

알고리즘

# 해시 테이블 (Hash Table)

교재: 쉽게 배우는 알고리즘: 관계 중심의 사고법


저자: 문병로

출판사: 한빛미디어, 2013년 발행

# 수업 목표

- ▶ 해시 테이블의 발생 동기를 이해한다.
- ▶ 해시 테이블의 원리를 이해한다.
- ▶ 해시 함수 설계 원리를 이해한다.
- ▶ 충돌 해결 방법들과 이들의 장단점을 이해한다.

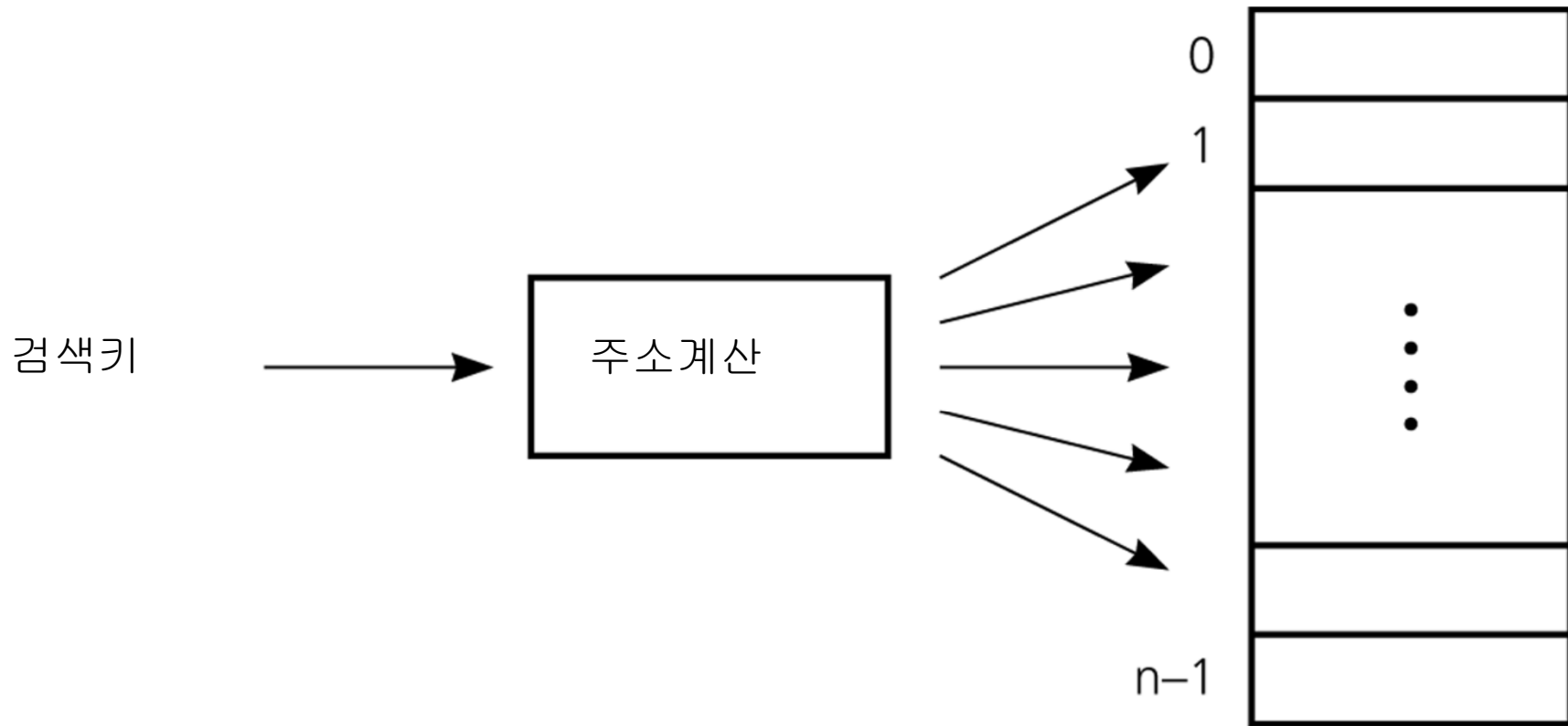
# 저장/검색의 복잡도

- 
- ▶ Array
    - ▶  $O(n)$
  - ▶ Binary search tree
    - ▶ 최악의 경우  $\Theta(n)$
    - ▶ 평균  $\Theta(\log n)$
  - ▶ Balanced binary search tree (RB Tree)
    - ▶ 최악의 경우  $\Theta(\log n)$
  - ▶ B-tree
    - ▶ 최악의 경우  $\Theta(\log n)$
    - ▶ Balanced binary search tree보다 상수 인자가 작다
  - ▶ Hash table
    - ▶ 평균  $\Theta(1)$

# 해시 테이블 (Hash Table)

- ▶ 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
    - ▶ 주민등록 시스템
- ▶ Hash table은 최소 원소를 찾는 것과 같은 작업은 지원하지 않는다.

