

TI DSP,MCU 및 Xilinx Zynq FPGA

프로그래밍 전문가 과정

이름	문지희
학생 이메일	mjh8127@naver.com
날짜	2018/3/7
수업일수	10 일차
담당강사	Innova Lee(이상훈)
강사 이메일	gcccompil3r@gmail.com

1. Tree 삭제부분

```
#include <stdio.h>
#include <stdlib.h>

typedef struct __tree{
    int data;
    struct __tree *left;
    struct __tree *right;
} tree;

tree *get_node(void){
    tree *tmp;
    tmp = (tree *)malloc(sizeof(tree));
    tmp->left = NULL;
    tmp->right = NULL;
    return tmp;
}

void tree_ins(tree **root, int data)
{
    if(*root == NULL)
    {
        *root = get_node();
        (*root)->data = data;
        return;
    }
    else if((*root)->data > data)
        tree_ins(&(*root)->left, data);
    else if((*root)->data < data)
        tree_ins(&(*root)->right, data);
}
```

```
void print_tree(tree *root){
    if(root)
    {
        printf("data = %d, ", root->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");

        print_tree(root->left);
        print_tree(root->right);
    }
}

tree *chg_node(tree *root){
    tree *tmp = root;

    if(!root->right)
        root = root->left;
    else if(!root->left)
        root = root->right;
    free(tmp);
    return root;
}
```

```

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

    return root;
}

tree *delete_tree(tree *root, int data)
{
    int num;
    tree *tmp;
    if(root == NULL)
    {
        printf("Not Found\n");
        return NULL;
    }
    else if(root->data > data)
        root->left = delete_tree(root->left, data);
    else if(root->data < data)
        root->right = delete_tree(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;
}

int main(void)
{
    int i;
    int data[14] = {50, 45, 73, 32, 48, 46, 16,
                   37, 120, 47, 130, 127, 124};

    tree *root = NULL;

    for(i = 0; data[i]; i++)
        tree_ins(&root, data[i]);

    print_tree(root);

    delete_tree(root, 50);
    printf("After Delete\n");

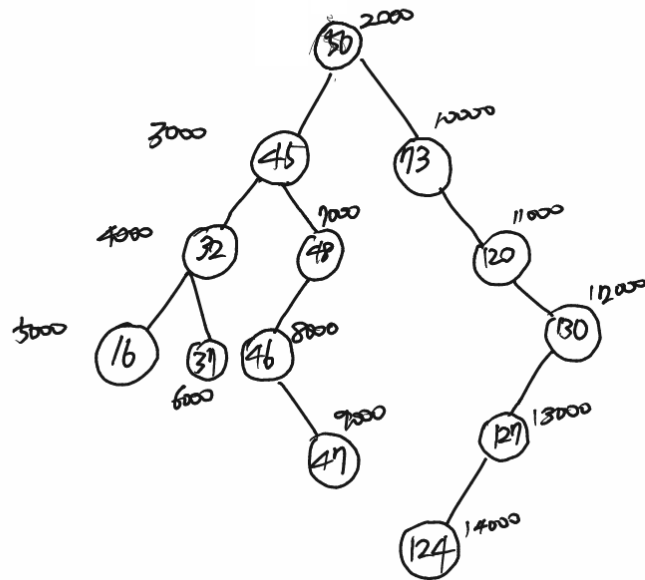
    print_tree(root);

    return 0;
}

```

2. 그림

-45



max

2000
root 1000

delete-free

2000 45
root data num

delete-free

3000 45 37
num 100

find-max

4000 100
root data

find-max

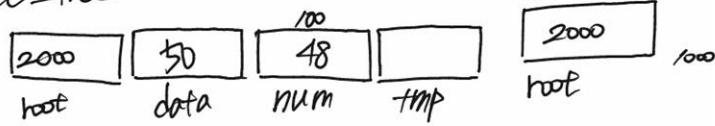
0 100
root data

chg-node

~~6000~~ → 0
root

- 50

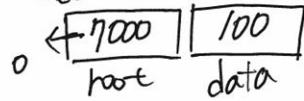
delete_tree



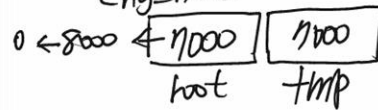
find_max



find_max

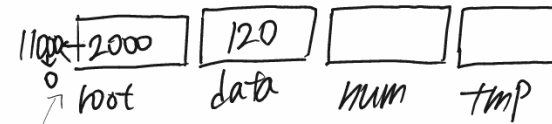


chg_node

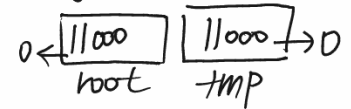


-120

delete tree



chg_node



main

