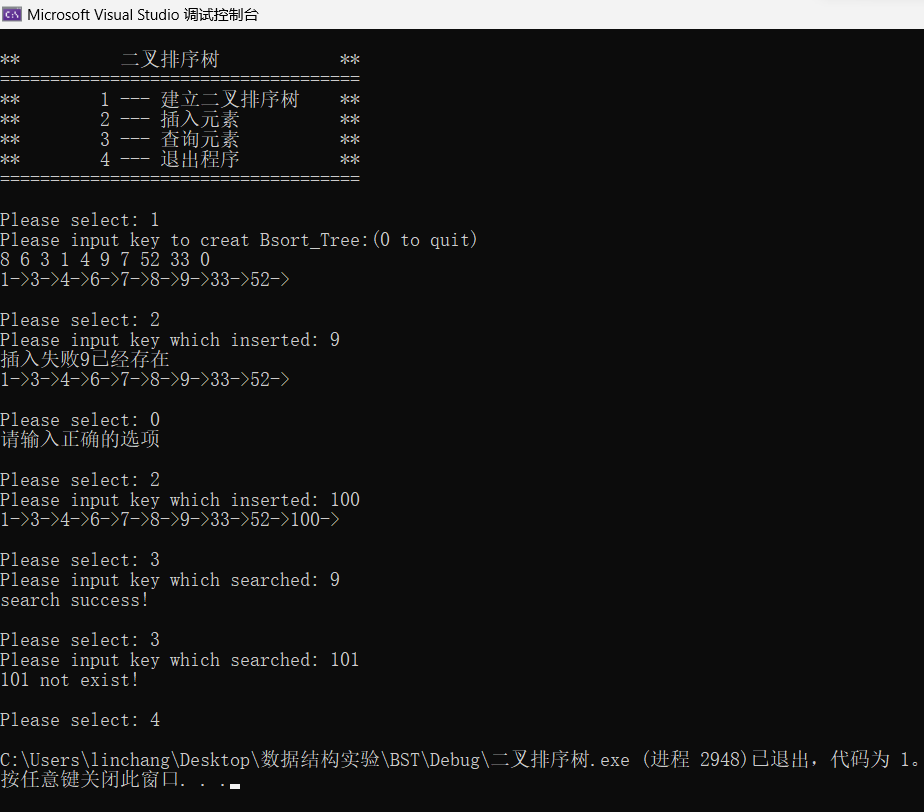
## 3.4 node\* BST::\_search(int n, node\*& p, node\* root) //搜索函数

具体实现：采用递归的形式，递归终止条件：1、当前的根节点和n相等则返回根节点。2、当前根结点为NULL返回NULL。如果当前根结点的值比n小则再次调用搜索函数在右子树中搜索，如果比n大则在左子树上搜索。最终如果搜索到则返回指针p，没找到则返回NULL。

## 3.5 搜索函数的截图功能展示



## 3.6 总体系统的实现

主程序先调用initialzeio函数初始化io界面然后再声明一个BST对象创建二叉搜索树，然后使用一个switch语句，跳转到建立二叉排序树、插入元素、查询元素、退出程序，四个操作。 3.6.1 总体系统截屏示例

