

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

강사 – Innova Lee (이상훈)

gcccompil3r@gmail.com

학생-김민주

alswnqodrl@naver.com

01



11일차 내용 복습 (avl)

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <time.h>
#define EMPTY 0

typedef struct __avl
{
    int data;
    struct __avl *left;
    struct __avl *right;
}avl;

avl *get_node()
{
    avl *tmp;
    tmp=(avl*)malloc(sizeof(avl));
    tmp->left=EMPTY;
    tmp->right=EMPTY;
    return tmp;
}
```

```
avl *chg_node(avl *root)
{
    if(!root->left)
        return root->right;
    if(!root->right)
        return root->left;
}

avl *update_level(avl *level)
{
    int left_level;
    int right_level;
    if(level == NULL)
        return 0;

    left_level = update_level();
    right_level = update_level();

    if(left_level>right_level)
        return left_level+1;
    else
        return right_level+1;
}
```

01



10일차 내용 복습 (avl)

```
alswnqodri@alswnqodri-900X3K: ~/Downloads
avl *find_max(avl *root, int *data)
{
    if(root->right)
        root-> right = find_max(root->right, data);
    else
    {
        *data = root->data;
        root = chg_node(root);
    }
    return root;
}

avl *find_max(avl *root, int *data)
{
    if(root->right)
        root-> right = find_max(root->right, data);
    else
    {
        *data = root->data;
        root = chg_node(root);
    }
    return root;
}

avl *delete_avl(avl *root, int data)
{
    int num;
    if(root == NULL)
    {
        printf("Not Found\n");
        return NULL;
    }
    else if(root->data > data)
    {
        root->left=delete_avl(root->left, data);
    }
    else if(root->data < data)
    {
        root->right=delete_avl(root->right, data);
    }
    else if(root-> left && root-> right)
    {
        root->left = find_max(root->left, &num);
        root->data = num;
    }
}
```

95,1-8

01



11일차 내용 복습 (avl)

```
alswnqodrl@alswnqodrl-900X3K: ~/Downloads
{
    root->left=delete_avl(root->left, data);
}
else if(root->data < data)
{
    root->right=delete_avl(root->right, data);
}
else if(root-> left && root-> right)
{
    root->left = find_max(root->left, &num);
    root->data = num;
}
else
    root = chg_node(root);
return root;
}

void insert_avl(avl **root, int data)
{
    if(!(*root))
    {
        *root = get_node();
        (*root)->data = data;
        return;
    }
    if((*root) -> data > data)
    {
        insert_avl(&(*root) -> left, data);
    }
    (*root)->level=update_level(*root);
    if(abs(rotation_check(*root))>1)
    {
        printf("Rotation\n")
        *root=rotation(*root, kinds_of_rot(*root, data));
    }
}

void print_avl(avl *root)
{
    acl*tmp=root;
    if(root)
    {
        printf("data=%d", tmp ->data);
    }
    if(root->left)
    {
        printf("left=%d", root ->left->data);
    }
}
```

01



11일차 내용 복습 (avl)

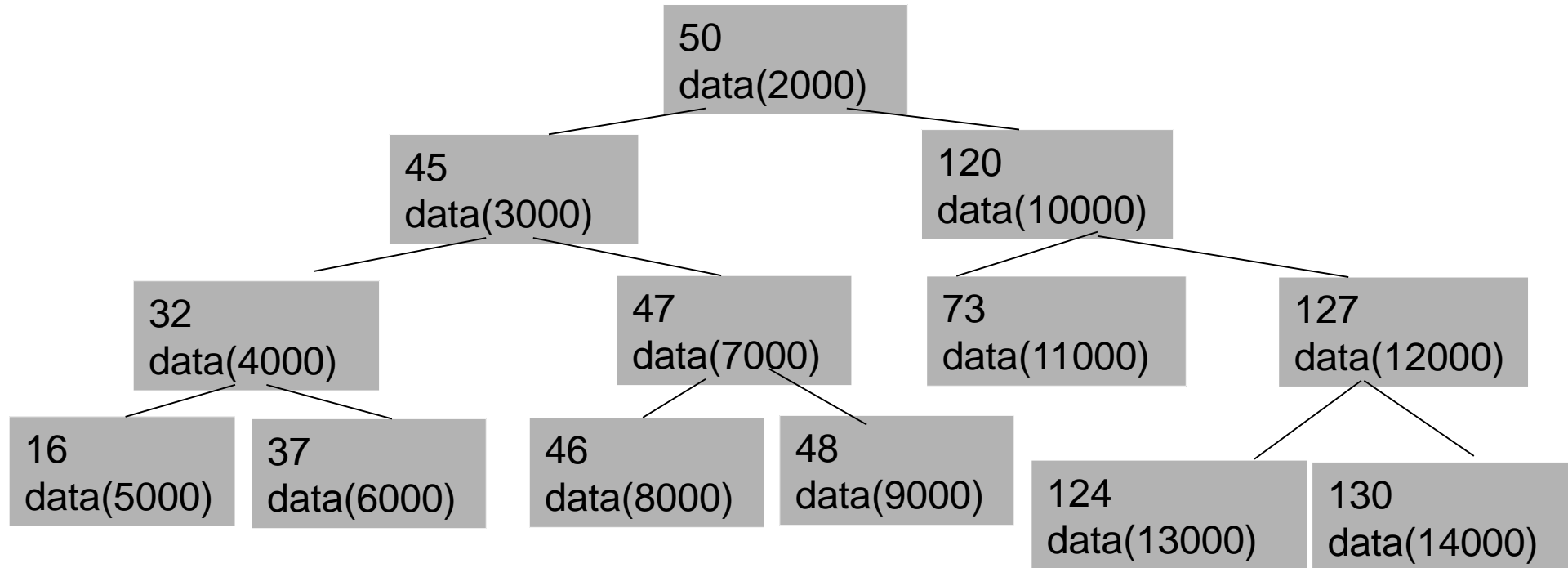
```
alswnqodrl@alswnqodrl-900X3K: ~/Downloads
return root;
}

void insert_avl(avl **root, int data)
{
    if(!(*root))
    {
        *root = get_node();
        (*root)->data = data;
        return;
    }
    if((*root) -> data > data)
    {
        insert_avl(&(*root) -> left, data);
    }
    (*root)->level=update_level(*root);
    if(abs(rotation_check(*root))>1)
    {
        printf("Rotation\n");
        *root=rotation(*root, kinds_of_rot(*root, data));
    }
}

void print_avl(avl *root)
{
    avl*tmp=root;
    if(root)
    {
        printf("data=%d", tmp ->data);
    }
    if(root->left)
    {
        printf("left=%d", root ->left->data);
    }
    else
        printf("left=NULL");
    if(root->right)
    {
        printf("right=%d\n", root->right->data);
    }
    else
        printf("right=NULL\n");
    printf(root->left);
    printf(root->right);
}
```

04

11일차 내용 복습 – 그림그리기 (avl)



<이진트리>

사용 이유: 큐에 비해 검색이 빠르고 효율적으로 가능함

단점: 이진 트리 내부에 큐와 같은 모양이 존재할 경우
큐와 다를 것 없는 검색 속도를 보이며 비효
율적임

<AVL 트리>

사용이유: 보편적인 이진트리의 문제를 해결하기 위해
완전한 이진트리를 구성하여 해당
문제를 보완