



2023_Autumn_Algorithm_Study_1주차 과제 기록

[통계](#) [수정](#) [삭제](#)

jaesung2da · 방금 전 · 비공개

0

[algorithm_study](#) [sejong](#) [smarcle](#)

1주차 과제는 자료구조 복습 차원에서

2346번: 풍선 터뜨리기

10828번: 스택

10845번: 큐

1991번: 트리 순회

연결리스트, 스택, 큐, 트리 예제문제 4개 중 2개를 풀어오고 나머지 2개는 스터디 시간에 풀어보기로 했다.

망했다... 하나도 기억이 안난다. 연결리스트는 어찌저찌 기억나는데 스택, 큐, 트리 구조체 공부를 다시 해야할 것 같다.

스터디에서 구입한 '이것이 자료구조+알고리즘이다' 책으로 개념부터 다시 익혀봐야 할 것 같다.

10828번: 스택을 먼저 풀기로 했다. ~~왠지 연결리스트는 스터디 시간에 풀 수 있을 것 같아서~~

문제

정수를 저장하는 스택을 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 다섯 가지이다.

- push X: 정수 X를 스택에 넣는 연산이다.
- pop: 스택에서 가장 위에 있는 정수를 빼고, 그 수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 스택에 들어있는 정수의 개수를 출력한다.
- empty: 스택이 비어있으면 1, 아니면 0을 출력한다.
- top: 스택의 가장 위에 있는 정수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.

입력

첫째 줄에 주어지는 명령의 수 N ($1 \leq N \leq 10,000$)이 주어진다. 둘째 줄부터 N 개의 줄에는 명령이 하나씩 주어진다. 주어지는 정수는 1보다 크거나 같고, 100,000보다 작거나 같다. 문제에 나와있지 않은 명령이 주어지는 경우는 없다.

출력

출력해야하는 명령이 주어질 때마다, 한 줄에 하나씩 출력한다.

예제 입력 1 복사

```
14
push 1
push 2
top
size
empty
pop
pop
pop
size
empty
pop
push 3
empty
top
```

예제 출력 1 복사

```
2
2
0
2
1
-1
0
1
-1
0
3
```

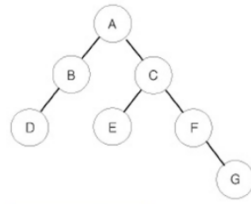
제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이
66430890	tlawotjd	10828	맞았습니다!!	1380 KB	4 ms	C99 / 수정	1624 B
66430781	tlawotjd	10828	컴파일 에러			C# / 수정	1624 B
66430623	tlawotjd	10828	컴파일 에러			C# / 수정	1624 B

문제를 보니 저번 자료구조 수업 예제 문제와 비슷해서 문제를 보니 우리 학교 문제가 더 어려운 거 같긴하다. 근데 문제 조건 중에 스택을 배열로 구현하라는 조건이 붙었고 실제로 나도 oj에 배열로 문제를 제출했길래 이번에 스택문제는 연결리스트로 구현해보았다.

제출

문제

이진 트리를 입력받아 전위 순회(preorder traversal), 중위 순회(inorder traversal), 후위 순회(postorder traversal)한 결과를 출력하는 프로그램을 작성하시오.



예를 들어 위와 같은 이진 트리가 입력되면,

- 전위 순회한 결과 : ABCDEFG // (루트) (왼쪽 자식) (오른쪽 자식)
- 중위 순회한 결과 : DBAECFG // (왼쪽 자식) (루트) (오른쪽 자식)
- 후위 순회한 결과 : DBEGFCA // (왼쪽 자식) (오른쪽 자식) (루트)

가 된다.

입력

첫째 줄에는 이진 트리의 노드의 개수 $N(1 \leq N \leq 26)$ 이 주어진다. 둘째 줄부터 N 개의 줄에 걸쳐 각 노드와 그의 왼쪽 자식 노드, 오른쪽 자식 노드가 주어진다. 노드의 이름은 A 부터 차례대로 알파벳 대문자로 매겨지며, 항상 A가 루트 노드가 된다. 자식 노드가 없는 경우에는 .으로 표현한다.

출력

첫째 줄에 전위 순회, 둘째 줄에 중위 순회, 셋째 줄에 후위 순회한 결과를 출력한다. 각 줄에 N 개의 알파벳을 공백 없이 출력하면 된다.

예제 입력 1 복사

```

7
A B C
B D .
C E F
E . .
F . G
D . .
G . .
  
