

관계 중심의 사고법

쉽게 배우는 알고리즘

7장. 해시 테이블 Hash Table

7장.해시 테이블 Hash Table


그에게서 배운 것이 아니라
이미 내 속에 있었던 것이 그와 공명을 한 것이다.

- 머레이 겔만

학습목표

- 해시 테이블의 발생 동기를 이해한다.
- 해시 테이블의 원리를 이해한다.
- 해시 함수 설계 원리를 이해한다.
- 충돌 해결 방법들과 이들의 장단점을 이해한다.
- 해시 테이블의 검색 성능을 분석할 수 있도록 한다.

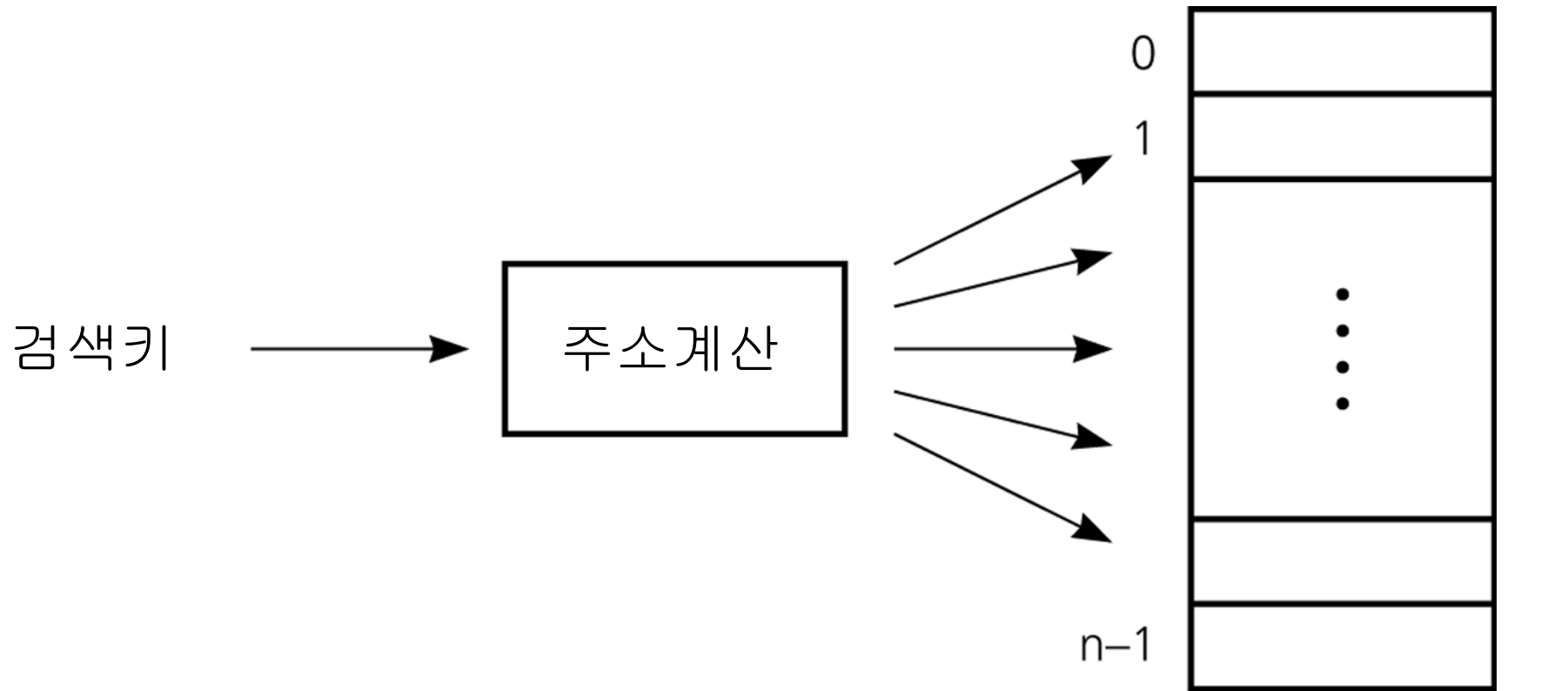
저장/검색의 복잡도

- 
- 배열
 - $O(n)$
 - 이진검색트리
 - 최악의 경우 $\Theta(n)$
 - 평균 $\Theta(\log n)$
 - 균형잡힌 이진검색트리(예: 레드블랙트리)
 - 최악의 경우 $\Theta(\log n)$
 - B-트리
 - 최악의 경우 $\Theta(\log n)$
 - Balanced binary search tree보다 상수 인자가 작다
 - 해시 테이블
 - 평균 $\Theta(1)$

