**本次实验我未尝试在Windows平台上编译，不保证能通过。如有需要，请在Linux平台测试代码。**

**实验3 红黑树和顺序统计树**

**王嵩超 PB15000102**

1. **实验要求**

实验1：实现红黑树的基本算法， 对n的取值分别为 12、24、36、48、60，随机生成n 个互异的正整数（K1, K2, K3, ……, Kn）作为节点的关键字，向一棵初始空的红黑树中依次插入这n 个节点，统计算法运行所需时间 ，画出时间曲线。（红黑树采用三叉链表）

实验2：对上述生成的红黑树，找出树中的第n/3小的节点和第n/4小的节点，并删除这两个节点，统计算法运行所需时间 ， 画出时间曲线。

1. **实验环境**

编译环境：Lubuntu 17.10 GCC 6.3

机器内存：16G

时钟主频：2.60GHz

1. **实验过程**

**实验过程本质上仍为写代码、调bug、编译、调bug的反复。**

**代码的修改、调试过程可在github查看：**

<https://github.com/songchaow/Algorithm-Labs/commits/master>

1. **编译步骤**

g++ test.cpp

(test.cpp会include tree.cpp，故只需将test.cpp作为g++参数即可)

1. **代码介绍**

**本实验仍使用C++编写。**

**定义了两个类：BinSearchTree和RedBlackTree。其中RedBlackTree继承BinSearchTree。**

**BinSearchTree实现了基本的插入、删除、查找等功能。而RedBlackTree类中实现了二叉树的旋转、插入删除所对应的fix-up操作等功能。**

**程序入口位于test.cpp，它定义了main函数，用来导入数据，调用类方法以及输出数据。**

1. **性能分析**

下面就插入操作进行性能分析：

每插入10个元素所花时间列表如下：

1-10元素插入耗时:2716

11-20元素插入耗时:7431

21-30元素插入耗时:12213

31-40元素插入耗时:16477

41-50元素插入耗时:19824

51-60元素插入耗时:24537

可以发现时间比较平稳的增长，时间曲线如图。可能是因为数量太小，显示不出对数的增长规律。

将删除操作的时间取为删除第n/3小元素所用时间，时间曲线如图所示：

可看出波动范围较大，可能是因为数据量太小所致。