

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

<자료구조>

2018.03.14-15일 차

강사 - 이상훈

[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

학생 - 안상재

[sangjae2015@naver.com](mailto:sangjae2015@naver.com)

## \* 재귀함수 없는 AVL 트리

### 1. 삭 제

- 알고리즘 순서

1. 삭제할 노드를 root 노드부터 검색한다.
2. 삭제할 노드를 찾으면 자식의 갯수에 따라 알맞은 방법으로 삭제를 한다.
3. 삭제를 하고 나서 해당 노드의 위치부터 root노드로 올라가면서 회전의 필요성을 검사하고 필요하다면 회전을 수행한다.

- push와 pop을 할 때마다 현재 스택의 노드 개수를 cnt 변수를 통해 실시간으로 표시한다.

- 소스 코드

```
void avl_del(avl **root, int data)
{
    int cnt = 0, num, i;
    avl **tmp = root;
    stack *top = NULL;

    while(*tmp)
    {
        printf("Save Stack: %d, data = %d\n", ++cnt, data);
        push(&top, tmp); // 삭제할 노드를 검색하기 위해 이동할 때마다 스택에 노드를 저장한다.

        if((*tmp)->data > data)
            tmp = &(*tmp)->left;
        else if((*tmp)->data < data)
            tmp = &(*tmp)->right;
        else if((*tmp)->left && (*tmp)->right) // 삭제할 노드가 양쪽 자식을 모두 가지고 있을 때
        {
            find_max(&(*tmp)->left, &num); // 왼쪽 서브트리에서 최댓값을 찾는다.
            (*tmp)->data = num; // 노드의 위치는 변경하지 않고 data값만 바꿔준다.
            goto lets_rot; // lets_rot 으로 이동
        }
        else // 삭제할 노드가 자식을 하나만 가지고 있거나 자식이 모두 없을 때
        {
            printf("Do one or nothing node\n");
            (*tmp) = chg_node(*tmp, &top); // 자식 노드를 삭제할 노드의 위치로 이동
            goto lets_rot; // lets_rot으로 이동
        }
    }
}
```

```

if(*tmp == NULL)          // 삭제할 노드가 없을 때
{
    printf("There are no data that you find %d\n", data);
    for(i = 0; i < cnt; i++)
    {
        printf("Extract Stack: %d, data = %d\n", --cnt, data);
        pop(&top);        // 스택에 저장된 노드들을 모두 빼냄.
    }
    return;
}

```

lets\_rot: // 스택에 저장된 노드들을 하나씩 빼내면서 root노드 쪽으로 이동하고, 이동할 때마다 회전의 필요성을 검사하고 필요하다면 회전을 수행함.

```

while(stack_is_not_empty(top))    // 스택이 비어있을때까지 반복.
{
    avl **t = (avl **)pop(&top);
    printf("Extract Stack: %d, data = %d\n", --cnt, data);
    (*t)->lev = update_level(*t);
    if(abs(rotation_check(*t)) > 1)
    {
        printf("Delete Rotation!\n");
        *t = rotation(*t, kinds_of_rot(*t, data));
    }
}
}

```

● 알고리즘 그림