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
 - 주민등록 시스템
- 해시 테이블은 최소 원소를 찾는 것과 같은 작업은 지원하지 않는다

주소 계산



배열 모양의 테이블

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

해시 함수 $h(x) = x \bmod 13$

해시 함수

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

- 나누기 방법 Division Method
 - $h(x) = x \bmod m$
 - m : 해시 테이블 사이즈. 대개 소수임.
- 곱하기 방법 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)

충돌 Collision

- 해시 테이블의 한 주소를 놓고 두 개 이상의 원소가 자리를 다투는 것
- 충돌 해결 방법은 크게 두 가지가 있다
 - 체이닝 Chaining
 - 개방주소 방법 Open Addressing

충돌의 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

← $h(29) = 29 \bmod 13 = 3$

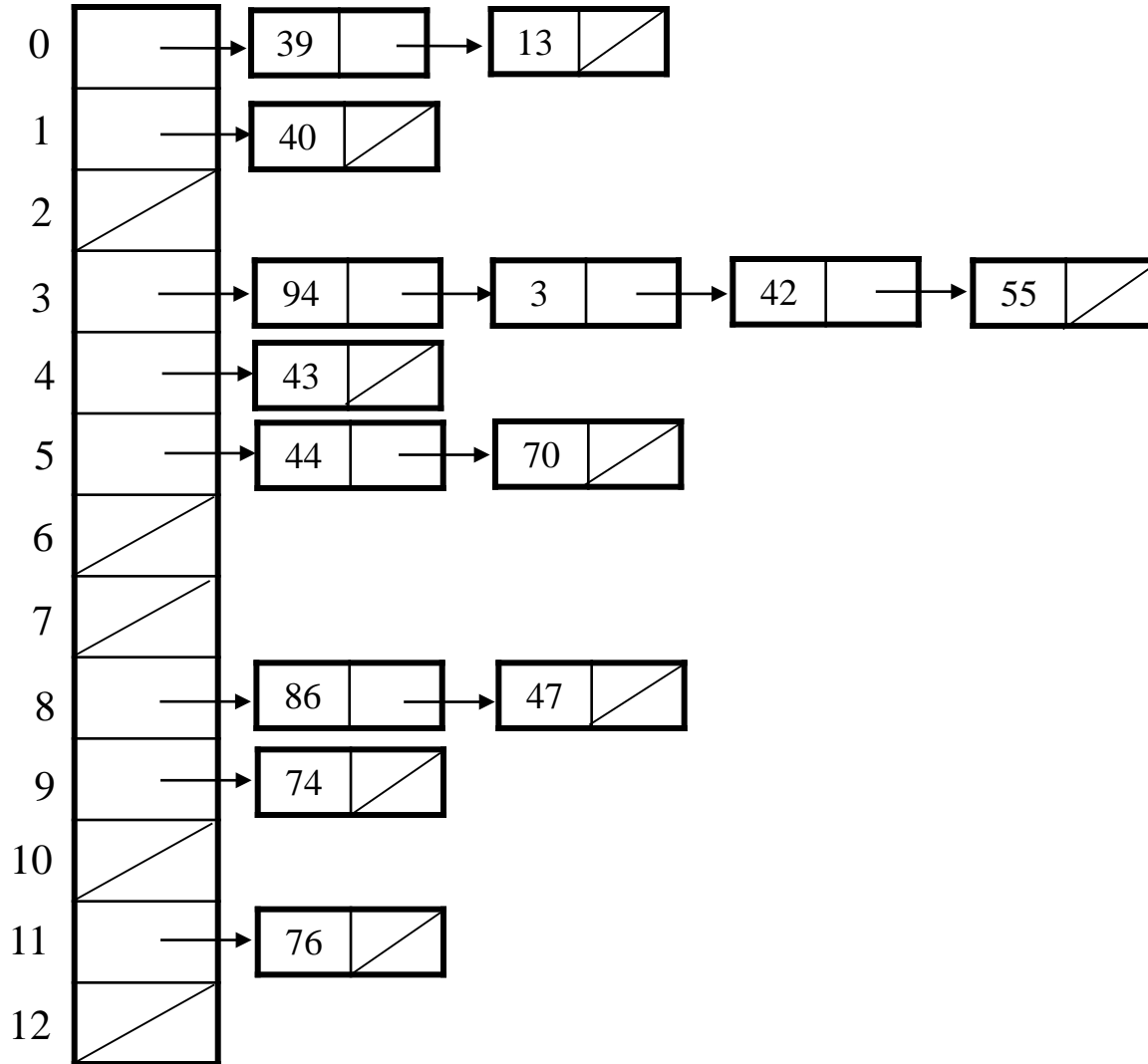
29를 삽입하려 하자
이미 다른 원소가 차지하고 있다!

