

해시 테이블

알고리즘 스터디#1

2021.08.12

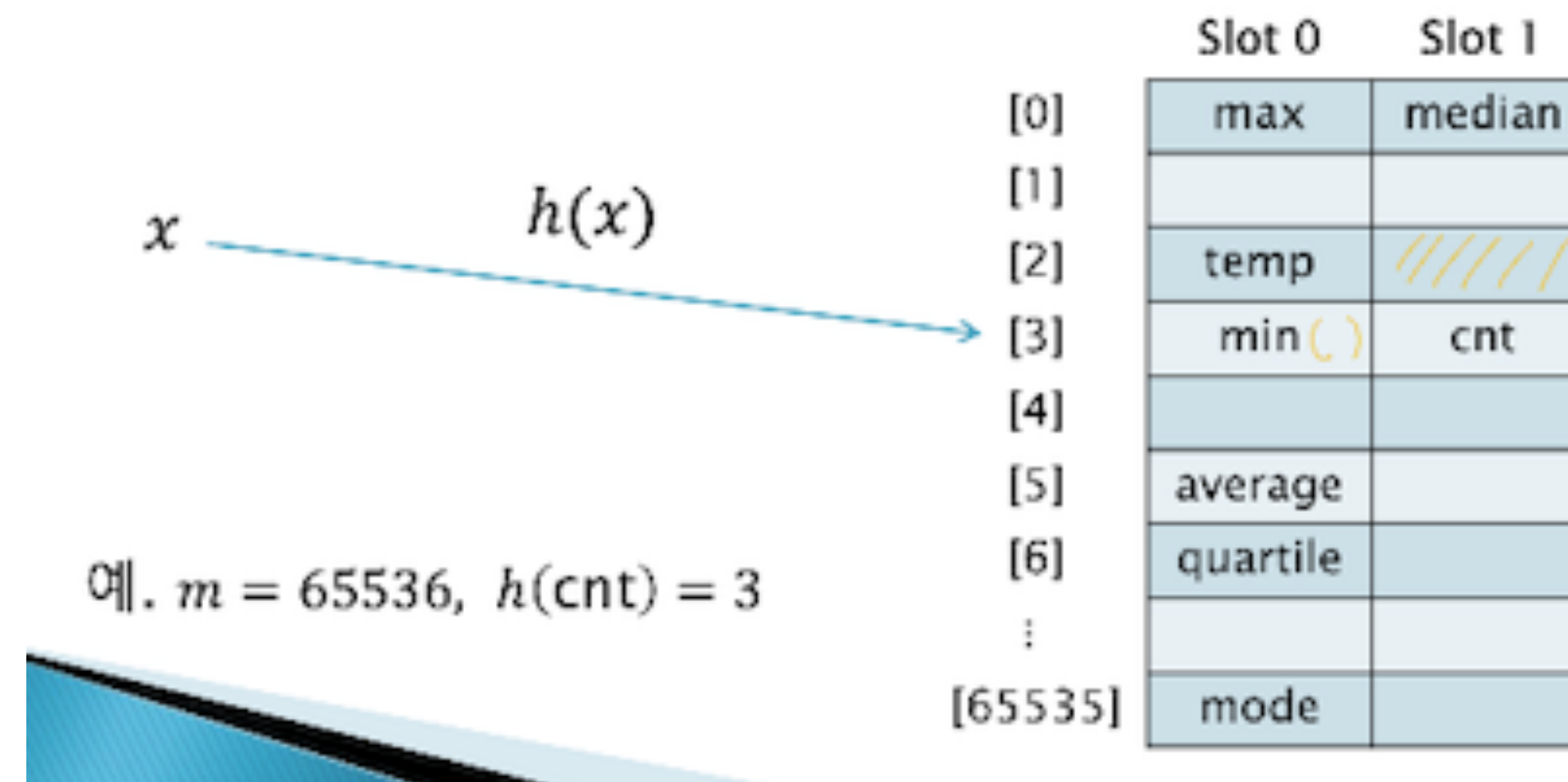
해싱이란?

- 임의의 결과값을 **해시함수(Hash Function)**를 사용하여 고정된 크기의 값으로 변환하는 작업
- 해시 함수로부터 얻은 값을 해시 값(hash value), 해시 코드(hash code) 등으로 부른다.



해시 테이블이란?

- 해싱을 사용하여 데이터를 저장하는 자료구조
- 해시함수를 사용하여 변환한 값을 색인(index)으로 삼아 키(key)와 데이터(value)를 저장하는 자료구조
- 기본연산으로는 탐색(Search), 삽입(Insert), 삭제>Delete)가 있다.



