数据结构 2022秋 Lab9

> TA：张皓捷 19302010021

说明

本Lab主要关于BST（Binary Search Tree，二叉查找树）。

问题

已知BST的节点定义如下。

```
class BSTNode {
public:
    int element;
    BSTNode *left{};
    BSTNode *right{};
    explicit BSTNode(int element) : element(element) {}
};
```

请你实现一个BST。该BST用来存储一组int整数。**我们规定这个Lab中，BST中的元素不可重复。**

你的BST应该支持如下操作。

```
class BST {
public:
    BST();
    ~BST();
    void insert(int n);
    bool exist(int n);
    void remove(int n);
    int size() const;
    int floor(int n);
};
```

1. **BST()**：构造函数。构造一个没有任何元素的BST。
2. **~BST()**：析构函数。在这里释放你new出来的所有BST节点，防止内存泄漏。
3. **insert(int n)**：向BST中插入元素n。因为元素是不可重复的，因此尝试insert一个已经存在的元素，BST应该不会被修改。
4. **exist(int n)**：判断n是否在BST中存在。如果存在返回true，否则返回false。
5. **remove(int n)**：从BST中删除元素n。
6. **size()**：返回BST中元素的个数。
7. **floor(int n)**：返回BST中，不超过n的最大元素。即返回$max\{x|x \in BST 且 x <= n\}$。如果不存在这样的元素，返回-2147483648。

例子

```
BST bst;
bst.insert(4)
bst.insert(5)
bst.insert(4)
bst.insert(8)
```

```
bst.insert(10)
bst.insert(8)
bst.remove(4)
bst.size() # 应该返回3，因为元素是不可重复的
bst.exist(8) # 应该返回true
bst.exist(4) # 应该返回false
bst.floor(9) # 应该返回8，因为BST中不超过9的最大元素是8
bst.floor(10) # 应该返回10，因为10已经在BST中，BST中不超过10的最大元素就是10自己
bst.floor(3) # 应该返回-2147483648，因为BST中不存在小于等于3的元素
```

提示

- 不要使用stl中自带的set、unordered_set、map和unordered_map
- 你可以修改的文件有：include/BST.h、include/BSTNode.h、src/BST.cpp
- floor的实现应该充分利用BST的特性，而不是直接遍历整个BST
- 使用普通的BST实现，无bug且时间效率正确即可拿到全部分数。勇者也可以尝试自己实现AVL树、红黑树或伸展树等高级数据结构

截止日期

2022年12月11日 周日 23:59~~（和PJ同一天）~~

提交

将你的include/BST.h、include/BSTNode.h、src/BST.cpp三个文件打包后上传到elearning。

文件命名为 学号-姓名-Lab9.zip，例如21302019999-张三-Lab9.zip。