

Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 장성환

redmk1025@gmail.com

* AVL 오류 코드 찾기

<LEC CODE>

```
int kinds_of_rot(avl *root, int data)
{
    printf("data = %d\n", data);

    // for RR and RL
    if(rotation_check(root) > 1)
    {
        if(root->right->data > data)
            return RL;

        return RR;
    }
    // for LL and LR
    else if(rotation_check(root) < -1)
    {
        if(root->left->data < data)
            return LR;

        return LL;
    }
}
```

노란색 마커 부분이 AVL 오류 코드이다.

만일 데이터가 이전 데이터와 동일 하다고 할때, 저장되는 값은 동일 데이터의 right 부분으로 저장이 된다.

이때, 만일 왼쪽 노드와 오른쪽 노드의 레벨차이가 2 가 나서 회전이 이루어 진다고 가정해 보자.

중간 노드가 연결리스트 처럼 구성이 되어 있으면 상관이 없으나, 이미 중간 노드에 왼쪽에 값이 입력되어 있으며 RL 회전이 이루어 져야 하는 상황인데 마커 부분대로 회전이 결정되면 RR 회전이 되어 버린다.
여기서 오류가 발생한다.

따라서 각각의 코드를

root → right → left

root → left → right 로 바꾸어 준다.

* AVL 입력 함수 재귀 풀기

<MY CODE>

```
typedef struct _stack{
    void **d_p;
    struct _stack *link;
}stack;

void push(stack **top, void **d_p){

    if(d_p ==NULL)
        return;
    if(*top == NULL){
        *top = (stack*)malloc(sizeof(stack));
        (*top)->d_p = d_p;
        (*top)->link = NULL;
    }
    stack *tmp = (stack*)malloc(sizeof(stack));
    tmp->d_p = d_p;
    tmp->link = *top;
    (*top)=tmp;
}

void** pop(stack **top){
    void **tmp;
    stack * stmp = *top;
    if(*top==NULL){
        return NULL;
    }
    tmp = stmp->d_p;
    (*top)=stmp->link;
    free(stmp);

    return tmp;
}
```

* 해석

강의에서 보여주신 코드를 이용하다 보니 warning 이 떴서 다시 스택을 구성하였다.

스택 구조체의 데이터 부분은 void 더블 포인터를 선언하였다.

Push 및 pop 은 기본 스택 자료구조형과 비슷하다.

Void insert_non_cur 함수의 경우 avl 구조체의 더블 포인터를 받아서 완전 이진 트리화 하는 함수이다. 스택에 저장하는 값을 avl 의 더블 포인터로 저장해야 해당 하는 구조체의 주소값의 left, right 의 주소값을 직접 컨트롤이 가능하다.

스택에 싱글 포인터를 사용할 경우 해당 구조체를 가리키는 주소값을 전달해줄 뿐이라서 이전 구조체의 left 나 right 값에 다가갈 방법이 없다.

더블 포인터를 전달해 주어야만 이전 구조체의 left 의 주소 → 다음 구조체의 주소 → 다음 구조체의 left or right 값에 접근이 가능하다.

즉 이전 구조체의 left 값을 변경하기 위해서 이다.

(AVL 삽입 함수를 구성하였을 때, 더블 포인터를 인자로 받아서 리턴 값 없이 함수 내에서 해결하도록 구성하여 스택도 더블 포인터를 받아야 잘 작동이 된다.

만일, 싱글 포인터를 인자로 받아서 리턴값을 싱글 포인터로 하는 AVL 삽입함수로 설계하였을 경우, 스택을 싱글포인터로 받아도 무리 없이 작동 할 수 있다고 생각한다. 나중에 한번 시도해 보아야 겠다.)

```
}

void insert_non_cur(avl** root,int val){

    stack * top = NULL;
    avl ** tmp = NULL;

    while(*root !=NULL){
        push(&top,(void**)root);
        if((*root)->data > val)
            root= &(*root)->left;
        else if((*root)->data <= val)
            root = &(*root)->right;
    }

    (*root) = get_node();
    (*root)->data = val;

    while(top !=NULL){
        tmp =(avl**)pop(&top);
        (*tmp)->level = update_level(*tmp);
        if(abs(check_rot(*tmp))>1){
            *tmp = rotation(*tmp,check_rot_type(*tmp));
        }
    }
}
```