



[백준] 1620번: 나는야 포켓몬 마스터 이다솜

통계 수정 삭제

yooool · 방금 전

❤ 0

2023_Autumn_Algorithm_Study



▼ 목록 보기

0/5



1620번: 나는야 포켓몬 마스터 이다솜

N: 포켓몬의 개수, M: 문제의 개수 ($1 \leq N, M \leq 100000$)

입력:

N개의 줄에 포켓몬의 번호가 1번인 포켓몬부터 N번에 해당하는 포켓몬까지 한 줄에 하나씩

입력

포켓몬의 이름은 모두 영어로 이루어짐, 최소 길이 2, 최대 길이 20

출력:

총 M개의 줄에 내가 맞춰야하는 문제가 입력

문자 입력 → 번호 출력, 번호 입력 → 문자 출력

탐색트리와 관련된 문제여서 이진 탐색 트리를 이용해 문제 푸는 것을 시도했다.

- 이진 탐색 트리(BST): 내부노드에 (키, 원소) 쌍을 저장하는 이진트리
 - u, v, w는 모두 트리 노드이며 u와 w가 각각 v의 왼쪽과 오른쪽 부트리에 존재할 때 $key(u) < key(v) \leq key(w)$ 가 성립
 - 적정 이진 트리로 구현
 - 중위 순회하는 경우, 키가 증가하는 순서로 방문

알고리즘 수업 시간에 작성했던 코드를 수정했다.

그런데 문자를 입력, 번호 출력을 할 때 자꾸 external이 출력되는 문제가 발생했다.

뭐가 문제인지는 찾지 못했다...

3번째 문자까지는 잘 출력되는데 4번째부터 오류가 발생한다.

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define true 1
#define false 0

typedef struct Node {
    int num;
    char name[20];
    struct Node* parent;
    struct Node* lChild;
    struct Node* rChild;
} Node;

Node* root;

int isInternal(Node* w);
int isExternal(Node* w);
void insertItem(int num, char name[]);
Node* treeSearch(Node* w, int num);
Node* treeSearchName(Node* w, char name[]);
Node* expandExternal(Node* z);
void findName(char input[]);
void findNum(char input[]);

int main() {

    char name[20] = { '\0' }, input[20] = { '\0' };
    int N, M, i;

    root = (Node*)malloc(sizeof(Node));
    root->parent = NULL;
    root->lChild = NULL;
    root->rChild = NULL;

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

    for (i = 0; i < N; i++) {
        scanf("%19s", name);
        insertItem(i + 1, name);
    }

    for (i = 0; i < M; i++) {
        scanf("%s", input);
        if ('0' <= input[0] && input[0] <= '9')
            findName(input);
        else
            findNum(input);
    }

    free(root);

    return 0;

}

int isInternal(Node* w) {
    if (w->lChild != NULL || w->rChild != NULL)
        return true;

```

```
    else
        return false;
}

int isExternal(Node* w) {
    if (w->lChild == NULL && w->rChild == NULL)
        return true;
    else
        return false;
}

void insertItem(int num, char name[]) {

    Node* w = treeSearch(root, num);

    if (isInternal(w))
        return;
    else {
        expandExternal(w);
        w->num = num;
        strcpy(w->name, name);
        return;
    }
}

Node* treeSearch(Node* w, int num) {
    if (isExternal(w))
        return w;
    if (num == w->num)
        return w;
    else if (num < w->num)
        return treeSearch(w->lChild, num);
    else
        return treeSearch(w->rChild, num);
}

Node* treeSearchName(Node* w, char name[]) {

    if (isExternal(w))
        return w;

    int cmp = strcmp(name, w->name);
    if (cmp == 0)
        return w;
    else if (cmp < 0)
        return treeSearchName(w->lChild, name);
    else
        return treeSearchName(w->rChild, name);
}

Node* expandExternal(Node* z) {

    Node* L = (Node*)malloc(sizeof(Node));
    Node* R = (Node*)malloc(sizeof(Node));

    L->lChild = NULL;
    L->rChild = NULL;
    L->parent = z;

    R->lChild = NULL;
    R->rChild = NULL;
```

```
R->parent = z;

z->lChild = L;
z->rChild = R;

return z;
}

void findName(char input[]) {

    int num = strtol(input, NULL, 10);

    Node* w = treeSearch(root, num);
    if (isExternal(w))
        printf("external\n");
    else
        printf("%s\n", w->name);
}

void findNum(char input[]) {

    Node* w = treeSearchName(root, input);
    if (isExternal(w))
        printf("external\n");
    else
        printf("%d\n", w->num);
}
```



김지울



이전 포스트

[백준] 1256번: 사전

0개의 댓글

댓글을 작성하세요

댓글 작성