Hashing Data Structure

List = [11, 12, 13, 14, 15]

$$H(x) = [x \% 10]$$

$11 \% 10$ $12 \% 10$ $13 \% 10$ $14 \% 10$ $15 \% 10$

0 1 2 3 4 5

Hash Table

	11	12	13	14	15
--	----	----	----	----	----



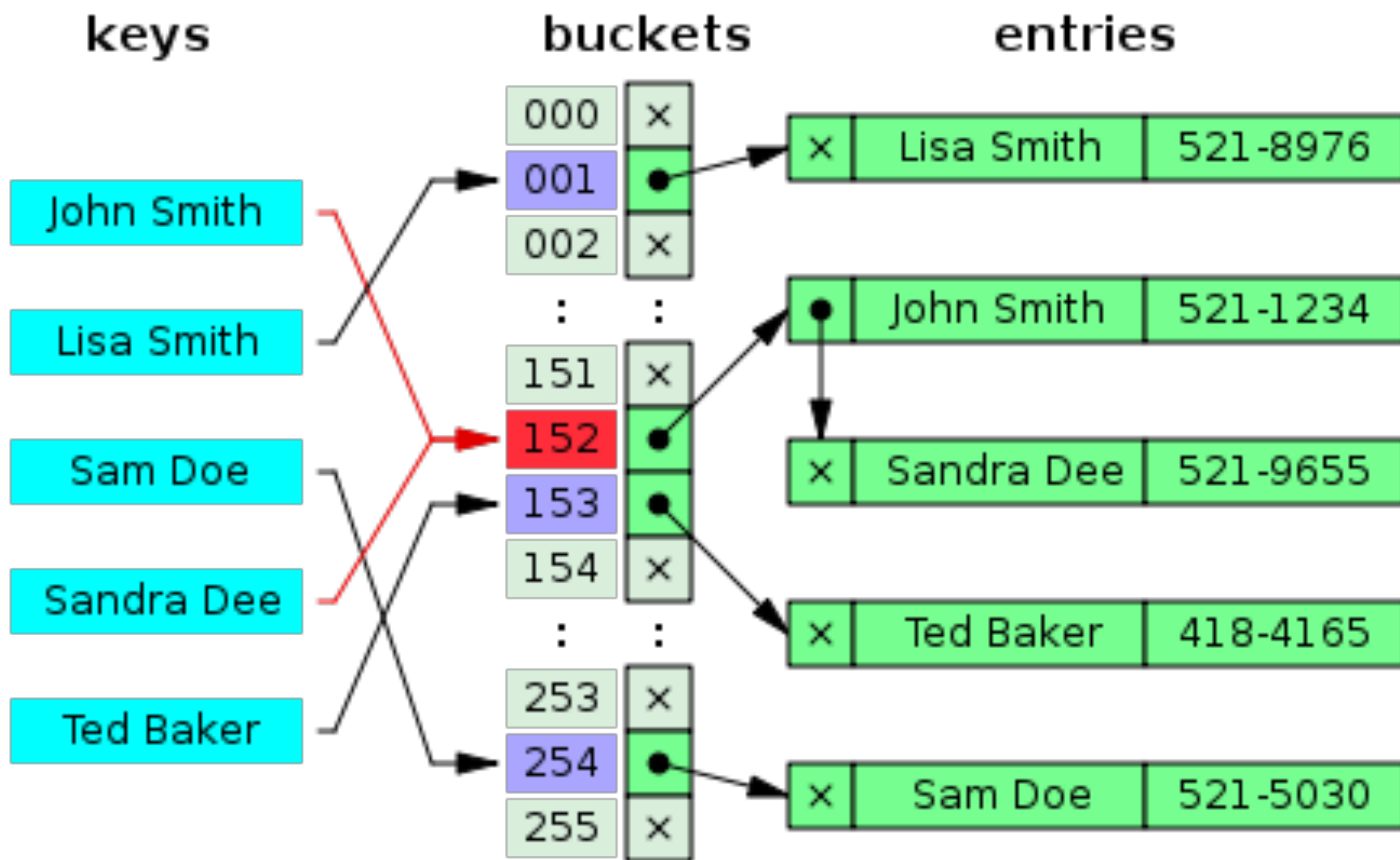
충돌 & 오버플로

충돌과 오버플로

- 체인법(chaining) : 오버플로된 동거자를 위한 저장 공간을 상대 화일 밖에서 찾아 해결, 즉 독립된 오버플로 구역을 할당
- 개방 주소법(open addressing) : 오버플로된 동거자를 저장할 공간을 상대 화일 내에서 찾아 해결
 - (1) 선형 조사(linear probing)
 - (2) 이차 조사(quadratic probing)
 - (3) 이중 해싱(double hashing)
 - (4) 재해싱(rehashing)

충돌 처리1

체이닝



충돌 처리2

선형 조사

[0]	72
[1]	
[2]	18
[3]	43
[4]	36
[5]	
[6]	6
[7]	

Add the keys 10, 5, and 15 to the previous table .

