



2023 Algorithm Study Week 7 - 과제

통계 수정 삭제

cheonwon · 방금 전 · 비공개

❤ 0

2023 Algorithm Study smarcle

탐색트리를 할 주치의 문제인데 탐색트리를 쓰면 시간초과가 날것 같고, 밑의 알고리즘 분류를 보면 해시라고 써져 있으며, 근데 정작 해시로는 어떻게 풀어야 할지 감이 안잡히는 그런 미묘한 문제였다...

나는야 포켓몬 마스터 이다솜 <백준 1620>

기본적으로 탐색트리도 좋은 자료형이지만 c언어에서는 최적화된 메소드가 없어서 연결리스트로 직접 구현을 해야 하며, 어떤 노드를 새로 삽입하거나 삭제하는데 있어서 꽤 많은 과정(한번 삽입할때마다 삽입할 위치를 찾아주어야 함(treeSearch) + expand External 등등)을 요구하므로 시간초과가 날 가능성이 높다고 생각했다. 시간 제한이 2초밖에 안되기때문에 배열로 구현하고자 했다.

배열로 구현할 경우 들어오는 번호를 그대로 인덱스로 가질 수 있기에 구조체를 따로 사용하지 않아도 되며, 동적할당을 통한다면 메모리를 최대한 적게 사용할 수 있어서 좋다. 또한 숫자로 들어오는 입력의 경우 그냥 바로 배열로 접근해서 탐색할 수 있으므로 최소한의 시간으로 탐색이 가능하다.

근데 문제는 문자로 들어오는 입력의 경우 일치하는 문자열 탐색을 위해 이중 for문을 사용한다는 것인데.... 그래서 이 경우엔 시간복잡도가 $O(n^2)$ 라 시간초과가 날 확률이 높다. 그래서 최악의 경우에 시간복잡도가 $O(n^2)$ 이라 시간초과가 날 것 같았는데, 역시나 시간초과가 나왔다.

첫번째, 일단 생각한 대로 의 흐름만 반영해서 그대로 짰 코드이다.

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

```

int main() {
    int M, n;

    scanf("%d%d", &M, &n);
    getchar();
    char **ch = (char**)malloc(sizeof(char*) * M);
    for (int i = 0; i < M; i++) {
        ch[i] = (char*)malloc(sizeof(char) * 21);
        scanf("%s", ch[i]);
        getchar();
    }

    char tmp[21]; //이것도 동적할당.
    for (int i = 0; i < n; i++) {
        scanf("%s", tmp);
        getchar();
        if (tmp[0] >= '0' && tmp[0] <= '9') { //맨처음자리가 숫자면
            int x = atoi(tmp);
            printf("%s\n", ch[x - 1]);
        }
        else {
            for (int j = 0; j < M; j++) {
                if (strcmp(tmp, ch[j]) == 0) {
                    printf("%d\n", j + 1);
                    break;
                }
            }
        }
    }

    return 0;
}

```

두번째 코드는 혹시나 코드에서 사용하는 메소드들 중에서 메모리 끝까지 탐색하는 메소드가 있나(혹시나 strcmp, atoi이런것들이 끝까지 봐서 시간이 오래 걸리나 싶었음 - 근데 사실 이런 문자열 조작 함수들은 문자열 특성을 이용해 널문자까지 인식하여 전체가 아니라 중간에 중단되는 것으로 알고 있음.) 싶어서 동적할당으로 메모리를 최대한 최적화해서 짜본 코드이다.

```

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

int main() {
    int M, n;

    scanf("%d%d", &M, &n);
    getchar();

    char tmp[21];

    char **ch = (char**)malloc(sizeof(char*) * M);
    for (int i = 0; i < M; i++) {

```

```

scanf("%s", tmp);
getchar();
ch[i] = (char*)malloc(sizeof(char) * strlen(tmp));
strcpy(ch[i], tmp); //strcpy안쓰고 =tmp로 해버리면 tmp의 주소가 ch[i]에 연결되서 배열이 꼬임
}

for (int i = 0; i < n; i++) {
    scanf("%s", tmp);
    getchar();
    if (tmp[0] >= '0' && tmp[0] <= '9') { //맨처음자리가 숫자면
        int x = atoi(tmp);
        printf("%s\n", ch[x - 1]);
    }
    else {
        for (int j = 0; j < M; j++) {
            if (strcmp(tmp, ch[j]) == 0) { //strcmp가 문자열 전체를 비교하는 건가?
                printf("%d\n", j + 1);
                break;
            }
        }
    }
}

return 0;
}

```

결과 : 시간초과

사실 문자로 입력이 들어올 경우 구조체로 정의하고 구조체 정렬을 사용해 정렬한 뒤 이진탐색을 걸면 아마 해결 될 것 같다.(구조체 배열이라 연결리스트 사용 X, 그냥 배열로 접근 가능할듯)

-> 추가적인 자료를 찾아봐야 할것 같다.



천승원

뭐든지 한걸음씩



이전 포스트

2023 Algorithm Study Week 5 - 과제

0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by
Stellate