



2023 Algorithm Study 1Week

[통계](#) [수정](#) [삭제](#)

sookyoung0620 · 방금 전 · 비공개

0

2023 Algorithm Study



▼ 목록 보기

1/1



이중연결리스트 개념

백준 2346번

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

항상 귀찮게 oj 들어가서 복붙하는 거.. 여기에 적어놔야지

```
typedef struct ListNode {
    int value;
    struct ListNode* prev;
    struct ListNode* next;
}ListNode;
```

먼저 풍선에 대한 구조체를 정의한다

이중연결리스트라면 이 형태를 그냥 외우자!

```
int main() {
    int N; //풍선의 수
    scanf("%d", &N);
    struct ListNode* balloons = (struct ListNode*)malloc(N * sizeof(struct ListNode));
```

이중연결리스트 생성을 위한 동적메모리할당

```
struct Balloon* header = (struct Balloon*)malloc(sizeof(struct Balloon));
struct Balloon* tail = (struct Balloon*)malloc(sizeof(struct Balloon));
header->prev = tail->next = NULL;
header->next = &balloons[0];
tail->prev = &balloons[N - 1];
balloons[0].prev = header;
balloons[N - 1].next = tail;
```

헤더와 테일 초기화... 원래 `init()`함수로 구현해보려고 했으나 첫번째 풍선과 풍선이 이어져 있다는 점이... 신경쓰여서 도전해보다가 포기!
포인터를 다 까먹어서.. `&balloons[0]`은 첫 번째 요소의 주소를 반환..
& 잊지말자

```
for (int i = 0; i < N; i++) {
    scanf("%d", &(balloons[i].value));
    balloons[i].prev = &balloons[(i - 1 + N) % N];
    balloons[i].next = &balloons[(i + 1) % N];
}
```

풍선의 안에 종이에 적힌 수 입력 받고 이중 연결 리스트 구성

```
struct Balloon* now = &balloons[0]; // 현재 위치

for (int i = 0; i < N; i++) {
    printf("%d ", now - balloons + 1); // 현재 풍선 번호 출력
    int move = now->value;
```

```

    now->prev->next = now->next;
    now->next->prev = now->prev;

    if (move > 0) { //양수이면
        while (move > 0) {
            now = now->next; //오른쪽으로 이동
            move--;
        }
    } else { //음수이면
        while (move < 0) {
            now = now->prev; //왼쪽으로 이동
            move++;
        }
    }
}

free(balloons); // 동적 할당 해제

return 0;

}

```

```

now->prev->next = now->next;
now->next->prev = now->prev;

```

이 코드... 알고리즘 1번인가 2번 과제에도 있어서 굉장히 혼란스러웠지만 이해했땅!

now->prev는 현재 위치 전 풍선을 가리키는 것이다... 즉 (now->prev)->next라고 이해하면 편한데
전 풍선의 next 포인터인 것이다~ prev, value, next가 있으면

원래는 현재 풍선의 전 풍선 next와 현재 풍선의 prev가 연결되어 있고, 현재 풍선의 next가 현재 풍
근데 풍선이 터지므로 현재 풍선의 전 풍선 next와 현재 풍선의 다음 풍선 prev를 연결했다고 생각할

백준 10828

```

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

typedef struct stack {
    char* Stack;
    int size;
    int t;
} STACK;

```

Stack: 문자열을 저장하는 배열
 size: 스택의 최대 크기
 t: 스택의 탑(Top) 인덱스를 가리킴

```
STACK* initStack(int stackN) {
STACK* stack = (STACK*)malloc(sizeof(STACK));
stack->Stack = (char*)malloc(sizeof(char) * 1000000);
//stack->size = stackN;
stack->t = -1;
return stack;
}
```

자료구조때 최대 스택 길이?를 설정해서 풀었는데 여기 문제에는 조건이 안나와있어서 1000000로 잡고 했당...

```
void push(STACK* stack, char ch) {

stack->t++;
stack->Stack[stack->t] = ch;
}
```

탑 인덱스 증가와 ch 저장

```
char pop(STACK* stack) {
if (stack->t == -1) {
return -1;
}
char value = stack->Stack[stack->t];
stack->t--;
return value;
}
```

여기에 문제가 있는 거 같은데 사실 백준 예시 입력을 넣었을 때 뜬금없는 50 49가 출력됨... 하지만.. 찾지 못했음..ㅠ

```

void top(STACK* stack) {
    if (stack->t == -1) {
        printf("-1\n");
        return;
    }
    printf("%c\n", stack->Stack[stack->t]);
}

int empty(STACK* stack) {
    if (stack->t == -1) {
        return 1;
    }
    else {
        return 0;
    }
}

int size(STACK* stack) {
    return stack->t + 1;
}

```

탐 인덱스의 1더한 것이 길이!

```

int main() {
    int stackN = 1000000, N;
    char command[6];
    char ch;

    scanf("%d", &N);
    getchar();

    STACK* stack = initStack(stackN);

    for (int i = 0; i < N; i++) {
        scanf("%s", command);
        getchar();

        if (strcmp(command, "push") == 0) {
            scanf("%c", &ch);
            getchar();
            push(stack, ch);
        }
        else if (strcmp(command, "pop") == 0) {
            printf("%d\n", pop(stack));
        }
        else if (strcmp(command, "top") == 0) {
            top(stack);
        }
        else if (strcmp(command, "empty") == 0) {
            printf("%d\n", empty(stack));
        }
    }
}

```

```
    }  
    else if (strcmp(command, "size") == 0) {  
        printf("%d\n", size(stack));  
    }  
}  
  
free(stack->Stack);  
free(stack);  
  
return 0;  
}
```



윤수경

0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by
Stellate