해시함수 $h(x) = x \bmod 13$

충돌 해결 Collision Resolution

- 체이닝
 - 같은 주소로 해싱되는 원소를 모두 하나의 연결 리스트 `linked list`로 관리한다
 - 추가적인 연결 리스트 필요
- 개방주소 방법
 - 충돌이 일어나더라도 어떻게든 주어진 테이블 공간에서 해결한다
 - 추가적인 공간이 필요하지 않다

체이닝



개방주소 방법

- 빈자리가 생길 때까지 해시값을 계속 만들어낸다
 - $h_0(x), h_1(x), h_2(x), h_3(x), \dots$
- 중요한 세가지 방법
 - 선형 조사
 - 이차원 조사
 - 더블 해싱

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

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

선형 조사는 1차군집에 취약하다

1차군집: 특정 영역에 원소가 몰리는 현상

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

← 1차군집의 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	

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

이차원 조사는 2차군집에 취약하다

2차군집: 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	

← 2차군집의 예

더블 해싱 Double Hashing

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

예: 입력 순서 15, 19, 28, 41, 67

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

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

$$h_1(67) = 3$$

$$h_1(28) = 8$$

$$h_1(41) = 10$$

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

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

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

삭제시 조심할 것

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 10이 삭제된다

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

해시 테이블에서의 검색 시간

- 적재율 α
 - 해시 테이블 전체에서 얼마나 원소가 차 있는지를 나타내는 수치
 - 해시 테이블에 n 개의 원소가 저장되어 있다면 $\alpha = n/m$ 이다
- 해시 테이블에서의 검색 효율은 적재율과 밀접한 관련이 있다

체이닝에서의 검색 시간

- 정리 1
 - 체이닝을 이용하는 해싱에서 적재율이 α 일 때, 실패하는 검색에서 조사 횟수의 기대치는 α 이다
- 정리 2
 - 체이닝을 이용하는 해싱에서 적재율이 α 일 때, 성공하는 검색에서 조사 횟수의 기대치는 $1 + \alpha/2 - \alpha/2n$ 이다

개방주소 방법에서의 검색 시간

- 가정
 - 조사순서 $h_0(x), h_1(x), \dots, h_{m-1}(x)$ 가 0부터 $m-1$ 사이의 수로 이루어진 순열을 이루고, 모든 순열은 같은 확률로 일어난다
- 정리 3
 - 적재율 $\alpha=n/m$ 인 개방주소 해싱에서 실패하는 검색에서 조사횟수의 기대치는 최대 $1/(1-\alpha)$ 이다
- 정리 4
 - 적재율 $\alpha=n/m$ 인 개방주소 해싱에서 성공하는 검색에서 조사횟수의 기대치는 최대 $(1/\alpha)\log(1/(1-\alpha))$ 이다

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



Thank you