```

예제 출력 1 복사

```

ABCDEFG
DBAECFG
DBEGFCA
  
```

아 이거 널 때 언어 C#으로 내야하는 줄 알았는데 C99로 내야하는 것 같다.

내가 작성한 코드

```

#pragma warning(disable:4996)
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
  
```

```

typedef struct Node {
    int data;
    struct Node* next;
} Node;
  
```

```

typedef struct Stack {
    Node* top;
    int size;
} Stack;
  
```

```

void initStack(Stack* stack) {
    stack->top = NULL;
  
```

```
    stack->size = 0;
}

int empty(Stack* stack) {
    if (stack->top == NULL) {
        return 1;
    }
    else {
        return 0;
    }
}

void push(Stack* stack, int N) {
    Node* new = (Node*)malloc(sizeof(Node));
    new->data = N;
    new->next = stack->top;
    stack->top = new;
    stack->size++;
}

int pop(Stack* stack) {
    if (stack->top == NULL) {
        printf("-1\n");
        return -1;
    }
    else {
        Node* temp = stack->top;
        int N = temp->data;
        stack->top = stack->top->next;
        stack->size--;
        free(temp);
        printf("%d\n", N);
        return N;
    }
}

int size(Stack* stack) {
    printf("%d\n", stack->size);
    return stack->size;
}

int top(Stack* stack) {
    if (stack->top == NULL) {
        printf("-1\n");
        return -1;
    }
    else {
        printf("%d\n", stack->top->data);
        return stack->top->data;
    }
}

int main() {
    Stack* stack = (Stack*)malloc(sizeof(Stack));
    initStack(stack);

    int cnt;
    scanf("%d", &cnt);
```

```

    getchar();

    while (cnt > 0) {
        char cmd[6];
        scanf("%s", cmd);
        if (strcmp(cmd, "push") == 0) {
            int N;
            scanf("%d", &N);
            getchar();
            push(stack, N);
            cnt--;
        }
        else if (strcmp(cmd, "pop") == 0) {
            pop(stack);
            cnt--;
        }
        else if (strcmp(cmd, "size") == 0) {
            size(stack);
            cnt--;
        }
        else if (strcmp(cmd, "empty") == 0) {
            printf("%d\n", empty(stack));
            cnt--;
        }
        else if (strcmp(cmd, "top") == 0) {
            top(stack);
            cnt--;
        }
    }

    free(stack);
    return 0;
}

```

문제를 풀 때 어떻게 풀까하는 고민보다 아 이 문법이 맞나... 애도 동적할당을 해야하나...문자열 비교 어떻게 했지... 등등 너무 오랜만에 코딩을 해서 생각보다 시간이 많이 걸렸다. 구글링을 아예 안한 것은 아니지만 구조체 선언 방식? 문법 위주로 부분부분만 검색해서 완성한 것 같다.

1991번: 트리 순회를 두번째로 풀었다.

저번 자료구조 수업 때 공부를 너무 안한 것 같다.

~~선택과 집중을 통해 연결리스트, 스택, 큐까지만 실습문제를 열심히 풀고 트리는...~~

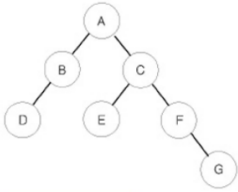
그래서 처음부터 한다는 마음가짐으로 저번 자료구조 교안과 책을 참고하여 개념을 익히려고 했으나 MZ세대 답게

<https://www.youtube.com/playlist?list=PL1xKqHsVFgkvjsEOITU-h7HAOmD1Bpadd>

저번 자료구조 시험기간에 공부한 것처럼 최유경 교수님 유튜브 들어가서 강의를 들었다..ㅎㅎ

문제

이진 트리를 입력받아 전위 순회(preorder traversal), 중위 순회(inorder traversal), 후위 순회(postorder traversal)한 결과를 출력하는 프로그램을 작성하시오.



예를 들어 위와 같은 이진 트리가 입력되면,

- 전위 순회한 결과 : ABDCEFG // (루트) (왼쪽 자식) (오른쪽 자식)
- 중위 순회한 결과 : DBAECFG // (왼쪽 자식) (루트) (오른쪽 자식)
- 후위 순회한 결과 : DBEGFCA // (왼쪽 자식) (오른쪽 자식) (루트)

가 된다.