# 주소 계산



배열 모양의 테이블

# 크기 13인 Hash Table에 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

Hash function  $h(x) = x \bmod 13$

# 해시 함수 (Hash Function)

- ▶ 입력 원소가 hash table에 고루 저장되어야 한다.
- ▶ 계산이 간단해야 한다.
- ▶ 여러가지 방법이 있으나 가장 대표적인 것은 division method와 multiplication method이다.



▶ Division Method

▶  $h(x) = x \bmod m$

▶  $m$ : table 사이즈. 대개 prime number임.

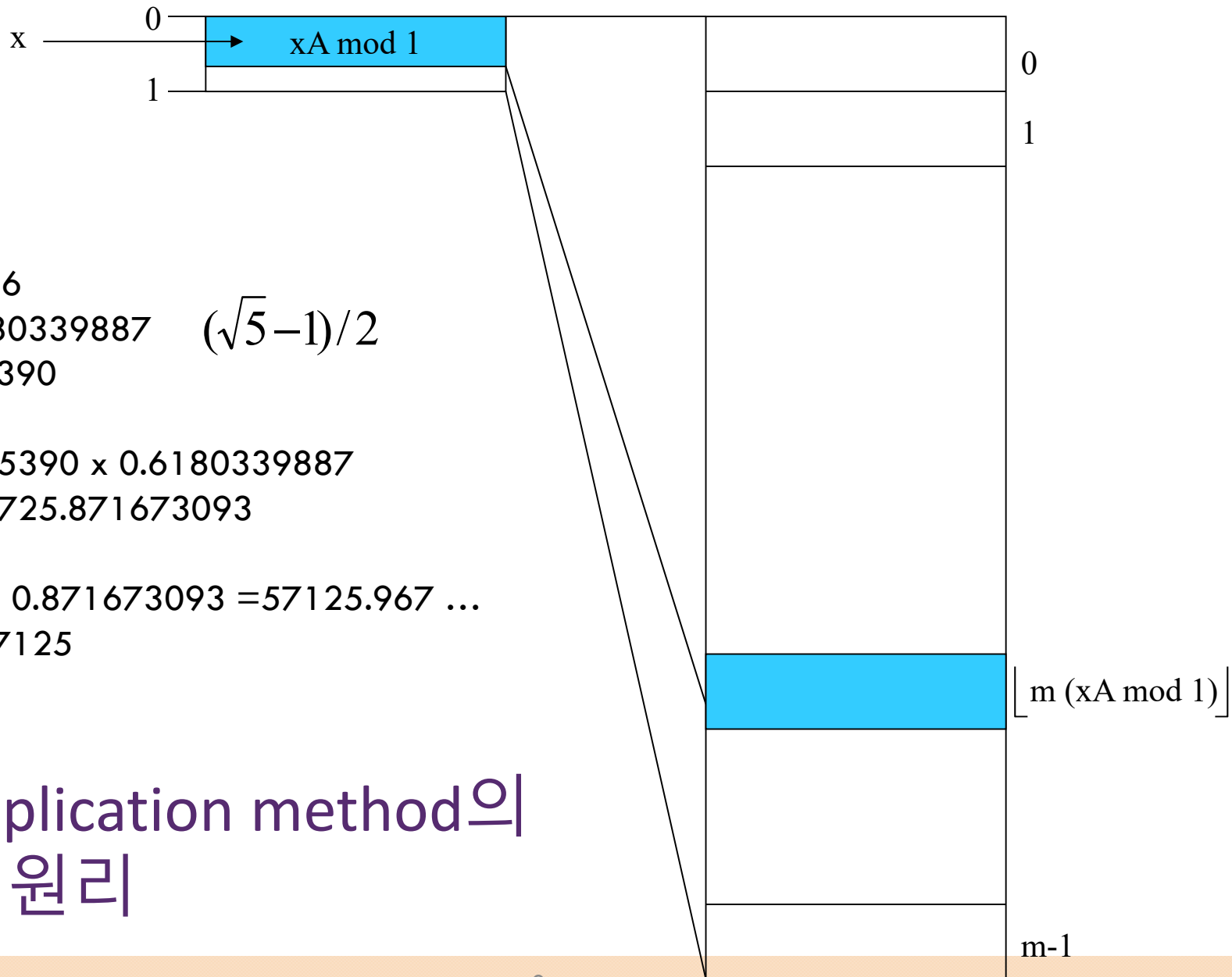
▶ Multiplication Method

▶  $h(x) = (xA \bmod 1) * m$

▶  $A$ :  $0 < A < 1$  인 상수

▶  $m$ 은 굳이 소수일 필요 없다. 따라서 보통  $2^p$ 으로 잡는다.  
( $p$ 는 정수)





$m=65536$

$A=0.6180339887 \quad (\sqrt{5}-1)/2$

$x=1025390$

$xA=1025390 \times 0.6180339887$   
 $=633725.871673093$

$65536 \times 0.871673093 = 57125.967 \dots$   
 $h(x) = 57125$

## Multiplication method의 작동 원리

# 충돌 (Collision)

- ▶ Hash table의 한 주소를 놓고 두 개 이상의 원소가 자리를 다투는 것
  - ▶ Hashing을 해서 삽입하려 하니 이미 다른 원소가 자리를 차지하고 있는 상황
- ▶ Collision resolution 방법은 크게 두 가지가 있다.
  - ▶ Chaining
  - ▶ Open Addressing

# Collision의 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

←  $h(29) = 29 \bmod 13 = 3$

←  
29를 삽입하려 하자 이미  
다른 원소가 차지하고 있다!

