

Ch.7 해시 테이블(hashing)

저장/검색의 복잡도

■ 배열

- $O(n)$

■ 이진 검색 트리

- 최악의 경우 $\Theta(n)$
- 평균 $\Theta(\log n)$

■ 균형 잡힌 이진 검색 트리(예: 레드 블랙 트리)

- 최악의 경우 $\Theta(\log n)$

■ B-트리

- 최악의 경우 $\Theta(\log n)$
- 균형 잡힌 이진 검색 트리보다 상수 인자가 작다

▼ ■ 해시 테이블

- 평균 $\Theta(1)$

해시 테이블

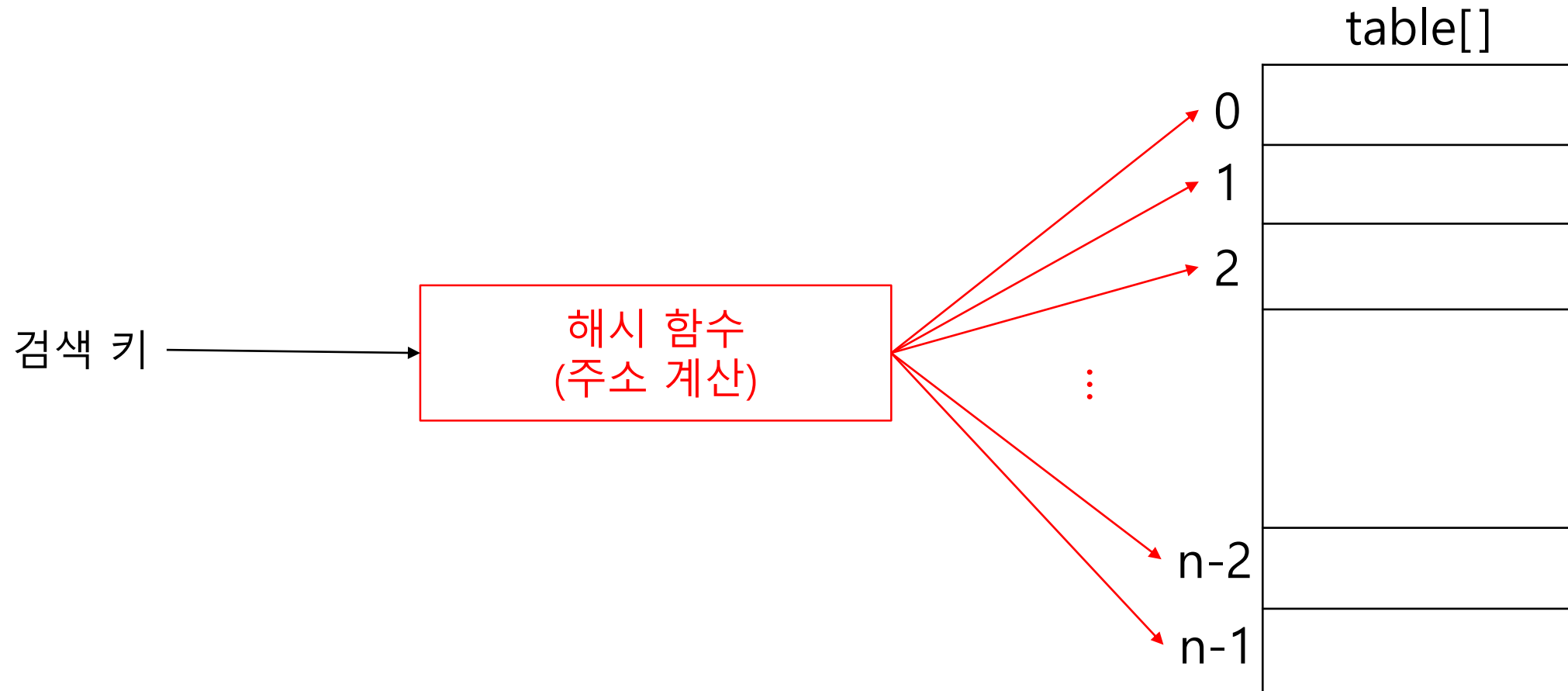


그림 7-2 해시 테이블의 고안 배경

해시 테이블

- 원소가 저장될 자리가 원소의 값에 의해 결정되는 자료구조
- 평균 상수 시간에 삽입, 삭제, 검색
- 매우 빠른 응답을 요하는 응용에 유용
 - 예) 119 긴급구조 호출과 호출번호 관련 정보 검색
 - 예) 주민등록 시스템
- 해시 테이블은 최소 원소를 찾는 것과 같은 작업은 지원하지 않는다

