AVL Tree

Given a set of keys, identify a particular permutation to insert the keys into an empty AVL tree. After the insertion of EACH key, following all conditions should be satisfied.

1. The resulting tree should be an AVL tree without requiring any rotation.

2. For every node in the tree, one of the following condition is always satisfied

- both right and left subtree contain either equal number of keys
- number of keys in the left subtree are greater than number of keys in the right subtree by one

Consider the following example input

```
50

20

17

40

30

100

900

700

200

400
```

Your code should take less than 30 seconds to run. In other words, do not use a brute force solution! Your output should the correct sequence of insertions.

Since this is a quiz for the theory course, I expect you to derive the logic for correct sequence of insertions. Therefore, no example output is given.

How your code should compile: `g++ main.cpp`

How your code should run: `./a.out <input1.txt  >output1.txt`