

자료구조 및 알고리즘 기말평가

임베디드 시스템 공학과

2015146003

김기덕

1. 소스코드

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

typedef char element;

int data[12] = { 13, 17, 8, 26, 55, 32, 21, 6, 34, 22, 3, 10 };

typedef struct treeNode {
    int key;
    struct treeNode* left;
    struct treeNode* right;
} treeNode;

treeNode* searchBST(treeNode* root, int x) {
    treeNode* p;
    p = root;
    while (p != NULL) {
        if (x < p->key) p = p->left;
        else if (x == p->key) return p;
        else p = p->right;
    }
    printf("\n 찾는 키가 없습니다!");
    return p;
}

treeNode* insertNode(treeNode *p, int x) {
    treeNode *newNode;
    if (p == NULL) {
        newNode = (treeNode*)malloc(sizeof(treeNode));
        newNode->key = x;
        newNode->left = NULL;
        newNode->right = NULL;
        return newNode;
    }
    else if (x < p->key)p->left = insertNode(p->left, x);
    else if (x > p->key)p->right = insertNode(p->right, x);
    else printf("\n 이미 같은 키가 있습니다!\n");
}
```

```

        return p;
    }

void deleteNode(treeNode *root, element key) {
    treeNode *parent, *p, *succ, *succ_parent;
    treeNode *child;

    parent = NULL;
    p = root;
    while ((p != NULL) && (p->key != key)) {
        parent = p;
        if (key < p->key) p = p->left;
        else p = p->right;
    }

    if (p == NULL) {
        printf("\n 찾는 키가 이진 트리에 없습니다!!");
        return;
    }

    if ((p->left == NULL) && (p->right == NULL)) {
        if (parent != NULL) {
            if (parent->left == p) parent->left = NULL;
            else parent->right = NULL;
        }
        else root = NULL;
    }

    else if ((p->left == NULL) || (p->right == NULL)) {
        if (p->left != NULL) child = p->left;
        else child = p->right;

        if (parent != NULL) {
            if (parent->left == p) parent->left = child;
            else parent->right = child;
        }
        else root = child;
    }

    else {
        succ_parent = p;

```

```

        succ = p->left;
        while (succ->right != NULL) {
            succ_parent = succ;
            succ = succ->right;
        }
        if (succ_parent->left == succ)succ_parent->left = succ->left;
        else succ_parent->right = succ->left;
        p->key = succ->key;
        p = succ;
    }
    free(p);
}

void displayInorder(treeNode* root) {
    if (root) {
        displayInorder(root->left);
        printf("%d_", root->key);
        displayInorder(root->right);
    }
}

void displayD_Node(treeNode* root) {
    if (root) {
        displayD_Node(root->left);
        if (root->left == NULL && root->right == NULL)
        {
            printf("%d_", root->key);
        }
        displayD_Node(root->right);
    }
}

void menu() {
    printf("\n*-----*");
    printf("\n\t1 : 트리출력");
    printf("\n\t2 : 트리검색");
    printf("\n\t3 : 단말노드");
    printf("\n\t4 : 종료");
    printf("\n*-----*");
    printf("\n메뉴입력 >> ");
}

```

```

int main() {
    treeNode* root = NULL;
    treeNode* foundedNode = NULL;
    int choice, key;

    root = insertNode(root, data[0]);
    for (int i = 1; i <= 11; i++)
    {
        insertNode(root, data[i]);
    }

    system("title Kimkideok");
    while (1) {
        menu();
        scanf_s(" %d", &choice);

        switch (choice) {
            case 1: printf("\t[이진 트리 출력] ");
                    displayInorder(root);
                    printf("\n");
                    break;
            case 2: printf("찾을 숫자를 입력하세요 : ");
                    scanf_s(" %d", &key);
                    foundedNode = searchBST(root, key);
                    if (foundedNode != NULL)
                        printf("\n\t\t\t\t\t%d\t\t\t\t\t숫자를\t\t\t\t\t찾았습니다!\t\t\t\t\t\n",
foundedNode->key);
                    else
                        printf("\n\t\t\t\t\t숫자를\t\t\t\t\t찾지\t\t\t\t\t못했습니다.\t\t\t\t\t\n");
                    break;
            case 3: printf("\t[단말 노드 출력] ");
                    displayD_Node(root);
                    printf("\n");
                    break;
            case 4:
                    return 0;
            default:
                    printf("없는 메뉴입니다. 메뉴를 다시 선택하세요! \n");
                    break;
        }
    }
}

```

```
}  
}
```

2. 실행화면

```
Kimkideok  
*-----*  
1 : 트리 출력  
2 : 트리 검색  
3 : 단말 노드  
4 : 종료  
*-----*  
메뉴입력 >> 1  
[이진 트리 출력] 3_6_8_10_13_17_21_22_26_32_34_55_  
*-----*  
1 : 트리 출력  
2 : 트리 검색  
3 : 단말 노드  
4 : 종료  
*-----*  
메뉴입력 >> 3  
[단말 노드 출력] 3_10_22_34_  
*-----*  
1 : 트리 출력  
2 : 트리 검색  
3 : 단말 노드  
4 : 종료  
*-----*  
메뉴입력 >> 2  
찾을 숫자를 입력하세요 : 20  
  
찾는 키가 없습니다!  
숫자를 찾지 못했습니다.  
*-----*  
1 : 트리 출력  
2 : 트리 검색  
3 : 단말 노드  
4 : 종료  
*-----*  
메뉴입력 >> 2  
찾을 숫자를 입력하세요 : 32  
  
32 숫자를 찾았습니다!  
*-----*  
1 : 트리 출력  
2 : 트리 검색  
3 : 단말 노드  
4 : 종료  
*-----*  
메뉴입력 >>
```