주소 계산



해시 테이블 예

■ 크기 13인 해시 테이블에 5 개의 원소가 저장된 예

입력: 25, 13, 16, 15, 7

0	13
1	
2	15
3	16
4	
5	
6	
7	7
8	
9	
10	
11	
12	25

그림 7-1 크기가 13인 해시 테이블에 5개의 원소가 들어간 예

해시 함수

- 입력 원소가 해시 테이블에 고루 저장되어야 한다
- 계산이 간단해야 한다
- 여러 가지 방법이 있으나 가장 대표적인 것은 나누기 방법과 곱하기 방법이다

해시 함수

■ 나누기 방법(Division Method)

- 해시 테이블 크기보다 큰 수를 해시 테이블 크기 범위에 들어오도록 수축
- $h(x) = x \bmod m$
 - m : 해시 테이블의 크기

■ 곱하기 방법(Multiplication Method)

- 입력값을 0과 1 사이의 소수로 대응시킨 다음 해시 테이블 크기 m 을 곱하여 0부터 $m-1$ 사이로 팽창
- $h(x) = (xA \bmod 1) * m$
 - A : $0 < A < 1$ 인 상수
 - m 은 굳이 소수일 필요 없어 보통 $m=2^p$ 로 잡음

곱하기 방법의 작동 과정

- x 에 A 를 곱한 다음 소수부만 취한다.
- 방금 취한 소수부에 m 을 곱하여 그 정수부를 취한다.

