**《数据结构》实验报告6**

**学号：09021227 姓名 ：金桥**

**实验题号：6 实验日期：\_**2022.11.30**\_\_ 实验类型：** 必做

**1．问题描述：**描述实验内容和要求以及需要解决的问题。

**提示：**结合教师课堂讲授内容，仔细分析实验要求，对实验内容进行需求分析。

Write a C++ class AVLtree

·Start with an empty AVL treePerform a sequence of insertions

·Complete an algorithm AVL::Insert to perform each insert

·Complete an algorithm AVL::Remove to delete the element with key k froman AVL tree

·The resulting tree should be restructured if necessary

·Show that the time required from this is o(logn) when there are n nodes inthe tree

测试数据:

·Input file:

·Example 1: DEC, JAN, APR, MAR, JUL, AUG, OCT, FEB, NOV, MAY, JUN

·Example 2: FEB, DEC, NOV, OCT, JUL, SEP, AUG, APR, MAR, MAY, JUN, JAN

·output file:

·the AVL tree following each insertion and state the rotation type (id any) for each insert

·the AVL tree following each deletion and state the rotation type (id any) for each delete

**2．算法思想：**详细描述解决相应问题所需要的算法设计思想。

采用AVL树实现，主要思想仍然是递归。

先实现四种旋转：LL, RR, LR, RL，之后根据不同的bf实施不同的旋转方法。

在每次插入/删除后对节点性质进行检查，并根据bf进行不同的旋转

**3．功能函数：**描述所设计的功能函数。如果有多个函数，需要描述它们之间的关系。

**AVLTreeNode类主要有以下的函数：**

DT getData() const; 获取节点数据

int getBalanceFactor() const; 获取bf

void updateHeight(); 更新节点高度

**AVLTree类主要包括以下函数：**

void insert(const DT& d); 添加节点

void remove(const DT& d); 删除节点

void print() const; 打印AVLTree

AVLTreeNode<DT>\* LL\_rotation(AVLTreeNode<DT>\* x); 进行LL旋转

AVLTreeNode<DT>\* RR\_rotation(AVLTreeNode<DT>\* y); 进行RR旋转

AVLTreeNode<DT>\* LR\_rotation(AVLTreeNode<DT>\* z); 进行LR旋转

AVLTreeNode<DT>\* RL\_rotation(AVLTreeNode<DT>\* z); 进行RL旋转

AVLTreeNode<DT>\* rebalance(AVLTreeNode<DT>\* node); 对node进行平衡

DT removeMinNode(AVLTreeNode<DT>\*& node); 移除子树中最小的节点并返回移除值

**更详细的描述见代码注释**

**4．测试数据：**设计测试数据，或具体给出测试数据。

提示：要求测试数据能全面地测试所设计程序的功能。

**input1.txt:**

11

DEC

JAN

APR

MAR

JUL

AUG

OCT

FEB

NOV

MAY

JUN

**input2.txt:**

12

FEB

DEC

NOV

OCT

JUL

SEP

AUG

APR

MAR

MAY

JUN

JAN

**5．测试情况：**给出程序的测试情况，分析运行结果，显示实验结果截图。

见output1.txt与output2.txt

**运行结果正常，符合预期。**

**6．实验总结：**写出实验过程中遇到的问题，以及问题的解决过程。分析算法的时间复杂度和空间复杂度，总结实验心得体会。

**复杂度分析：**

AVLTree插入与删除时间复杂度O(logn)，空间复杂度O(n)

**心得体会：**

对于C++写数据结构有了更深入的理解

**7. 源代码：**给出项目所有源程序清单。

建议：源程序中应有充分的注释，例如注释每个函数参数的含义、函数返回值的含义、函数的功能、主要语句段的功能，等等。

实验源程序清单

(1) AVLTree.hpp

(2) main.cpp

(3) input1.txt

(4) input2.txt

(5) output1.txt

(6) output2.txt

…