

Week8_과제 (1920)

[통계](#) [수정](#) [삭제](#)

yuyu7123 · 방금 전 · 비공개

 0

baekjoon

Smarcle_23Algorithm



▼ 목록 보기

8/8



Baekjoon_1920 : 수 찾기

해시테이블로 구현

풀이 과정

1. N입력받기
2. 해시테이블을 배열로 구현하여 N개의 수 배열에 저장
3. M입력받기
4. 해시테이블에서 M개의 수 찾기

문제 해결

- 배열로 구현하였는데 Segfault 오류 발생
메모리 초과문제인 것 같아 구조체로 구현
- 구조체 구현에서도 Segfault 오류 발생
int배열 대신 <stdbool.h> 라이브러리의 bool배열로 변경(각 요소가 1비트만 사용)
해시함수를 N으로 나누는 것이 아닌 100,000보다 큰 소수인 100,019를 이용하여
저장되는 값 = 각 인덱스로 두고 값이 있다면 1, 없다면 0으로 나타내도록 함 -> Segfault
오류

- bool 배열만 사용

해시테이블을 버리고 간단하게 배열로만 구현했는데 Segfault 오류 $\pi.\pi$

코드

int 배열 -> Segfault 오류

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

int h(int x, int N) {
    return x % N;
}

void insertItem(int* A, int k, int N) {
    int v = h(k, N);
    while (A[v] != 0) {
        v = (v + 1) % N;
    }
    A[v] = k;
}

int findElement(int* A, int k, int N) {
    int v = h(k, N);
    int original_v = v;
    while (A[v] != 0) {
        if (A[v] == k) {
            return 1;
        }
        v = (v + 1) % N;
        if (v == original_v) {
            break;
        }
    }
    return 0;
}

int main() {

    int N, n, M;
    scanf("%d", &N);
    int* A = (int*)malloc((N + 1) * sizeof(int));
    for (int i = 0; i <= N; i++)
        A[i] = 0;

    for (int i = 0; i < N; i++) {
        scanf("%d", &n);
        insertItem(A, n, N);
    }

    scanf("%d", &M);
    for (int i = 0; i < M; i++) {
        scanf("%d", &n);
```

```

        printf("%d\n", findElement(A, n, N));
    }

    free(A);

    return 0;
}

```

구조체 -> Segfault 오류

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

struct node {
    int n;
    struct node* next;
};

struct node** hashTable;

int h(int x, int N) {
    return x % N;
}

void initBucketArray(int N) {
    hashTable = (struct node**)malloc(N * sizeof(struct node*));
    for (int i = 0; i < N; i++)
        hashTable[i] = NULL;
}

void insertItem(int k, int N) {
    int v = h(k, N);
    struct node* newnode = (struct node*)malloc(sizeof(struct node));
    newnode->n = k;
    newnode->next = hashTable[v];
    hashTable[v] = newnode;
}

int findElement(int k, int N) {
    int v = h(k, N);
    struct node* p = hashTable[v];
    while (p) {
        if (p->n == k) return 1;
        p = p->next;
    }
    return 0;
}

int main() {

    int N, n, M;
    scanf("%d", &N);
    initBucketArray(N + 1);

    for (int i = 0; i < N; i++) {

```

```

        scanf("%d", &n);
        insertItem(n, N);
    }

    scanf("%d", &M);
    for (int i = 0; i < M; i++) {
        scanf("%d", &n);
        printf("%d\n", findElement(n, N));
    }

    for (int i = 0; i < N; i++) {
        struct node* p = hashTable[i];
        while (p) {
            struct node* temp = p;
            p = p->next;
            free(temp);
        }
    }
    free(hashTable);

    return 0;
}

```

bool 배열 -> Segfault 오류

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

int main() {

    int N, n, M;
    scanf("%d", &N);
    bool* A = (bool*)calloc(N + 1, sizeof(bool));

    for (int i = 0; i <= N; i++)
        A[i] = false;

    for (int i = 0; i < N; i++) {
        scanf("%d", &n);
        A[n] = true;
    }

    scanf("%d", &M);
    for (int i = 0; i < M; i++) {
        scanf("%d", &n);
        printf("%d\n", A[n]);
    }

    free(A);

    return 0;
}

```



유정



이전 포스트

Week7_과제 (1620)

0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by
Stellate