$$h(x) = \lfloor m(xA \bmod 1) \rfloor$$

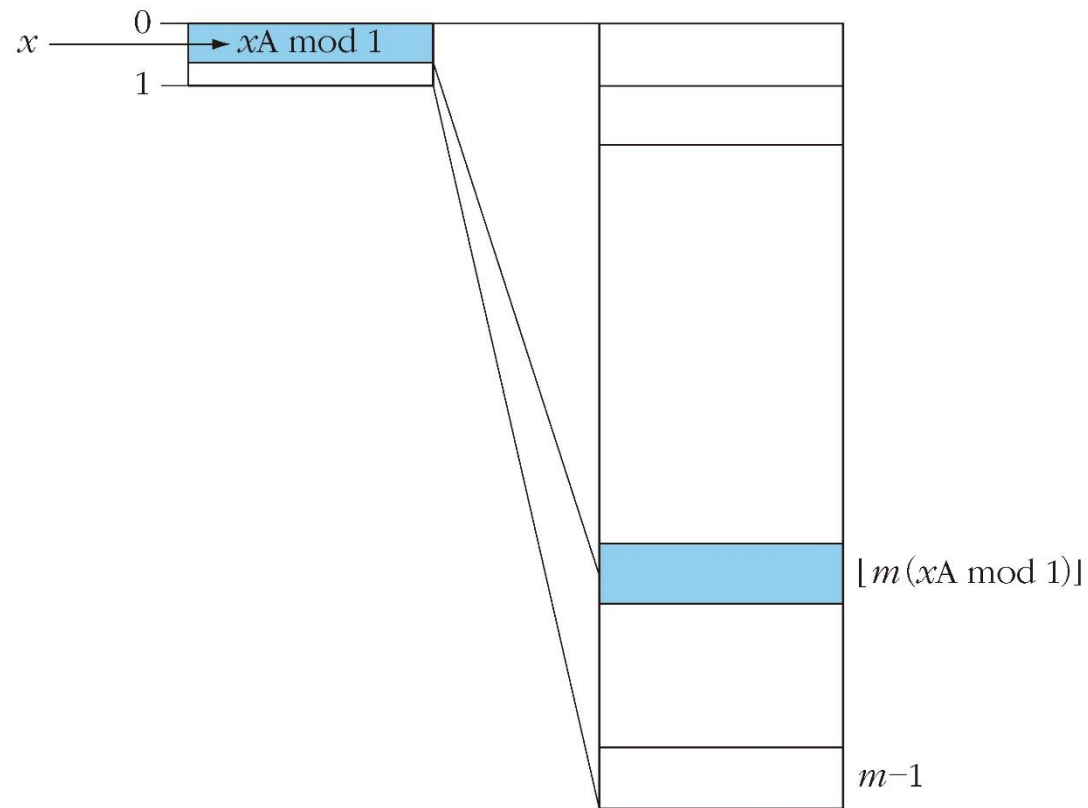


그림 7-3 곱하기 방법의 작동 과정을 보여주는 예

충돌

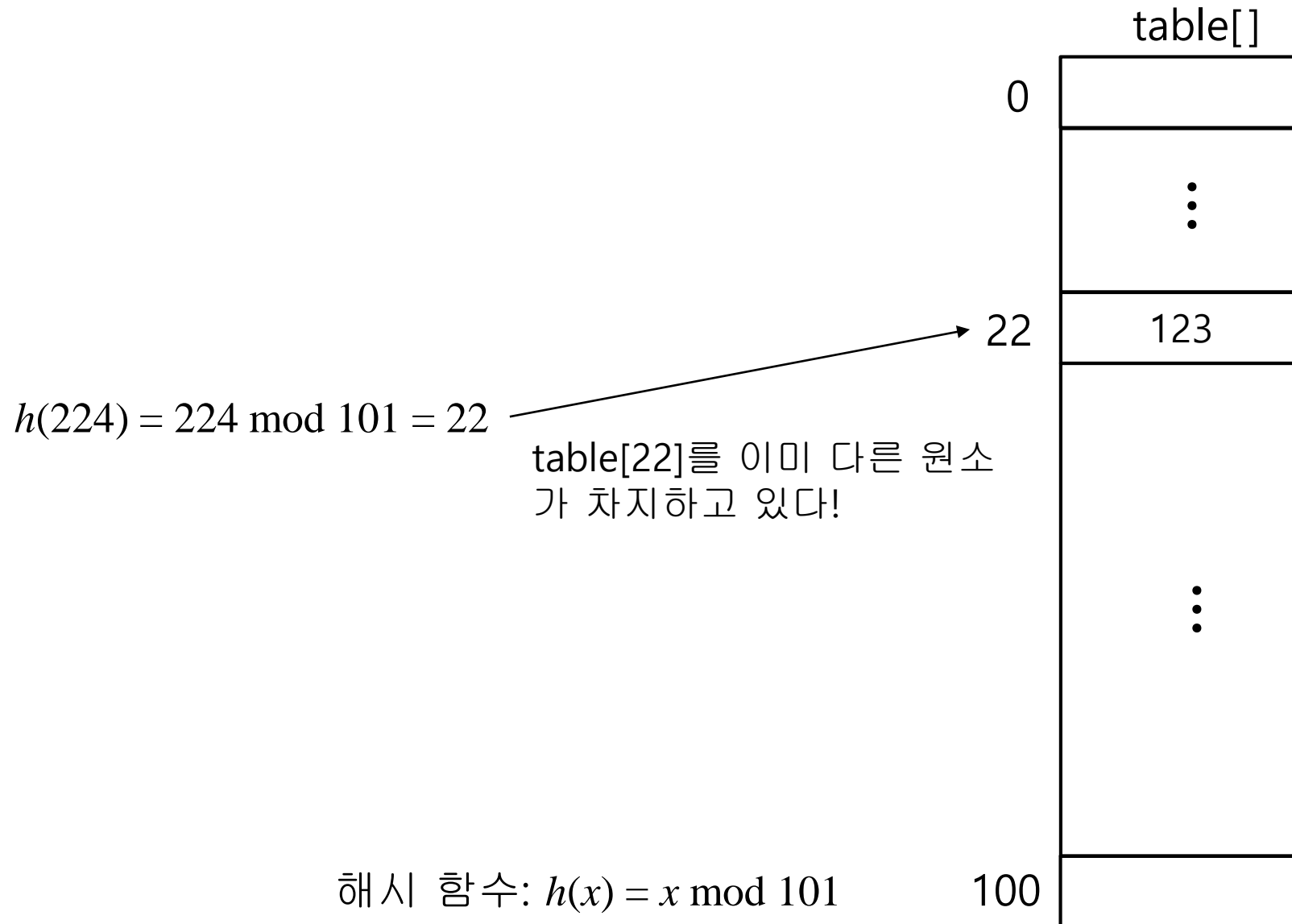
■ 충돌

- 해시 테이블의 한 주소를 놓고 두 개 이상의 원소가 자리를 다투는 것

■ 충돌 해결 방법

- 체이닝(Chaining)
 - 같은 주소로 해싱되는 원소를 모두 하나의 연결 리스트에 매달아 관리
- 개방 주소 방법(Open Addressing)
 - 빈자리가 생길 때까지 해시값을 계속 만들어 주어진 테이블 공간에서 해결
 - $h_0(x)(=h(x)), h_1(x), h_2(x), h_3(x), \dots$

