算法实验二报告

邓一川 PB19000050

本次实验我们要求对以下几种树实现生成，插入和删除的操作

- 二叉排序树
- AVL树
- 红黑树
- B树
- B+树

我的代码文件构成如下:

```
.
├── AVLTree.h
├── BSTree.h
├── BTree.h
├── BpTree.h
├── RBTree.h
├── Report.pdf
├── main.c
```

各个文件内容如命名所示。

不同头文件内部格式基本一致，以红黑树为例，举例如下：

```
RBTree RBTree_gen(int *array, int n, RBTree Tree); // generate an RB Tree
RBTree RBTreeRotateRight(RBTree root); // Right-rotate a node
RBTree RBTreeRotateLeft(RBTree root); // Left-rotate a node
RBTree RBTreeInsert(int data, RBTree Tree); // Insert a data into the tree
RBTree RB_Recursive_Insert(RBNode * p, RBTree Tree); // Recursive insertion
procedure
RBTree RBTreeBalance(RBTree p); // Adjust the tree from node p to top, make it
balanced
RBTree RBTree_search(int data, RBTree Tree); // Search a data in the tree
RBTree RBTree_Delete(int data, RBTree Tree); // delete a data in the tree
RBTree RBTree_node_rm(RBNode *p, RBTree Tree); // remove a node from the tree
RBNode *RBTree_prior(RBNode *p, RBTree Tree); // found the prior node of the
node p
void RB_print(RBTree Tree); // print the tree
```

对于不同的树分别进行generation，insertion和remove操作，得到结果如下

```
size:2000
generation time
Tree_Kind       time_used(microseconds)
BSTree           2378.000000
AVLTree          1301800.700000
RBTree           1913.800000
BTree            915.000000
B+Tree           956.300000
```

```
query time:
Tree_Kind       time_used(microseconds)
BSTree           53.000000
AVLTree          43.700000
RBTree           38.900000
BTree            32.700000
B+Tree           24.500000

delete time:
Tree_Kind       time_used(microseconds)
BSTree           73.300000
AVLTree          2.400000
RBTree           45.300000
BTree            36.600000
B+Tree           22.800000
```

```
size:5000
generation time
Tree_Kind       time_used(microseconds)
BSTree           4793.700000
AVLTree          9946790.500000
RBTree           4688.200000
BTree            2411.800000
B+Tree           2385.900000

query time:
Tree_Kind       time_used(microseconds)
BSTree           54.400000
AVLTree          53.400000
RBTree           55.000000
BTree            37.100000
B+Tree           33.700000

delete time:
Tree_Kind       time_used(microseconds)
BSTree           98.500000
AVLTree          2.400000
RBTree           12.500000
BTree            54.000000
B+Tree           34.400000
```