Hash key = key % table size

$$2 = 10 \% 8$$

$$5 = 5 \% 8$$

$$7 = 15 \% 8$$

[0]	72
[1]	15
[2]	18
[3]	43
[4]	36
[5]	10
[6]	6
[7]	5

충돌 처리3

이차 조사

$$89 \% 10 = 9$$

$$18 \% 10 = 8$$

$$49 \% 10 = 9 - 1 \text{ attempt needed} - 1^2 = 1 \text{ spot}$$

$$58 \% 10 = 8 - 3 \text{ attempts} - 3^2 = 9 \text{ spots}$$

$$69 \% 10 = 9 - 2 \text{ attempts} - 2^2 = 4 \text{ spots}$$

[0]	49
[1]	
[2]	
[3]	69
[4]	
[5]	
[6]	
[7]	58
[8]	18
[9]	89

충돌 처리4

이중 해싱

Table Size = 10 elements

$\text{Hash}_1(\text{key}) = \text{key} \% 10$

$\text{Hash}_2(\text{key}) = 7 - (\text{k} \% 7)$

Insert keys : 89, 18, 49, 58, 69

$\text{Hash}(89) = 89 \% 10 = 9$

$\text{Hash}(18) = 18 \% 10 = 8$

$\text{Hash}(49) = 49 \% 10 = 9$ a collision !
= $7 - (49 \% 7)$
= 7 positions from [9]

$\text{Hash}(58) = 58 \% 10 = 8$
= $7 - (58 \% 7)$
= 5 positions from [8]

$\text{Hash}(69) = 69 \% 10 = 9$
= $7 - (69 \% 7)$
= 1 position from [9]

[0]	49
[1]	
[2]	
[3]	69
[4]	
[5]	
[6]	
[7]	58
[8]	18
[9]	89

충돌 처리5

재해싱

Rehashing Example

Hash Table with linear probing with input 13, 15, 6, 24

$$h(x) = x \bmod 7$$
$$\lambda = 0.57$$

0	6
1	15
2	
3	24
4	
5	
6	13

Insert 23
 $\lambda = 0.71$

0	6
1	15
2	23
3	24
4	
5	
6	13

$$h(x) = x \bmod 17$$
$$\lambda = 0.29$$

Rehashing

0	
1	
2	
3	
4	
5	
6	6
7	23
8	24
9	
10	
11	
12	
13	13
14	
15	15
16	

해싱 시간복잡도

- 평균 $O(1)$
- 최악 $O(n)$

풀어올 예제

https://programmers.co.kr/learn/courses/30/lessons/42577

코딩테스트 연습 > 해시 > 전화번호 목록

도움말 컴파일 옵션

전화번호 목록

dark light sublime vim emacs C++ ▾

문제 설명

전화번호부에 적힌 전화번호 중, 한 번호가 다른 번호의 접두어인 경우가 있는지 확인하려 합니다. 전화번호가 다음과 같을 경우, 구조대 전화번호는 영석이의 전화번호의 접두사입니다.

- 구조대 : 119
- 박준영 : 97 674 223
- 지영석 : 11 9552 4421

전화번호부에 적힌 전화번호를 담은 배열 phone_book 이 solution 함수의 매개변수로 주어질 때, 어떤 번호가 다른 번호의 접두어인 경우가 있으면 false를 그렇지 않으면 true를 return 하도록 solution 함수를 작성해주세요.

제한 사항

- phone_book의 길이는 1 이상 1,000,000 이하입니다.
 - 각 전화번호의 길이는 1 이상 20 이하입니다.
 - 같은 전화번호가 중복해서 들어있지 않습니다.

입출력 예제

phone_book	return
["119", "97674223", "1195524421"]	false
["123","456","789"]	true
["12","123","1235","567","88"]	false

입출력 예 설명

입출력 예 #1
앞에서 설명한 예와 같습니다.

입출력 예 #2
한 번호가 다른 번호의 접두사인 경우가 없으므로, 답은 true입니다.

solution.cpp

```
1  #include <string>
2  #include <vector>
3
4  using namespace std;
5
6  bool solution(vector<string> phone_book) {
7      bool answer = true;
8      return answer;
9  }
```

실행 결과

실행 결과가 여기에 표시됩니다.

질문하기 (161)

테스트 케이스 추가하기

로그인하기