입력

첫째 줄에는 이진 트리의 노드의 개수 $N(1 \leq N \leq 26)$ 이 주어진다. 둘째 줄부터 N개의 줄에 걸쳐 각 노드와 그의 왼쪽 자식 노드, 오른쪽 자식 노드가 주어진다. 노드의 이름은 A 부터 차례대로 알파벳 대문자로 매겨지며, 항상 A가 루트 노드가 된다. 자식 노드가 없는 경우에는 .으로 표현한다.

출력

첫째 줄에 전위 순회, 둘째 줄에 중위 순회, 셋째 줄에 후위 순회한 결과를 출력한다. 각 줄에 N개의 알파벳을 공백 없이 출력하면 된다.

예제 입력 1 복사

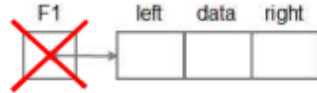
```
7
A B C
B D .
C E F
E . .
F . G
D . .
G . .
```

예제 출력 1 복사

```
ABDCEFG
DBAECFG
DBEGFCA
```

[문제 1] 위에서 설명한 방식대로 트리 정보와 탐색 정보가 주어졌을 때, 트리를 생성하고 탐색 도중 방문하는 노드의 번호를 차례로 출력하는 프로그램을 작성하시오.

- 트리 1주차 실습에서처럼 모든 노드마다 자신의 위치를 가리키는 포인터변수를 만들어 사용하면 안 됨.



- 오직 루트(root) 노드에 대해서만 허용. 즉, 트리는 루트 노드를 통해서만 접근 가능

입력 상세:

- 트리 정보
 - 첫 째 줄에 노드의 개수 n 이 주어진다.
 - 다음 n 개의 줄에, 전위(선위) 순회 순서로 노드의 정보가 주어진다. (위 설명 참조)
- 탐색 정보 (트리 정보가 모두 주어진 후)
 - 탐색 횟수 s 가 주어진다.
 - 다음 s 개의 줄에, 탐색 정보가 주어진다. (각 탐색은 루트 노드에서 새로 시작)
 - 하나의 탐색 정보는 공백없이, 'L'과 'R'로 구성된 문자열(최대 길이 100)로 주어진다.
 - 유효하지 않은 탐색 정보는 주어지지 않는다. 예를 들어, 위 트리에서 "RRR" 과 같은 탐색 정보는 유효하지 않다. 두 번 오른쪽 자식을 따라 내려가면 노드 10인데, 노드 10의 오른쪽 자식은 정의되지 않았다.

출력 상세:

- 탐색 시 방문하는 노드의 번호를 순서대로 출력한다. (하나의 줄에 한 번의 탐색 결과 출력)

입력 예시 1

```

9           ↳ 노드 개수
5 3 9
3 8 15
8 0 2
2 0 0
15 0 0
9 7 10
7 12 0
12 0 0
10 0 0
3           ↳ 탐색 횟수
RLL
LL
LR
  
```

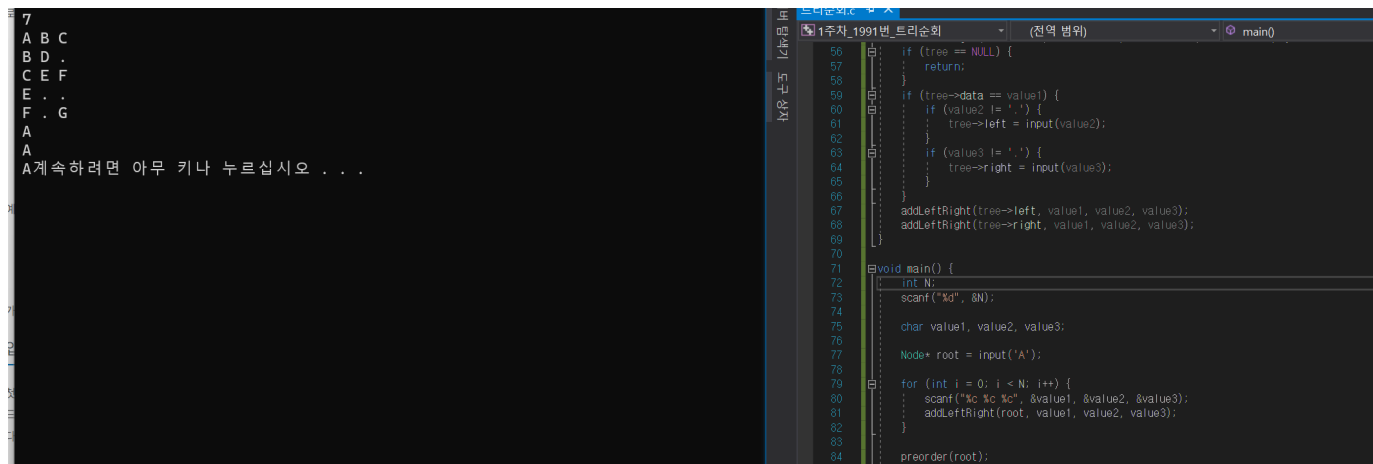
출력 예시 1

```

□ 5 9 7 12   ↳ 첫 번째 탐색 결과
□ 5 3 8       ↳ 두 번째 탐색 결과
□ 5 3 15      ↳ 두 번째 탐색 결과
  
