자료 구조 스터디 08

2021-2 KCA

해시 테이블 (Hash Table)

본 ppt의 자료는 Pt.J님의 자료와 김성열 교수님의 강의 및 여러 블로그를 참고하였음을 밝힙니다.

Hash Table (== Hash Map)

- 데이터를 아주 빠르게(O(1)) 삽입하거나 꺼낼 때 사용하는 자료구조
- 단, 최소값과 최댓값을 찾을 때, 전체 자료를 Search할 때, 해시 충돌이 많이 일어날 때는 효율이 떨어진다.

- (key, value) 구조로 이루어짐
  파이썬의 딕셔너리가 해시 테이블로 구현되었다.

- 공간복잡도를 희생해서(공간을 많이 써서) 시간복잡도를 줄인다.

- 희소 배열로 구현된다. (희소 배열 : 배열 내 값이 연속적이지 않은 배열)

Hash Table

<작동 예시>

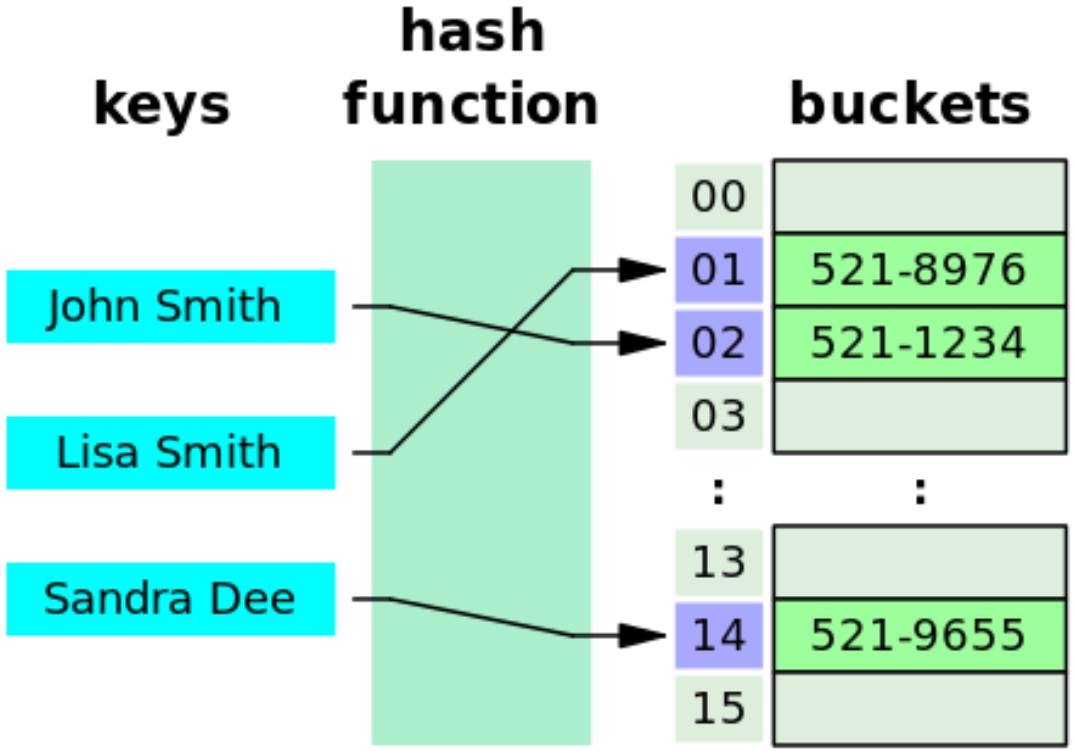
1. key 값들이 주어짐.

2. 어떠한 과정(hash function)을 거쳐서 key 값들을 서로 중복되지 않는 숫자값으로 변환.

3. 변환된 숫자값을 배열 index로 사용.