충돌의 예



체이닝을 이용한 충돌 해결 예

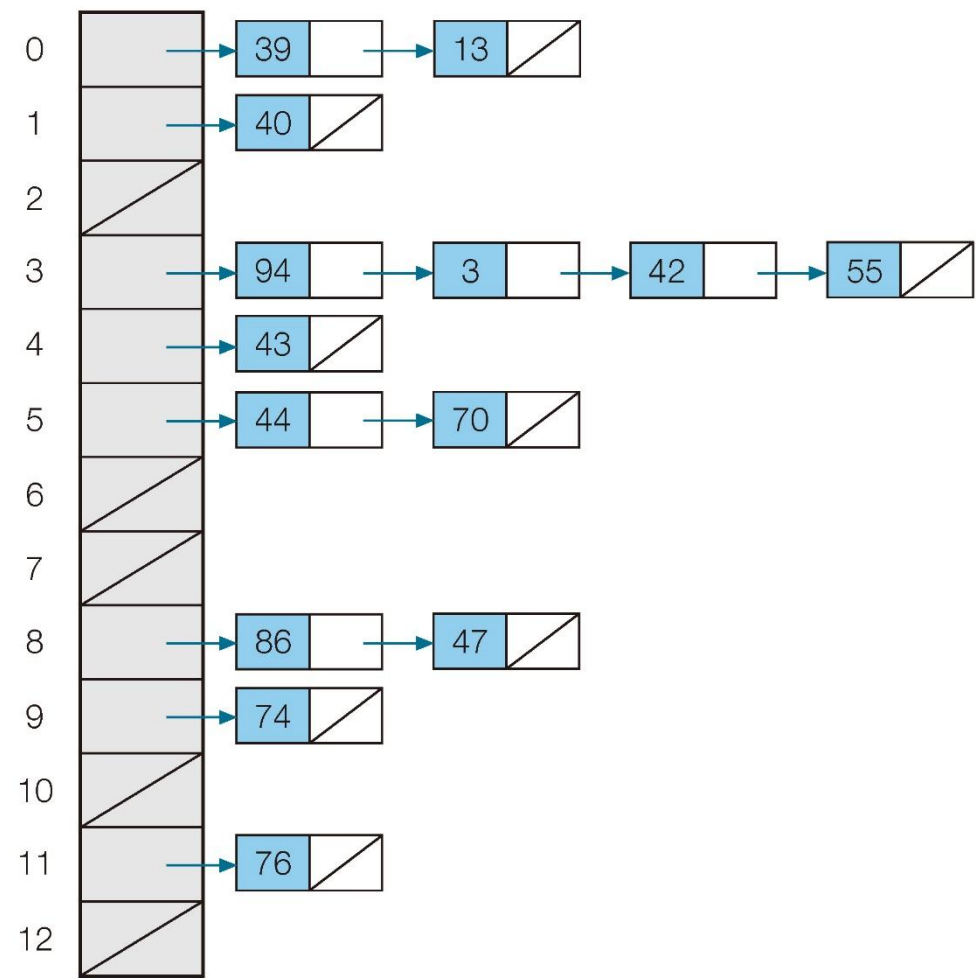


그림 7-4 체이닝을 이용한 충돌 해결을 보여주는 예

체이닝을 이용한 해시 테이블 알고리즘

알고리즘 7-1

체이닝을 사용하는 해시 테이블에서의 작업

`chainedHashInsert($T[]$, x):`

▷ T : 해시 테이블, x : 삽입 원소

리스트 $T[h(x)]$ 의 맨 앞에 x 를 삽입

`chainedHashSearch($T[]$, x):`

▷ T : 해시 테이블, x : 검색 원소

리스트 $T[h(x)]$ 에서 x 값을 가지는 원소를 검색

`chainedHashDelete($T[]$, x):`

▷ T : 해시 테이블, x : 삭제 원소

리스트 $T[h(x)]$ 에서 x 의 노드를 삭제

개방 주소 방법의 중요한 세 가지 방법

■ 선형 조사

- 충돌이 일어난 바로 뒷자리를 봄

- $h_i(x) = (h(x) + i) \% m$ $i = 0, 1, 2, \dots$

■ 이차원 조사

- 충돌이 일어난 바로 뒷자리를 보는 대신 보폭을 이차 함수로 넓혀가면서 봄

- $h_i(x) = (h(x) + c_1 i^2 + c_2 i) \% m$ $i = 0, 1, 2, \dots$

■ 더블 해싱

- 2개의 함수 사용

- $h_i(x) = (h(x) + i \cdot f(x)) \% m$ $i = 0, 1, 2, \dots$

선형 조사

- 충돌이 일어난 바로 뒷자리를 봄
(충돌이 일어난 자리에서 i 에 관한 일차 함수의 보폭으로 점프)
- $h_i(x) = (h(x) + i) \% m$ $i = 0, 1, 2...$

입력 : 25, 13, 16, 15, 7, 28, 31, 20, 1, 38

0	13
1	
2	15
3	16
4	28
5	
6	
7	7
8	
9	
10	
11	
12	25

0	13
1	
2	15
3	16
4	28
5	31
6	
7	7
8	20
9	
10	
11	
12	25

0	13
1	1
2	15
3	16
4	28
5	31
6	38
7	7
8	20
9	
10	
11	
12	25

그림 7-5 선형 조사의 예

선형 조사

- 특정 영역에 원소가 몰리면 치명적으로 성능이 떨어지는 1차 군집(Primary Clustering) 현상 발생

0	
1	
2	15
3	16
4	28
5	31
6	44
7	
8	
9	
10	
11	37
12	

그림 7-6 1차 군집의 예

이차원 조사

- 충돌이 일어난 바로 뒷자리를 보는 대신 보폭을 이차 함수로 넓혀가면서 봄
- $h_i(x) = (h(x) + c_1 i^2 + c_2 i) \% m$ $i = 0, 1, 2...$

0	
1	
2	15
3	16
4	28
5	31
6	44
7	29
8	
9	
10	
11	37
12	




그림 7-7 1차 군집을 빨리 벗어나는 예

0	
1	
2	15
3	28
4	
5	54
6	41
7	
8	21
9	
10	
11	67
12	

그림 7-8 2차 군집의 예

더블 해싱

■ 2개의 함수 사용

■ $h_i(x) = (h(x) + i \cdot f(x)) \% m$

$i = 0, 1, 2, \dots$

$h(x) = x \bmod 13$

$f(x) = 1 + (x \bmod 11)$

$h_i(x) = (h(x) + i \cdot f(x)) \bmod 13$

0	
1	
2	15
3	
4	67
5	
6	19
7	
8	
9	28
10	
11	41
12	

$h_0(15) = h_0(28) = h_0(41) = h_0(67) = 2$

$h_1(67) = 4$

$h_1(28) = 9$

$h_1(41) = 11$

그림 7-9 2차 군집에서 해방된 예

개방 주소 방법 알고리즘

알고리즘 7-2

개방 주소 방법

hashInsert($T[], x$):

