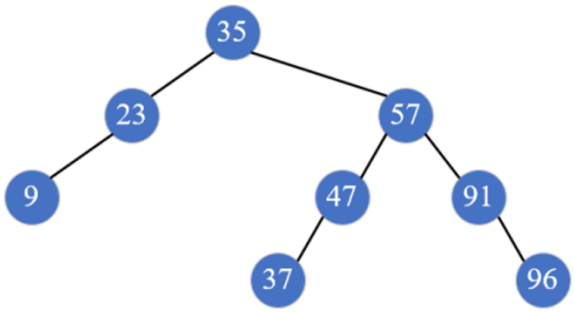
hw1报告

姓名：杜心敏

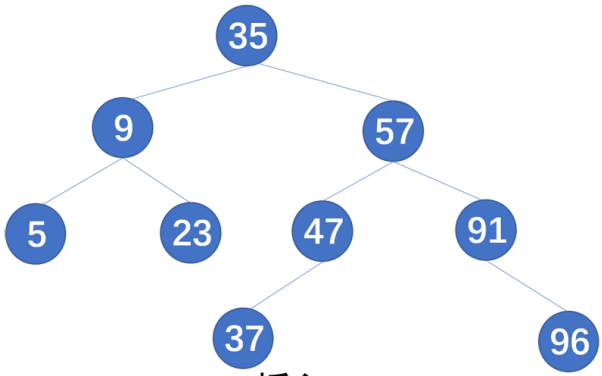
学号：521021910952

EXE1

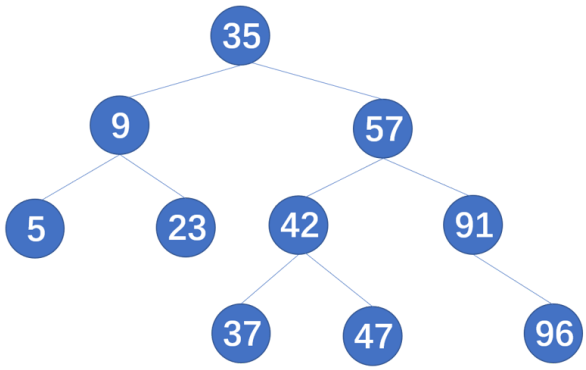
依次插入5，42，51



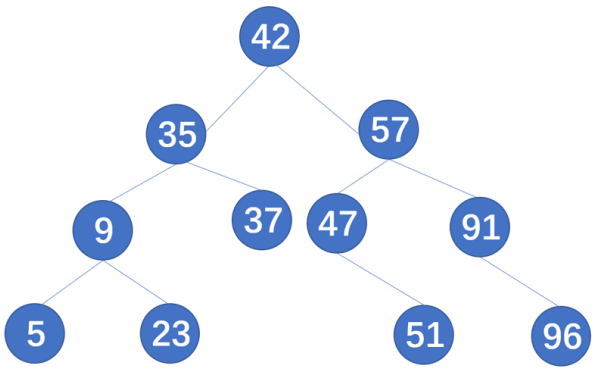
初始



插入5



插入42



插入51

EXE2

实验过程

- 随机生成500，5000，50000个**不重复、乱序的**int型数据

  依次insert，构建树，记录操作时间和树高
- 随机生成500个int型数（都已insert），依次search，记录search的总操作时间

实验结果

```
TEST 1
randomly generate 500 int to test
AVL:: Time for building trees is 1
AVL:: Height is 10
```

```
BST:: Time for building trees is 0
BST:: Height is 18
AVL:: Successfully find 500 keys. Time: 0
BST:: Successfully find 500 keys. Time: 0

TEST 2
randomly generate 5000 int to test
AVL:: Time for building trees is 2
AVL:: Height is 14
BST:: Time for building trees is 1
BST:: Height is 27
AVL:: Successfully find 500 keys. Time: 0
BST:: Successfully find 500 keys. Time: 0

TEST 3
randomly generate 50000 int to test
AVL:: Time for building trees is 26
AVL:: Height is 18
BST:: Time for building trees is 16
BST:: Height is 37
AVL:: Successfully find 500 keys. Time: 0
BST:: Successfully find 500 keys. Time: 1


Process finished with exit code 0
```

结果分析

- BST `insert`、 `find`的时间复杂度都为$O(\log N)$，最差的情况（退化为单链表）$O(N)$
- AVL `insert`、 `find`的时间复杂度都为$O(\log N)$，插入时有时需要调整以保持平衡
- 插入值时，当数据量达到50000时，AVL树耗时明显比BST慢（26：16），在500和5000的测试集中，两者构建的速度差不多，AVL耗时稍多。
- 从树高角度来看，AVL明显比BST更矮。在多次重复实验中也发现，相同的数据量下，AVL树高稳定不变，而BST树高会有明显变化，受插入顺序影响较大。
- 在查找方面，由于只查找了500个值，两个结构都很快完成。在50000测试集中，AVL体现出优势，BST耗时开始增长。