=> 변환된 값이 다르면 원래 값도 다르다는 것이 보장. (역은 보장x, 해시 충돌)

배열에 index로 접근하므로 삽입, 삭제 O(1)



https://en.wikipedia.org/wiki/Hash_table

Hash function

- 임의 길이 데이터를 고정된 길이의 데이터로 매핑하는 함수
- 매핑된 해시 값만으로 원래 데이터를 복원하는 것이 힘들다는 것을 이용해서 암호화에 사용 (SHA-2, SHA-3 등)
- 해시 함수의 시간복잡도가 매우 빠를수록, 해시 충돌이 덜 일어날수록 좋은 함수

Hash function 작동 방식 예시

keys

91
32
558
6
204
50
39

해시 함수 :
f(hash_key) = hash_key % array_size

Hash Table

0	50
1	91
2	32
3	
4	204
5	
6	6
7	
8	558
9	39

Hash function의 대표적 예시

1. Division Hashing : 입력 key 값을 테이블의 크기로 나눈 나머지. 테이블의 크기가 소수이고 2의 거듭제곱과 먼 값일수록 효과가 좋다.

2. Digit Folding : 각 key의 문자열을 ASCII 코드로 바꾸고 모든 값을 합한 데이터를 테이블 주소로 사용.

+) Multiplication Hashing, Univeral Hashing, Fibonacci Hashing...

참고 : https://en.wikipedia.org/wiki/Hash_function

Hash 충돌 예시 (분리 연결법, Separate Chaining)

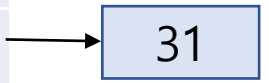
keys

91
32
558
6
204
50
31

f(hash_key) =
hash_key % array_size

Hash Table

0	50
1	91
2	32
3	
4	204
5	
6	6
7	
8	558
9	



연결리스트로 연결해 나간다.

Hash 충돌 시 저장 방법

언어	방식
C++(GCC libstdc++)	개별 체이닝
자바	개별 체이닝
고(Go)	개별 체이닝
루비	오픈 어드레싱
파이썬	오픈 어드레싱

- Separate Chaining
  연결리스트로 중복된 index 뒤에서 연결해 나가는 방식

- Open Addressing
  충돌 발생 시 그 아래 index부터 탐색하면서 빈 공간에 집어넣는 방식
  (보통 hash table은 충분히 크므로 이 선형 탐색도 꽤 효율적이다.)

- 해시 충돌은 대부분의 해시 함수에서 피할 수 없다. 그래서 해시 충돌이 일어나기까지 필요한 계산량을 압도적으로 늘리는 방식으로 회피.
- 보통 table 크기에 비해 data가 저장되는 한계를 50%~75%로 정해두기 때문에 공간 복잡도가 커지게 된다.

Hash와 암호

<salt>

- 원문에 추가 문자열을 집어넣어 암호화 하는 방식.
- 대부분의 해시 함수는 공개되어 있어, 해커들이 많은 입력값에 대한 해시 함수값을 들고 와서 대조하면서 알아낼 위험이 있기 때문.
- 난수 생성으로 만든 수천 비트의 문자열을 사용하는 것을 권장.

<rainbow table> : 많은 입력값에 대한 해시 함수값을 저장한 표

회원가입 서비스를 구현할 때, 개인정보보호법에 의해 db에 사용자의 비밀번호를 해싱(PBKDF2 보안성 이상의 함수로)해서 저장해야 한다!!

보통 rainbow table이 만들어질 가능성을 줄이기 위해 해시함수가 돌아가는데 오래 걸리는 bcrypt, scrypt를 쓰고, salt도 뿌려야 한다.

Hash와 Git

- Git은 commit 끼리, 파일 끼리, 소스 코드가 변경되었는지 비교하기 위해 SHA-1 해시를 사용

- SHA-1 해시는 1GB 크기의 파일들도 160비트로 줄여주기 때문에 빠른 비교가 가능

- Git은 각 파일을 SHA-1으로 이름을 붙여 저장한다.

+) git은 commit 할 때, 변경되지 않은 파일은 복사하지 않고 링크(바로가기)만 저장해서 총 용량을 줄인다.