

2018. 3. 7 수 - 11 회차

과정 : TI, DSP, Xilinx Zynq FPGA, MCU 기반의 프로그래밍 전문가 과정

Prof. 이상훈

gcccompil3r@gmail.com

Stu. 정상용

fstopdg@gmail.com

자료구조_4

AVL_tree

Ex>

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

```
#define EMPTY 0
```

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

```
typedef enum __rot
{
    RR,
    RL,
    LL,
    LR
}rot;
```

```
avl *get_node()
{
    avl *tmp;
    tmp = (avl *)malloc(sizeof(avl));
    tmp -> left = EMPTY;
    tmp -> right = EMPTY;
    tmp -> level = 1;
    return tmp;
}
```

```
int update_level(avl *root)
{
    int left = root -> left ? root -> left -> level : 0;
    int right = root -> right ? root -> right -> level : 0;
```

```

    if(left > right)
        return left + 1;
    return right + 1;
}

int rotation_check(avl *root)
{
    int left = root -> left ? root -> left -> level : 0;
    int right = root -> right ? root -> right -> level : 0;

    return right - left;
}

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

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

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;
}

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

```

```

void print(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");
    print(root -> left);
    print(root -> right);
}

```

```

void avl_ins(avl **root, int data)
{
    if(!(*root))
    {
        (*root) = get_node();
        (*root) -> data = data;
        return ;
    }
    if((*root) -> data > data)
        avl_ins(&(*root) -> left, data);
    else if((*root) -> data < data)
        avl_ins(&(*root) -> right, data);

    (*root) -> level = update_level(*root);

    if(abs(rotation_check(*root)) > 1)
    {

```

```

        printf("Rotation \n");
        *root = rotation(*root, kinds_of_rot(*root, data));
    }
}

avl *debinary(avl *root,int data)
{
    int num;
    avl *tmp;
    if(root == NULL)
    {
        printf("Not Found\n");
        return NULL;
    }
    else if(root->data > data)
        root-> left = debinary(root-> left,data);
    else if(root->data < data)
        root -> right = debinary(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)
{
    avl *root=EMPTY;
    int a[]={50,45,73,32,48,46,16,37,120,127,124};
    int i, num=0;
    int len = sizeof(a)/sizeof(int);

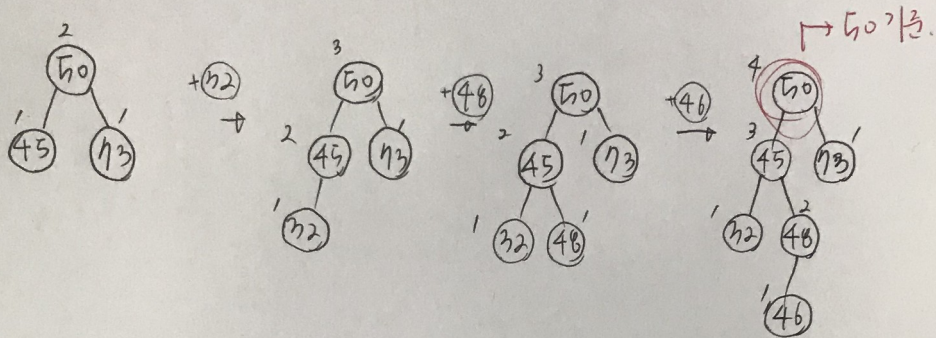
    for(i=0 ; i <len;i++)
        avl_ins(&root, a[i]);
    print(root);
    root=debinary(root,50);
    print(root);

    return 0;
}

```

Sol>

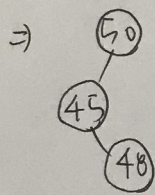
Arr [13] = { 50, 45, 73, 32, 48, 46, 16, 37, 120, 47, 130, 127, 124 }



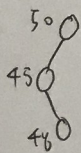
1. 50 기준. 왼쪽 & 오른쪽 밑의 레벨 값은 확인.

2. | 왼쪽 레벨 - 오른쪽 레벨 | ≥ 2 일 경우, 변환 \rightarrow rotation-check

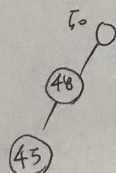
3. 50 기준에서 46으로 가는 route를 확인. \rightarrow kinds of rot.



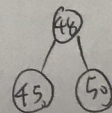
4.



이 경우는

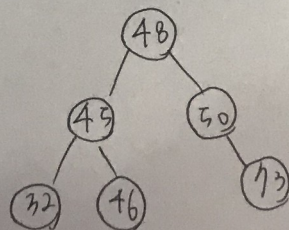


로 바꾼다.



이 경우

\Rightarrow



rotation 종류 4가지.