```

트리 문제의 경우 저번 자료구조 트리3-1 문제에서 트리 정보를 입력 받는 방식이 비슷해서 트리3-1을 저번에 작성했던 OJ코드와 함께 먼저 공부해주었다.

다른 점이 있다면 data정보가 숫자냐, 문자냐의 차이 정도? int와 char표시의 차이 정도가 있었다.



제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이
66449644	tlawotjd	1991	맞았습니다!!	1116 KB	0 ms	C99 / 수정	1560 B
66449622	tlawotjd	1991	틀렸습니다			C99 / 수정	1559 B
66449552	tlawotjd	1991	런타임 에러 (NZEC)			C99 / 수정	1520 B
66449531	tlawotjd	1991	틀렸습니다			C99 / 수정	1549 B
66449496	tlawotjd	1991	런타임 에러 (NZEC)			C99 / 수정	1550 B
66448468	tlawotjd	1991	틀렸습니다			C99 / 수정	1837 B
66446955	tlawotjd	1991	런타임 에러 (NZEC)			C99 / 수정	2467 B

하지만 신기하게도 scanf받는 부분에서 (%c %c %c)로 입력받으니 위와 같은 오류가 나고 한 칸
띄어서 (%c %c %c)로 받으니 문제가 해결되었다... 이거 때문에 시간을 얼마나
날린건지

내가 작성한 코드

```
#pragma warning(disable:4996)
#include <stdio.h>
#include <stdlib.h>

typedef struct node {
    char data;
    struct node* left;
    struct node* right;
} Node;

Node* input(char data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
```



```
newNode->data = data;
newNode->left = NULL;
newNode->right = NULL;
return newNode;
}

void freeTree(Node* tree) {
    if (tree == NULL) {
        return;
    }
    freeTree(tree->left);
    freeTree(tree->right);
    free(tree);
}

void preorder(Node* tree) {
    if (tree == NULL) {
        return;
    }
    printf("%c", tree->data);
    preorder(tree->left);
    preorder(tree->right);
}

void inorder(Node* tree) {
    if (tree == NULL) {
        return;
    }
    inorder(tree->left);
    printf("%c", tree->data);
    inorder(tree->right);
}

void postorder(Node* tree) {
    if (tree == NULL) {
        return;
    }
    postorder(tree->left);
    postorder(tree->right);
    printf("%c", tree->data);
}

void addLeftRight(Node* tree, char value1, char value2, char value3) {
    if (tree == NULL) {
        return;
    }
    if (tree->data == value1) {
        if (value2 != '.') {
            tree->left = input(value2);
        }
        if (value3 != '.') {
            tree->right = input(value3);
        }
    }
    addLeftRight(tree->left, value1, value2, value3);
    addLeftRight(tree->right, value1, value2, value3);
}
```

```
int main() {
    int N;
    scanf("%d", &N);

    char value1, value2, value3;

    Node* root = input('A');

    for (int i = 0; i < N; i++) {
        scanf("%c %c %c", &value1, &value2, &value3);
        addLeftRight(root, value1, value2, value3);
    }

    preorder(root);
    printf("\n");
    inorder(root);
    printf("\n");
    postorder(root);

    freeTree(root);

    return 0;
}
```

분명 visualstudio에서는 나오는데 런타임 에러가 자꾸떠서 눌러보니 백준oj에서 c언어를 사용할 때 void main()을 쓰면 그렇게 된다더라... int main으로 바꿔주니 해결됨 암튼 난 과제 다했음.



심재성

0개의 댓글

댓글을 작성하세요

댓글 작성

