AVL TREES WRITEUP

```
Insert (Node * node, int key)
{       if ( node == NULL)
            return new Node(key);
        if (key < node->key){
        {   insert
            node->left = insert(node->left, key);
        else if (key > node->key)
        {   node->right = insert(node->right,
                                 key);
        }

        node->height = 1 + max(node->height

        node->heigh = 1 + max( heigne(node->left), height
                               (node->right))
        int balance = getBalance(node)
        if ( balance > 1)
        {   if (key < node->left->key)
                return rightRotate(node);
            else if (key {
                node->left = leftRotate(node->left);
                return rightRotate(node)
        }
```

```
        if (balance <-1)
    {   if (key > node->right->key)
            return leftRotate(node)
        else {
            node->left = leftRotate(node->left)
            return leftRotate(node);
        }
    }
}

Delete (Node* node, int key)
    {   if (node == NULL)
            return node;
        if (key < node->key)
        node->left = Delete(node->left, key)
        else if( key > node->key)
        node->right = Delete(node->right, key)

root_node = delete_from_tree (node);

    int balance = getBalance (node);
    if ( balance > 1)
    {   if ( getBalance(node->left) >=0)
            return rightRotate(node),
        else {
            node->left = leftRotate(node->left
            return rightRotate();
    } }
```

```
if ( balance < -1)
    if ( getBalance(root->right) > 0)
        return leftRotate(new_node);
    else {
        root->right = rightRotate(root->right);
        return leftRotate(root);
    }
}
return node;
}
```