



# 2023\_Autumn\_Algorithm\_Study\_7주차 과제 기록

[통계](#) [수정](#) [삭제](#)

jaesung2da · 19분 전 · 비공개

0

7주차 과제는

1620번: 나는야 포켓몬 마스터 이다솜

탐색트리를 이용한 문제

## 문제 설명

## 예제 입력 1 복사

```

26 5
Bulbasaur
Ivysaur
Venusaur
Charmander
Charmeleon
Charizard
Squirtle
Wartortle
Blastoise
Caterpie
Metapod
Butterfree
Weedle
Kakuna
Beedrill
Pidgey
Pidgeotto
Pidgeot
Rattata
Raticate
Spearow
Fearow
Ekans
Arbok
Pikachu
Raichu
25
Raichu
3
Pidgey
Kakuna

```

## 예제 출력 1 복사

```

Pikachu
26
Venusaur
16
14

```

요즘 들어 긴 글을 못 읽겠다 문제 서론이 길어서 입력 출력만 봤는데 푸는데 문제는 없어 보인다  
포켓몬 이름을 입력해놓고 나중에 포켓몬 이름을 입력하면 순번을 순번을 입력하면 포켓몬 이름을  
출력하면 되는 것 같다. 포켓몬 도감?을 만들고 찾으려면 되는 느낌이어서 쉬워보인다 일단은. 일단은...

## 탐색트리

그래도 문제가 탐색트리와 관련이 되어있으니 탐색트리에 대해 가볍게 복습을 하고 가면 좋을 것 같다.

탐색트리는 우선 트리 형태를 가지고 있고 값이 들어오면 Root Node에서부터 값 비교를  
시작해서 작으면 왼쪽 자식노드로 크면 오른쪽 자식노드로 이동해서 값을 비교한 뒤에 제  
위치를 찾아간다.

insert 메서드 : node 생성 기능

search 메서드 : node 조회 기능

delete 메서드 : node 삭제 기능

기본 적인 메서드는 이렇게 3개가 있고 본격적으로 문제를 풀면서 메서드 코드에 대해서는 학교  
알고리즘 강의 교안을 참고해야 할 것 같다.

## 문제 풀기 전 전략

문제에서 처음에 입력할 포켓몬의 개수 N을 알려준다 -> 노드의 총 개수

풀어야 할 문제 수 M -> 입력 개수

노드의 개수만큼 노드 생성 메서드를 통해 포켓몬을 입력해주고 모든 입력이 끝났을 때 부터 숫자가 입력되면 포켓몬 이름을 출력하고 포켓몬 이름이 입력되면 탐색후 숫자를 출력하게끔 해보야겠다.

## 문제를 풀면서 오류

우리가 알고리즘 수업시간에 푼 문제들의 경우에는 숫자를 무작위로 입력하여 트리를 생성하고 탐색하곤 했는데 이번의 경우에는 포켓몬의 이름을 순서대로 입력받고 사실상 나의 입력받은 순서대로 트리가 만들어지다 보니 그냥 단순하게 N이 26이면 이름을 26개 입력받고 순서대로 트리에 저장을 했다. 그리고 나중에 명령을 받을 때 숫자 명령과 이름명령을 구분하여 해당 노드의 저장된 값을 출력하는 형태로 문제를 풀어보았다.

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

struct Node {
    char name[21];
    int number;
    struct Node* left;
    struct Node* right;
};

struct Node* insert(struct Node* node, char* name, int number) {
    if (node == NULL) {
        struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
        strcpy(newNode->name, name);
        newNode->number = number;
        newNode->left = NULL;
        newNode->right = NULL;
        return newNode;
    }

    if (strcmp(name, node->name) < 0) {
        node->left = insert(node->left, name, number);
    }
    else if (strcmp(name, node->name) > 0) {
        node->right = insert(node->right, name, number);
    }

    return node;
}

struct Node* searchByName(struct Node* node, char* name) {
```

```
if (node == NULL || strcmp(name, node->name) == 0) {
    return node;
}

if (strcmp(name, node->name) < 0) {
    return searchByName(node->left, name);
}
else {
    return searchByName(node->right, name);
}
}

struct Node* searchByNumber(struct Node* node, int number) {
    if (node == NULL || node->number == number) {
        return node;
    }

    if (number < node->number) {
        return searchByNumber(node->left, number);
    }
    else {
        return searchByNumber(node->right, number);
    }
}

void freeTree(struct Node* node) {
    if (node == NULL) {
        return;
    }

    freeTree(node->left);
    freeTree(node->right);
    free(node);
}

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

    struct Node* root = NULL;

    for (i = 0; i < N; i++) {
        char name[21];
        scanf("%s", name);
        root = insert(root, name, i + 1);
    }

    for (i = 0; i < M; i++) {
        char name[21];
        scanf("%s", name);
        if (name[0] >= '0' && name[0] <= '9') {
            int number = atoi(name);
            struct Node* result = searchByNumber(root, number);
            if (result != NULL) {
                printf("%s\n", result->name);
            }
        }
        else {
            struct Node* result = searchByName(root, name);
            if (result != NULL) {
                printf("%d\n", result->number);
            }
        }
    }
}
```

```
        }  
    }  
  
    freeTree(root);  
  
    getchar();  
  
    return 0;  
}
```

```
10 5  
a1  
a2  
a3  
a4  
b1  
v2  
b3  
c1  
c2  
c3  
5  
b1  
3  
a3  
b3  
7  
a2  
2  
4  
a4  
계속하려면 아무 키나 누르십시오 . . . |
```

```
Squirtle
Wartortle
Blastoise
Caterpie
Metapod
Butterfree
Weedle
Kakuna
Beedrill
Pidgey
Pidgeotto
Pidgeot
Rattata
Raticate
Spearow
Fearow
Ekans
Arbok
Pikachu
Raichu
25
Raichu
26
3
Venusaur
Pidgey
16
Kakuna
14
계속하려면 아무 키나 누르십시오 . . .
```

의도한대로 입출력이 되었고 문제를 푼 방식이 출제의도대로 풀지는 않은 것 같아서 시간초과를 예상했으나

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
68932586	tlawotjd	1620	시간 초과			C99 / 수정	2085 B	17분 전
68932349	tlawotjd	1620	틀렸습니다			C99 / 수정	1955 B	22분 전
68931857	tlawotjd	1620	틀렸습니다			C99 / 수정	2012 B	31분 전
68931729	tlawotjd	1620	틀렸습니다			C99 / 수정	2012 B	34분 전
68929995	tlawotjd	1620	시간 초과			C99 / 수정	2556 B	1시간 전

그냥 틀렸다.

내일 학교 공강시간에 다시 풀어보도록 하겠다..

### 잘못된 이유가 무엇일까?

문제를 보면 대소문자에 대한 구분에 대한 언급이 있더라 근데 내가 폰 방식대로라면 그건 굳이 언급 대상이 아니라는 생각이 들고 문자열을 비교해서 탐색한다고 하면 그것 역시 굳이 순서대로 도감을 만들면 될 것 같은데 이름순으로 할 이유가 있을까 싶다...그렇게 된다면 index번호와 순번이 동시에 저장되기 때문에 더 비효율적이라고 생각하는데,,



심재성



이전 포스트

2023\_Autumn\_Algorithm\_Study\_5주차 과제 기록

### 0개의 댓글

댓글을 작성하세요

댓글 작성

