TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 최대성
c3d4s19@naver.com

```
* AVL 이진트리 Insert 함수
```

트리 Level 리턴 함수

```
int getTreeLvl(Tree** tree){
    int Lvl, Lvl_L, Lvl_R;

    if(*tree == NULL){
        return 0;
    }
    else if( (*tree)->Right == NULL && (*tree)-
>Left == NULL ){
        return 1;
    }
    else
    {
        Lvl_L = getTreeLvl(&((*tree)->Left));
        Lvl_R = getTreeLvl(&((*tree)->Right));
        Lvl = ( (Lvl_L < Lvl_R) ? Lvl_R : Lvl_L ) + 1;
        return Lvl;
    }
}</pre>
```

트리 균형 상태 리턴 함수

```
int TreeLvlCmp(Tree** tree){
    if(*tree == NULL){
        return 0;
    }
    Tree* rightSubtree = (*tree)->Right;
    Tree* leftSubtree = (*tree)->Left;
    return getTreeLvl(&rightSubtree) -
    getTreeLvl(&leftSubtree);
}
```

RR, LL, RL, LR 회전 후 바뀐 루트노드 반환 함수

```
Tree* RRrot(Tree** tree){
    Tree* tmpTree = (*tree)->Right;
    (*tree)->Right = tmpTree->Left;
    tmpTree->Left = *tree;
    return tmpTree;
Tree* LLrot(Tree** tree){
    Tree* tmpTree = (*tree)->Left;
    (*tree)->Left = tmpTree->Right;
    tmpTree->Right =(*tree);
    return tmpTree;
Tree* RLrot(Tree** tree){
    Tree* tmpTree = (*tree)->Right;
    tmpTree = LLrot(&tmpTree);
    (*tree)->Right = tmpTree;
    tmpTree = RRrot(tree);
    return tmpTree;
Tree* LRrot(Tree** tree){
    Tree* tmpTree = (*tree)->Left;
    tmpTree = RRrot(&tmpTree);
    (*tree)->Left = tmpTree;
    tmpTree = LLrot(tree);
    return tmpTree;
```

루트노드 기준으로 양옆 트리 Level의 균형을 맞추고 바뀐 루트 노드를 반환하는 함수 (루트 노드 아래있는 각 자식노드의 균형은 맞춰주지 않는다)

```
Tree* Rebalance(Tree** root){
    if(*root == NULL){
        return *root;
    }
    Tree* cmpTree = *root;
    Tree* rootTree = *root;
```

```
while(1)
{
    //RR or RL 필요
    if (TreeLvlCmp(&rootTree) > 1)
        cmpTree = rootTree->Right;
        //RR 실행
        if (TreeLvlCmp(&cmpTree) >= 0)
            cmpTree = RRrot(&rootTree);
        }
        //RL 실행
        else
            cmpTree = RLrot(&rootTree);
        }
    }
    //LL or LR 필요
    else if (TreeLvlCmp(&rootTree) < -1)</pre>
        cmpTree = rootTree->Left;
        //LL 실행
        if (TreeLvlCmp(&cmpTree) <= 0)</pre>
            cmpTree = LLrot(&rootTree);
        }
        //LR 실행
        else
        {
            cmpTree = LRrot(&rootTree);
        }
    //균형 잡힘
    else
        break;
    rootTree = cmpTree;
return rootTree;
```

AVL Insert 함수

```
void AVL_InsertDataToTree(Tree** tree, Data data){
    Data cmpData;
    if( *tree == NULL ){
        *tree = getNewTree();
        (*tree)->data = data;
    }
    else if( data <= (*tree)->data ){
        AVL_InsertDataToTree( &((*tree)->Left),
        data );
    }
    else{
        AVL_InsertDataToTree( &((*tree)->Right),
        data );
    }
    *tree = Rebalance(tree);
}
```