$i \leftarrow 0$

repeat

$j \leftarrow h_i(x)$

if ($T[j] = \text{NIL}$ or $T[j] = \text{DELETED}$)

$T[j] \leftarrow x$; return j

else $i++$

until ($i = m$)

error “테이블 오버플로우”

hashSearch($T[], x$):

$i \leftarrow 0$

repeat

$j \leftarrow h_i(x)$

if ($T[j] = x$) return j

else $i++$

until ($T[j] = \text{NIL}$ or $i = m$)

return NIL

해시 테이블에서 자료가 삭제될 경우의 처리 방법

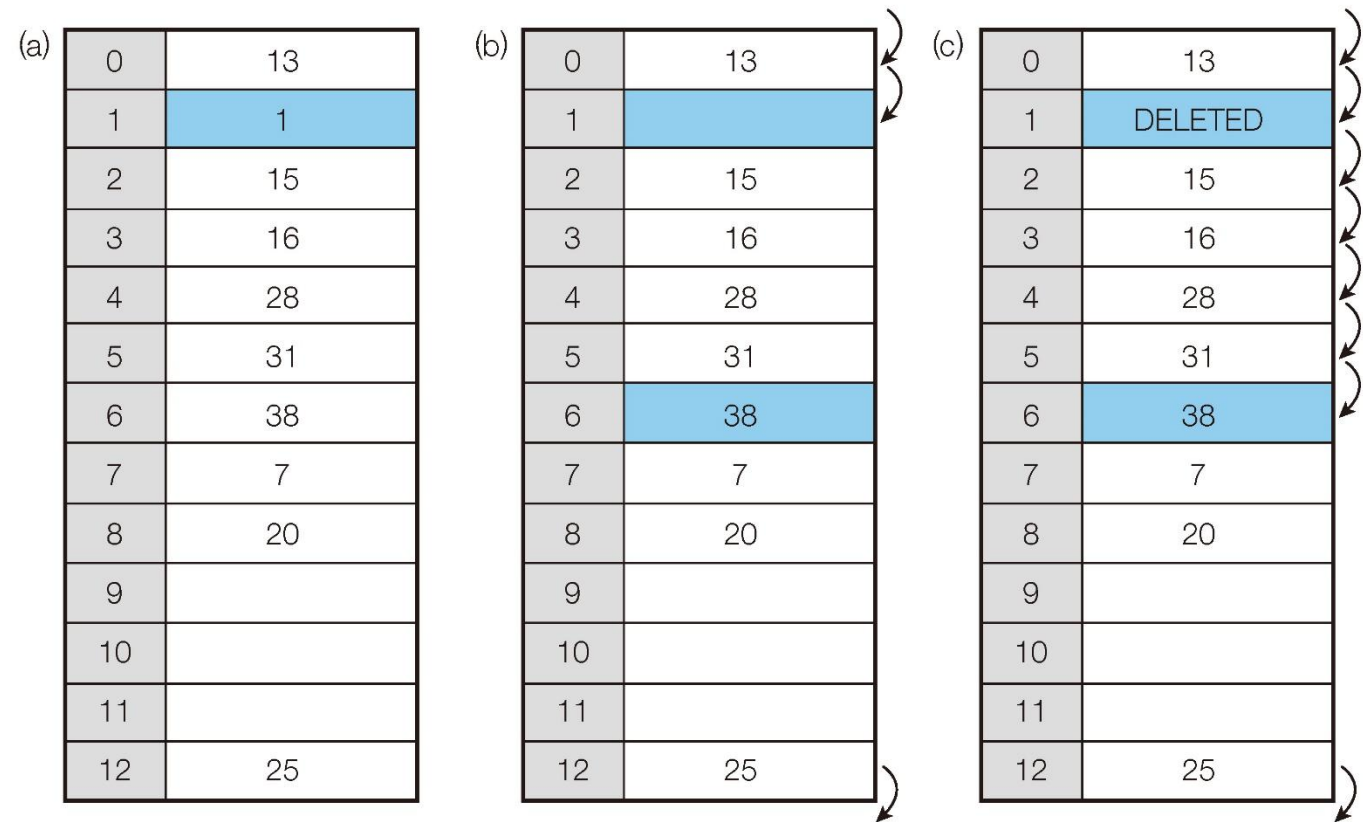


그림 7-10 해시 테이블에서 자료가 삭제될 경우의 처리 방법

해시 테이블에서의 검색 시간 분석

정리 7-1

체이닝 방법을 이용하는 해싱에서 적재율이 α 일 때, 실패하는 검색에서 조사 횟수의 기대치는 α 이다.