Hash function  $h(x) = x \bmod 13$

# 충돌 해결 (Collision Resolution)

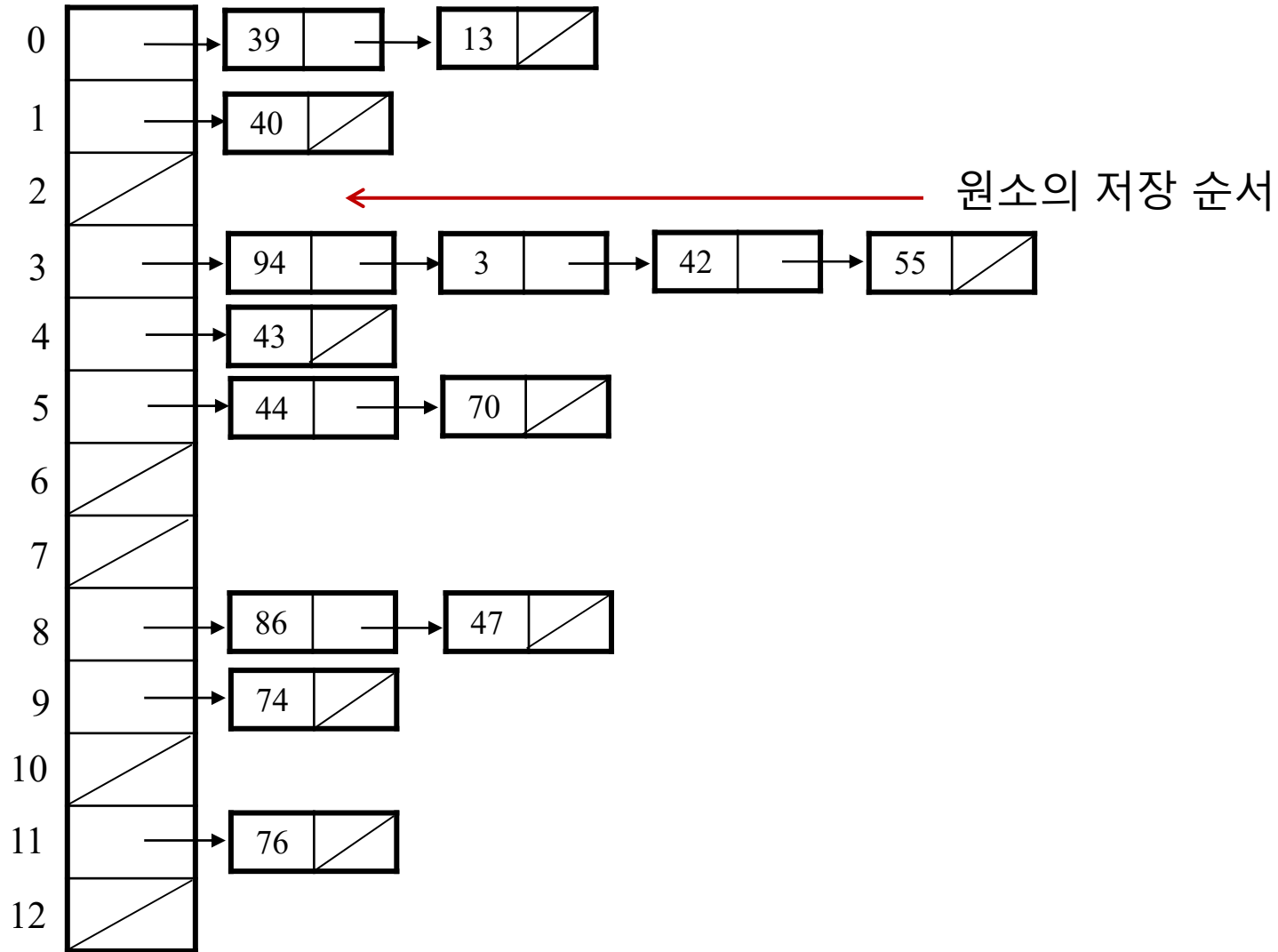
## ▶ Chaining

- ▶ 같은 주소로 hashing되는 원소를 모두 하나의 linked list로 관리한다.
- ▶ 추가적인 linked list 필요

## ▶ Open addressing

- ▶ Collision이 일어나더라도 어떻게든 주어진 테이블 공간에서 해결한다.
- ▶ 추가적인 공간이 필요하지 않다.

# Chaining을 이용한 Collision Resolution의 예



# 개방 주소 방법 (Open Addressing)

- ▶ 빈자리가 생길 때까지 해시값을 계속 만들어낸다.
  - ▶  $h_0(x), h_1(x), h_2(x), h_3(x), \dots$
- ▶ 중요한 세가지 방법
  - ▶ Linear probing
  - ▶ Quadratic probing
  - ▶ Double hashing

# 선형 조사 (Linear Probing)

$$h_i(x) = (h(x) + i) \bmod m$$

예: 입력 순서 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

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

# Linear Probing은 Primary Clustering에 취약하다

Primary clustering: 특정 영역에 원소가 몰리는 현상

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

← Primary clustering의 예



# 이차원 조사 (Quadratic Probing)

$$h_i(x) = (h(x) + c_1 i^2 + c_2 i) \bmod m$$

예: 입력 순서 15, 18, 43, 37, 45, 30

0	
1	
2	15
3	
4	43
5	18
6	45
7	
8	30
9	
10	
11	37
12	

$$h_i(x) = (h(x) + i^2) \bmod 13$$

$$30 \% 13 = 4$$

$$[(30 \% 13) + 1^2] \% 13 = 5$$

$$[(30 \% 13) + 2^2] \% 13 = 8$$

# Quadratic Probing은 Secondary Clustering에 취약하다

Secondary clustering: 여러 개의 원소가 동일한 초기 해시 함수값을 갖는 현상

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

$$28\%13 = 2$$

$$[(28\%13)+1^2]\%13 = 3$$

$$41\%13 = 2$$

$$[(41\%13)+1^2]\%13 = 3$$

$$[(41\%13)+2^2]\%13 = 6$$

$$67\%13 = 2$$

$$[(67\%13)+1^2]\%13 = 3$$

$$[(67\%13)+2^2]\%13 = 6$$

$$[(67\%13)+3^2]\%13 = 11$$

← Secondary clustering의 예

# 더블 해싱 (Double Hashing)

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

예: 입력 순서 15, 19, 28, 41, 67

$$h(x) = x \bmod 13$$

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

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

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

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

$$h_1(67) = 4$$

$$h_0(19) = 6$$

$$h_1(28) = 9$$

$$h_1(41) = 11$$

# 삭제시 유의 (예: 선형 조사의 경우)

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

(a) 원소 1 삭제

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

(b) 38 검색, 문제발생

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

(c) 표식을 해두면 문제없다

# Question & Answer

