```
template <typename T>
struct Node {
    T value;
    int bfactor;    // Balance Factor
    Node<T>* left;
    Node<T>* right;
};
```

```
template <typename T>
void AVLTree<T>::rotateLeft(Node<T>*& k1)
{
    Node<T>* k2 = k1->right;
    k1->right = k2->left;
    k2->left = k1;
    k1 = k2;
}

template <typename T>
void AVLTree<T>::rotateRight(Node<T>*& k1)
{
    Node<T>* k2 = k1->left;
    k1->left = k2->right;
    k2->right = k1;
    k1 = k2;
}
```

```
template <typename T>
int AVLTree<T>::getBalance(Node<T>* r) const
{
        int balance;
        if (r == nullptr) {
          balance = 0;
        }
        else {
          balance = height(r->left) - height(r->right);
        }
        return balance;
}
```

```
template <typename T>
void AVLTree<T>::updatebFactors(Node<T>* r)
{
        if (r != nullptr) {
              r->bFactor = getBalance(r);
              updatebFactors(r->left);
              updatebFactors(r->right);
       }
}
```