

# 제 11 장 해싱

## 1. 해싱(hashing)이란?

키 값의 계수적인 성질을 이용하여 저장주소를 구한 후 자료를 저장(삽입)하고 같은 방법으로 자료 검색, 제거 등을 수행하는 방법이다.

### (1) 기본용어

- ① 해시함수 : 키에 대한 저장주소를 구할 때 적용되는 함수
- ② 홈주소 : 해시함수에 의해 계산된 번지로 자료가 저장되는 주소(버킷주소)
- ③ 해시표(hash table)

해시함수에 의해 산출된 함수값(홈주소)의 위치에 해당 자료를 기억시킨 표이다. 해시표는 버킷(bucket)과 슬롯(slot)으로 구성된다.

[예] 버킷 크기는 7, 슬롯은 2인 배열구조로 정수 k에 대한 해시표를 만들어 보자.  
(단, 정수 k에 대한 해시함수는  $f(k) = k \% 7$ 로 정의한다)

k	함수값	버킷주소	슬롯 0	슬롯 1
30	$30 \% 7 = 2$	0	14	
14	$14 \% 7 = 0$	1	29	
23	$23 \% 7 = 2$	2	30	23
48	$48 \% 7 = 6$	3		
12	$12 \% 7 = 5$	4	25	
29	$29 \% 7 = 1$	5	12	
25	$25 \% 7 = 4$	6	48	

④ 버킷 : 자료를 저장할 기억장소로 1개 이상의 슬롯으로 구성된다.

⑤ 충돌(collision)

서로 다른 심벌(키)이 동일한 버킷주소를 가지는 경우이다.

충돌이 발생하면 자료를 빈 슬롯에 차례로 저장한다.

빈 슬롯이 더 이상 없으면 즉, 해당 주소 버킷이 가득 찼을 때 'overflow가 발생했다'고 한다.

2 <http://cafe.daum.net/pass365>(홍재연)

⑥ 동의어(synonym, 동거자)

서로 다른 심볼이 동일한 버킷주소를 가질 때 두 심볼을 동의어라 한다.

(앞의 예에서 키 30과 23은 동의어 관계이다)

⑦ 해시표의 적재밀도(loading density) 또는 적재인수(loading factor)는 다음과 같다.

$$\cdot \text{적재밀도 } \alpha = \frac{n}{sb} = \frac{7}{2 \times 7} = 0.5$$

n : 해시표에 저장된 키의 수

s : 슬롯 크기

b : 버킷 크기

(2) 포트란에서 변수명을 해시표로 구현

초기 포트란 언어에서 변수명은 영문자로 시작하고, 숫자 조합, 최대 6자이다.

① 포트란에서 구성 가능한 변수 개수 : T

$$T = \sum_{i=0}^5 26 \times 36^i = 26(36^0 + 36^1 + \dots + 36^5) = 1.6 \times 10^9 (\text{약 } 10^9)$$

② 어떤 프로그램에서 n개의 변수를 사용했다면

변수밀도(variable density) =  $\frac{n}{T}$  → 이 값은 보통 매우 작은 값이 된다.(해시표 사용 이유)

[예] 포트란 컴파일러가 변수의 심볼 테이블을 해시표로 구현하였다. 포트란에서 가능한 변수의 개수가 T이고, 심볼 테이블의 기억공간 개수가 S일 때 다음 포트란 프로그램에 대한 변수밀도와 적재인수는?

<pre>A = 0.5 B = 7.2 10 A = A + 1 B = B + A IF (A .LT. 10) GO TO 10 WRITE(*,*) B END</pre>	<p>(풀이) 사용된 변수는 2개(A와 B)이므로</p> $\text{변수밀도} = \frac{2}{T}$ $\text{적재인수} = \frac{2}{S}$
--	---

**기출문제 분석**

1. 해싱(hashing)에 관한 다음 설명 중 옳은 것만으로 묶은 것은? [2009년 국가 7급]

- ㉠ n개의 자료가 있을 때 자료의 탐색에 걸리는 시간은  $O(\log n)$ 이다.
- ㉡ 해시함수는 서로 다른 자료는 항상 서로 다른 버킷(bucket)에 사상(mapping)시킨다.
- ㉢ 탐색 성능은 적재밀도(load density)가 높을수록 좋아진다.
- ㉣ 최대값을 갖는 자료를 찾는데 걸리는 시간은  $O(n)$ 이다.
- ㉤ 해시함수 값은 충돌이 적어야 하고, 해시테이블 주소에 고르게 분포하는 것이 좋다.