# 4 测试

## 4.1 功能测试

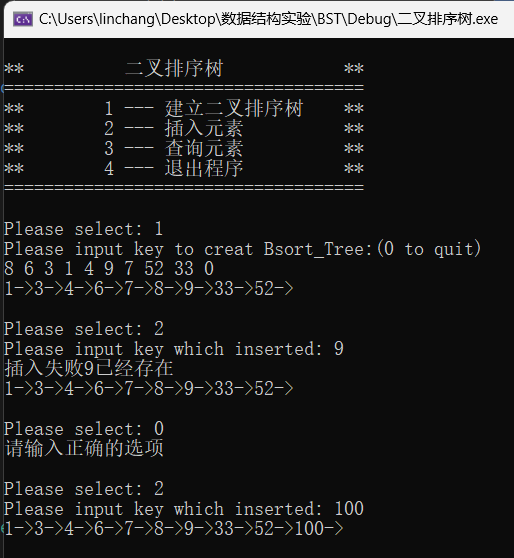
### 4.1.1 建立二叉排序树功能

测试用例：8 6 3 1 4 9 7 5 2 33 0

测试结果

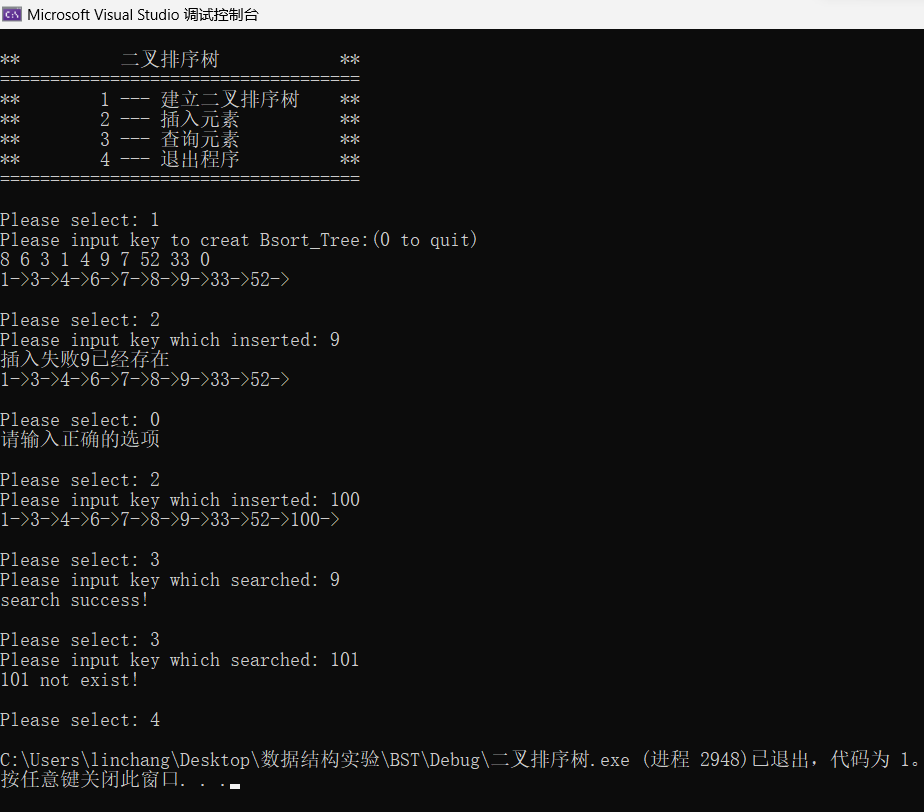
### 4.1.2 插入元素功能

测试用例：插入数字100

测试结果：

### 4.1.2 查询元素功能

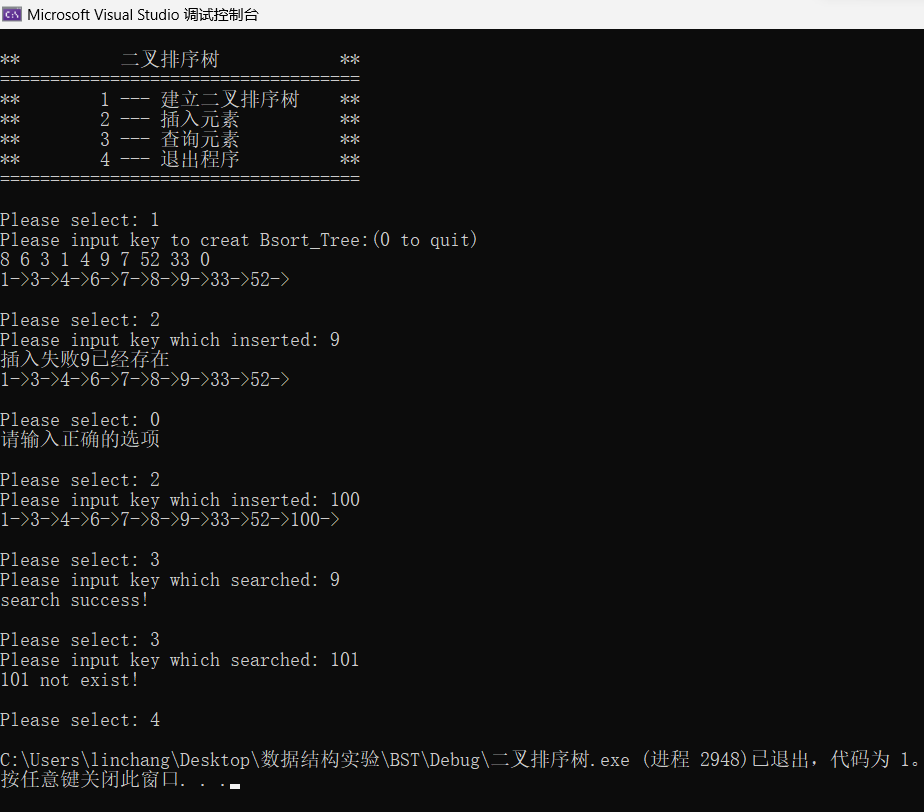
测试用例：查询元素6

