



# 8주차 해시테이블

[통계](#) [수정](#) [삭제](#)

iming03 · 방금 전 · 비공개

0

[분리연쇄법](#)[해시테이블](#)

## 알고리즘 스터디

[▼ 목록 보기](#)

5/5



### 예제 입력 1 복사

```
5
4 1 5 2 3
5
1 3 7 9 5
```

### 예제 출력 1 복사

```
1
1
0
0
1
```

# 백준 1920 : 수찾기

## 문제

N개의 정수  $A[1], A[2], \dots, A[N]$ 이 주어져 있을 때, 이 안에 X라는 정수가 존재하는지 알아내는 프로그램을 작성하시오.

## 입력

첫째 줄에 자연수  $N(1 \leq N \leq 100,000)$ 이 주어진다. 다음 줄에는 N개의 정수  $A[1], A[2], \dots, A[N]$ 이 주어진다. 다음 줄에는  $M(1 \leq M \leq 100,000)$ 이 주어진다. 다음 줄에는 M개의 수들이 주어지는데, 이 수들이 A안에 존재하는지 알아내면 된다. 모든 정수의 범위는  $-2^{31}$ 보다 크거나 같고  $2^{31}$ 보다 작다.

## 출력

M개의 줄에 답을 출력한다. 존재하면 1을, 존재하지 않으면 0을 출력한다.

## 예제 입력 1 복사

```
5
4 1 5 2 3
5
1 3 7 9 5
```

## 예제 출력 1 복사

```
1
1
0
0
1
```

# 해시테이블: 키 값을 조작해(해시) 새로운 값을 만들고 탐색(테이블)

## 해시테이블

- 해시테이블은 사전 ADT이다.
- 해시테이블 = 버킷 배열 + 해시함수

## 버킷 배열: 크기 M의 배열 A

- 각 셀은 버킷, 또는 슬롯이라고 함
- 정수 M은 배열의 용량
- 키 k를 가진 원소 e는 버킷 A[k]에 삽입
- 속하지 않는 버킷 셀들은 NoSuchKey라는 특별한 개체를 담는 것으로 가정

## 해시함수 h: 주어진 형의 키를 고정범위 [0,M-1]로 매핑

- $h(x) = x \% M$
- $h(x)$ : 키 x의 해시값
- h1: 해시코드맵 (keys  $\rightarrow$  integers)

- $h_2$ : 압축맵 ( $\text{integers} \rightarrow [0, M-1]$ )

## 충돌해결

- 충돌: 두 개 이상의 원소들이 동일한 셀로 매핑
- 다른 키  $k_1, k_2$ 에 대해  $h(k_1) = h(k_2)$ 이면 "충돌이 일어났다"고 말함
- 충돌 해결을 위한 일관된 전략 필요

## 분리연쇄법

- 각 버킷  $A[i]$ 는 리스트  $L_i$ 에 대한 참조를 저장 ( 연결리스트로 연결 )
- 무순/순서리스트 사용
- 장점: 단순하고 빠름
- 단점: 추가적인 저장공간 요구

## 분리연쇄법 알고리즘

<b>Alg <i>findElement</i>(<math>k</math>)</b> input bucket array $A[0..M-1]$ , hash function $h$ , key $k$ output element with key $k$  1. $v \leftarrow h(k)$ 2. return $A[v].findElement(k)$	<b>Alg <i>initBucketArray</i>()</b> input bucket array $A[0..M-1]$ output bucket array $A[0..M-1]$ initialized with null buckets  1. for $i \leftarrow 0$ to $M-1$ $A[i] \leftarrow \text{empty list}$ 2. return
<b>Alg <i>insertItem</i>(<math>k, e</math>)</b> 1. $v \leftarrow h(k)$ 2. $A[v].insertItem(k, e)$ 3. return	
<b>Alg <i>removeElement</i>(<math>k</math>)</b> 1. $v \leftarrow h(k)$ 2. return $A[v].removeElement(k)$	

## 문제해결

1. 입력받는 원소들을  $N$ 으로 나눠 테이블에 저장
  1.  $h(x) = x \% N$
2. 충돌이 일어난 경우 분리연쇄법으로 해결해보자

## 알고리즘 실습 1번문제를 활용해서 코드 변형

segfault오류

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

typedef struct node {
    int elem;
    struct node* next;
}NODE;

NODE* Table;
int N;

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

void insertItem(int x) {

    int v = h(x);
    NODE* p = Table + v;

    NODE* new = (NODE*)malloc(sizeof(NODE));
    new->elem = x;
    new->next = (Table + v)->next;
    (Table + v)->next = new;
}

int searchItem(int x) {

    int v = h(x);
    int rank = 0;
    NODE* p = Table + v;

    if (p->next == NULL) {
        return 0;
    }
    else {
        while (1) {
            p = p->next;
            rank++;

            if (p->elem == x) {
                return rank;
            }

            if (p->next == NULL) {
                return 0;
            }
        }
    }
}
```

```

void freeTable() {

    NODE* p1, * p2;

    for (int i = 0; i < N; i++) {
        if ((Table + i)->next != NULL) {
            p1 = (Table + i)->next;
            p2 = p1;

            while (p2->next != NULL) {
                p1 = p1->next;
                free(p2);
                p2 = p1;
            }
        }
    }
    free(Table);
}

int main() {
    int key, M;
    char com;
    scanf("%d", &N);