- ① ㉠㉡                      ② ㉠㉢                      ③ ㉠㉣                      ④ ㉢㉤

☞ 해싱

- ㉠ n개의 자료가 있을 때 자료의 탐색에 걸리는 시간은  $O(\log n)$ 이다.(x)  
→  $O(1)$ 이다. 단, 최악의 경우는  $O(n)$ 이 된다.
- ㉡ 해시함수는 서로 다른 자료는 항상 서로 다른 버킷(bucket)에 사상(mapping)시킨다.(x)  
→ 하나의 버킷이 여러 개의 슬롯으로 구성되어 있으면, 동의어는 같은 버킷에 저장된다.  
→ 여기서, 동의어는 서로 다른 자료이면서 같은 버킷 주소를 가지는 것이다.
- ㉢ 탐색 성능은 적재밀도(load density)가 높을수록 좋아진다.(x)  
→ 적재밀도가 높을수록 많은 자료가 저장되어 있으므로 탐색 성능은 조금 떨어질 수 있다.  
→ 여기서, 조금 떨어진다는 것은 충돌이 발생한 것이 존재하기 때문이다.

정답 : ④

2. 키(key) 값의 순서(ordering)와 가장 관련이 없는 것은? [2021년 군무원 7급]

- ① 이진탐색                      ② 선형보간탐색
- ③ 해싱                              ④ AVL 트리

☞ 키(key) 값의 순서

- ① 이진탐색 → 정렬된 리스트에서만 수행할 수 있다.
- ② 선형보간탐색 → 정렬된 리스트에서만 수행할 수 있다.
- ③ 해싱 → 키(key) 값의 순서와 무관
- ④ AVL 트리 → 부모를 기준으로 작은 값은 좌측, 큰 값은 우측에 배치한다.

정답 : ③

### 3. 해싱(hashing)에 대한 설명으로 옳지 않은 것은? [2018년 컴일 국가 9급]

- ① 서로 다른 탐색키가 해시함수를 통해 동일한 해시 주소로 사상될 수 있다.
- ② 충돌(collision)이 발생하지 않는 해시함수를 사용한다면 해싱의 탐색 시간복잡도는  $O(1)$ 이다.
- ③ 선형조사법(linear probing)은 연결리스트(linked list)를 사용하여 오버플로 문제를 해결한다.
- ④ 폴딩함수(folding function)는 탐색키를 여러 부분으로 나누어 이들을 더하거나 배타적 논리합을 하여 해시 주소를 얻는다.

#### ☞ 해싱(hashing)

---

- 선형조사법(linear probing)은 **연결리스트**를 사용하여 오버플로 문제를 해결한다.(×)
    - 선형조사법은 **배열**을 사용하여 오버플로 문제를 해결한다.
    - 연결리스트를 사용하여 오버플로를 해결하는 것은 체이닝(폐쇄주소법)이다.
- 

정답 : ③

### 4. 해싱(hashing)에 대한 설명으로 옳지 않은 것은? [2015년 서울 9급]

- ① 검색 속도가 빠르며 삽입, 삭제의 빈도가 높을 때 유리한 방식이다.
- ② 해싱기법에는 숫자 분석법(digit analysis), 제산법(division), 제곱법(mid-square), 접지법(folding) 등이 있다.
- ③ 충돌 시 오버플로(overflow) 해결의 부담이 과중되나, 충돌 해결에 필요한 기억공간이 필요하지는 않다.
- ④ 오버플로(overflow)가 발생했을 때 해결기법으로 개방주소법(open addressing)과 폐쇄주소법(close addressing)이 있다.

#### ☞ 해싱(hashing)에서 충돌 해결

---

- 충돌 시 충돌 해결에 필요한 기억공간이 필요하지는 않다.(×)
    - 충돌 해결을 위해 추가적인 기억공간 등이 필요하다.
- 

정답 : ③

5. 해시(hash)에 대한 설명으로 옳지 않은 것은? [2021년 국가 9급]

- ① 연결리스트는 체이닝(chaining) 구현에 적합하다.
- ② 충돌이 전혀 없다면 해시 탐색의 시간복잡도는  $O(1)$ 이다.
- ③ 최악의 경우에도 이진탐색보다 빠른 성능을 보인다.
- ④ 해시함수는 임의의 길이의 데이터를 입력받을 수 있다.

↳ 해시함수

---

- 최악의 경우에도 이진탐색보다 빠른 성능을 보인다.(x)  
→ 해시함수는 최악의 경우 시간복잡도는  $O(n)$ 이다. (연속적인 충돌이 발생하는 경우)
  - 해시함수의 기본 목적은 시간복잡도가  $O(1)$ 이다. (충돌이 없다면)
  - 이진탐색의 시간복잡도는  $O(\log n)$ 이다.
- 

정답 : ③