해시 테이블에서의 검색 시간 분석

정리 7-2

체이닝을 이용하는 해싱에서 적재율이 α 일 때, 성공하는 검색에서 조사 횟수의 기대치는 $\frac{1+\alpha}{2} + \frac{\alpha}{2n}$ 이다.

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n \left(1 + \sum_{j=i+1}^n \frac{1}{m}\right) &= 1 + \frac{1}{mn} \sum_{i=1}^n \sum_{j=i+1}^n 1 \\ &= 1 + \frac{1}{mn} \sum_{i=1}^n (n-i) \\ &= 1 + \frac{1}{mn} \left(\sum_{i=1}^n n - \sum_{i=1}^n i\right) \\ &= 1 + \frac{1}{mn} \left(n^2 - \frac{n(n+1)}{2}\right) \\ &= 1 + \frac{n-1}{2m} \\ &= 1 + \frac{\alpha}{2} - \frac{\alpha}{2n}\end{aligned}$$

해시 테이블에서의 검색 시간 분석

정리 7-3

해시 함수가 앞에서 가정한 특성을 만족한다고 할 때, 적재율 $\alpha = \frac{n}{m} < 1$ 인 개방 주소 해싱의 실패하는 검색에서 조사 횟수의 기대치는 최대 $\frac{1}{1-\alpha}$ 이다.

- p_i : 빈자리를 찾기 전에 정확히 i 번 이미 점유된 주소를 조사할 확률
- q_i : 빈자리를 찾기 전에 적어도 i 번 이미 점유된 주소를 조사할 확률

$$\begin{aligned} q_1 &= \frac{n}{m} & p_i &= q_i - q_{i+1} \\ q_2 &= \frac{n}{m} \frac{n-1}{m-1} & q_i &= \frac{n}{m} \frac{n-1}{m-1} \frac{n-2}{m-2} \cdots \frac{n-i+1}{m-i+1} \leq \left(\frac{n}{m}\right)^i = \alpha^i \\ &\dots & & \\ q_i &= \frac{n}{m} \frac{n-1}{m-1} \frac{n-2}{m-2} \cdots \frac{n-i+1}{m-i+1} \end{aligned}$$

$$\begin{aligned} 1 + \sum_{i \geq 0} i p_i &= 1 + \sum_{i \geq 1} i (q_i - q_{i+1}) \\ &= 1 + \sum_{i \geq 1} q_i \\ &\leq 1 + \sum_{i \geq 1} \alpha^i \\ &= \frac{1}{1-\alpha} \end{aligned}$$

해시 테이블에서의 검색 시간 분석

정리 7-4

해시 함수가 앞에서 가정한 특성을 만족한다고 할 때, 적재율 $\alpha = \frac{n}{m} < 1$ 인 개방 주소 해싱의 성공하는 검색에서 조사 횟수의 기대치는 최대 $\frac{1}{\alpha} \log \frac{1}{1-\alpha}$ 이다.

$$\begin{aligned} \frac{1}{n} \sum_{i=0}^{n-1} \frac{m}{m-i} &= \frac{m}{n} \sum_{i=0}^{n-1} \frac{1}{m-i} \\ &\leq \frac{1}{\alpha} \int_0^n \frac{1}{m-x} dx \\ &= \frac{1}{\alpha} \log \frac{1}{1-\alpha} \end{aligned}$$

[참고] 적재율이 우려스럽게 높아지면

- 적재율이 높아지면 일반적으로 해시 테이블의 효율이 떨어진다
- 일반적으로, 임계값을 미리 설정해 놓고 적재율이 이에 이르면
 - 해시 테이블의 크기를 두 배로 늘인 다음 해시 테이블에 저장되어 있는 모든 원소를 다시 해싱하여 저장한다