    Table = (NODE*)malloc(sizeof(NODE) * N);

    for (int i = 0; i < N; i++) {
        (Table + i)->elem = NULL;
        (Table + i)->next = NULL;
    }

    for (int i = 0; i < N; i++) {
        scanf("%d", &key);
        insertItem(key); //삽입
        getchar();
    }
    scanf("%d", &M);

    for (int i = 0; i < M; i++) {
        scanf("%d", &key);
        if (searchItem(key) == 0) {
            printf("0\n");
        }
        else
            printf("1\n");
        getchar();
    }

    freeTable();
}

```

```

5
4 1 5 2 3
5
1 3 7 9 5
1
1
0
0
1
C:\Users\sweet\Documents\SMARCLE\알고리즘 스터디\Project1\x64\Debug\Project1.exe(프로세스 48096개)이(가) 종료되었습니다(
코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

단순하게 찾았을 때 0이 아니면 1로 0이면 0으로 했는데 정답 그대로 나왔다...오호,,

출력은 제대로 나오는데 런타임에러다..

69237072	iming03	1920	<a href="#">런타임 에러 (Segfault)</a>			C99 / 수정	1359 B
----------	---------	------	-----------------------------------	--	--	----------	--------

문제가 있을 법한 걸 살짝 고쳐봤다.

```

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

typedef struct node {
    int elem;
    struct node* next;
}NODE;

NODE* Table;
int N;

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

void insertItem(int x) {

    int v = h(x);
    NODE* p = Table + v;

    NODE* new = (NODE*)malloc(sizeof(NODE));
    new->elem = x;
    new->next = (Table + v)->next;
    (Table + v)->next = new;
}

int searchItem(int x) {

    int v = h(x);
    int rank = 0;

```

```

    NODE* p = Table + v;

    if (p->next == NULL) {
        return 0;
    }
    else {
        while (1) {
            p = p->next;
            rank++;
            if (p->elem == x) {
                return rank;
            }
            if (p->next == NULL) {
                return 0;
            }
        }
    }
}

void freeTable() {

    NODE* p1, * p2;

    for (int i = 0; i < N; i++) {
        if ((Table + i)->next != NULL) {
            p1 = (Table + i)->next;
            p2 = p1;

            while (p2->next != NULL) {
                p1 = p1->next;
                free(p2);
                p2 = p1;
            }
        }
        free(Table);
    }

    int main() {
        int key, M;
        char com;
        scanf("%d", &N);

        Table = (NODE*)malloc(sizeof(NODE) * N);
        for (int i = 0; i < N; i++) {
            (Table + i)->elem = 0;
            (Table + i)->next = NULL;
        }

        for (int i = 0; i < N; i++) {
            scanf("%d", &key);
            insertItem(key); //삽입
        }
        scanf("%d", &M);

        for (int i = 0; i < M; i++) {
            scanf("%d", &key);
            if (searchItem(key) == 0) {

```



```

        printf("0\n");
    }
    else
        printf("1\n");
}

freeTable();

}

```

그래도...

69237367	iming03	1920	<a href="#">런타임 에러 (Segfault)</a>			C99 / 수정	1325 B
----------	---------	------	-----------------------------------	--	--	----------	--------

다시 한 번 고쳐봤다.

```

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

typedef struct node {
    int elem;
    struct node* next;
}NODE;

int N;
NODE* Table;

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

void insertItem(int x) {

    int v = h(x);
    NODE* p = Table + v;

    NODE* new = (NODE*)malloc(sizeof(NODE));
    new->elem = x;
    new->next = p->next;
    p->next = new;
}

int searchItem(int x) {

    int v = h(x);
    int rank = 0;
    NODE* p = Table + v;

    if (p->next == NULL) {
        return 0;
    }
}

```

```

    }
    else {
        while (1) {
            p = p->next;
            rank++;
            if (p->elem == x) {
                return rank;
            }
            if (p->next == NULL) {
                return 0;
            }
        }
    }
}

void freeTable() {

    NODE* p1, * p2;

    for (int i = 0; i < N; i++) {
        if ((Table + i)->next != NULL) {
            p1 = (Table + i)->next;
            p2 = p1;

            while (p2->next != NULL) {
                p1 = p1->next;
                free(p2);
                p2 = p1;
            }
        }
    }
    free(Table);
}

int main() {
    int key, M;
    char com;
    scanf("%d", &N);

    Table = (NODE*)malloc(sizeof(NODE) * N);
    for (int i = 0; i < N; i++) {
        (Table + i)->elem = 0;
        (Table + i)->next = NULL;
    }

    for (int i = 0; i < N; i++) {
        scanf("%d", &key);
        insertItem(key); //삽입
    }
    scanf("%d", &M);

    for (int i = 0; i < M; i++) {
        scanf("%d", &key);
        if (searchItem(key) == 0)
            printf("0\n");
        else
            printf("1\n");
    }
}

```

```
freeTable();  
  
return 0;  
}
```

69237690	iming03	1920	런타임 에러 (Segfault)			C99 / 수정	1310 B
----------	---------	------	-------------------	--	--	----------	--------

ㅏㅏ..



**강민돌**

민돌이의 공부



이전 포스트  
7주차. 탐색트리

**0개의 댓글**

댓글을 작성하세요

댓글 작성