测试结果：

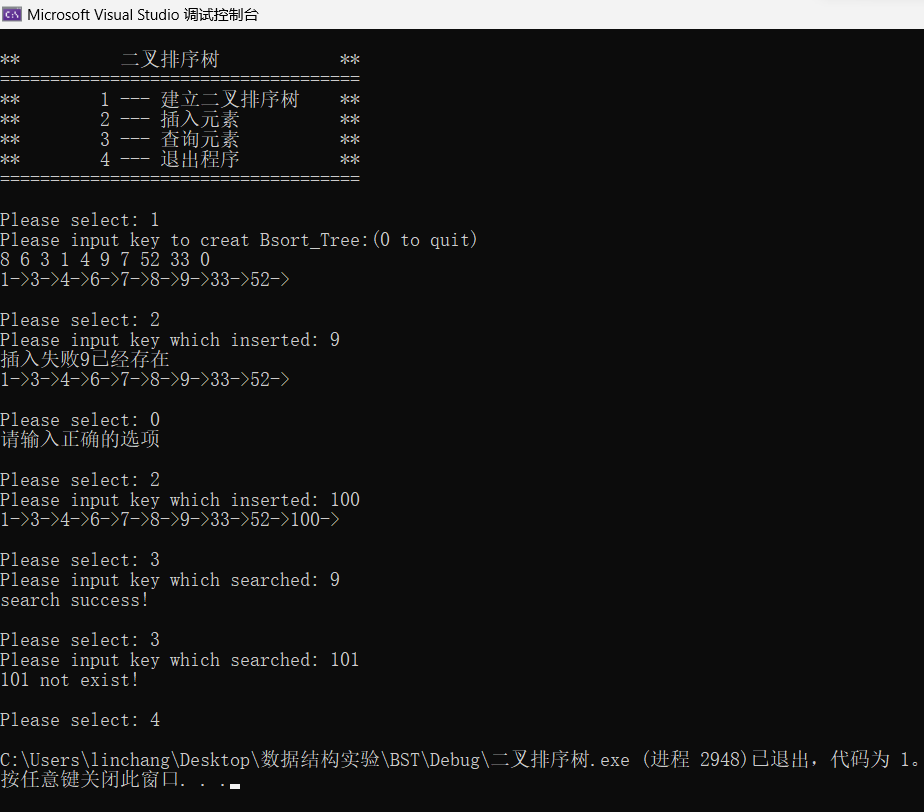
### 4.2 错误输入测试

### 4.2.1 插入已存在数据

测试用例：插入元素9

测试结果：

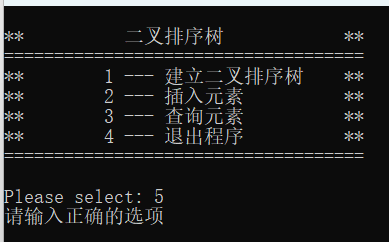
### 4.2.2 查询不存在数据

测试用例：查询101

测试结果：

### 4.2.3 输入不正确的选项

测试用例：输入5

测试结果：

### 4.2 边界测试

### 4.2.3 建立空二叉树

测试用例：选择1，输入0

测试结果：

