

```

1  template <class T> void AVLTree<T>::_insert(const T & x, treeNode<T> * & t ) {
2      if( t == NULL ) {
3          t = new TreeNode<T>( x, 0, NULL, NULL);
4      }
5
6      else if( x < t->key ) {
7          _insert( x, t->left );
8          int balance = height(t->right) - height(t->left);
9          int leftBalance = height(t->left->right) - height(t->left->left);
10         if ( balance == -2 ) {
11             if ( leftBalance == -1 ) { rotate_____ ( t ); }
12             else { rotate_____ ( t ); }
13         }
14     }
15
16     else if( x > t->key ) {
17         _insert( x, t->right );
18         int balance = height(t->right) - height(t->left);
19         int rightBalance = height(t->right->right) - height(t->right->left);
20         if( balance == 2 ) {
21             if( rightBalance == 1 ) { rotate_____ ( t ); }
22             else { rotate_____ ( t ); }
23         }
24     }
25
26     t->height = 1 + max(height(t->left), height(t->right));
27 }

```