해싱 (Hashing)

01 해싱 이론
02 해시 셋 이론
03 해시 테이블 이론
04 해시 충돌
05 해시의 활용

신 제 용

01 해싱 이론

해싱이란 무엇인지 이해하고, 해시 함수의 특성과 의의를 학습합니다.

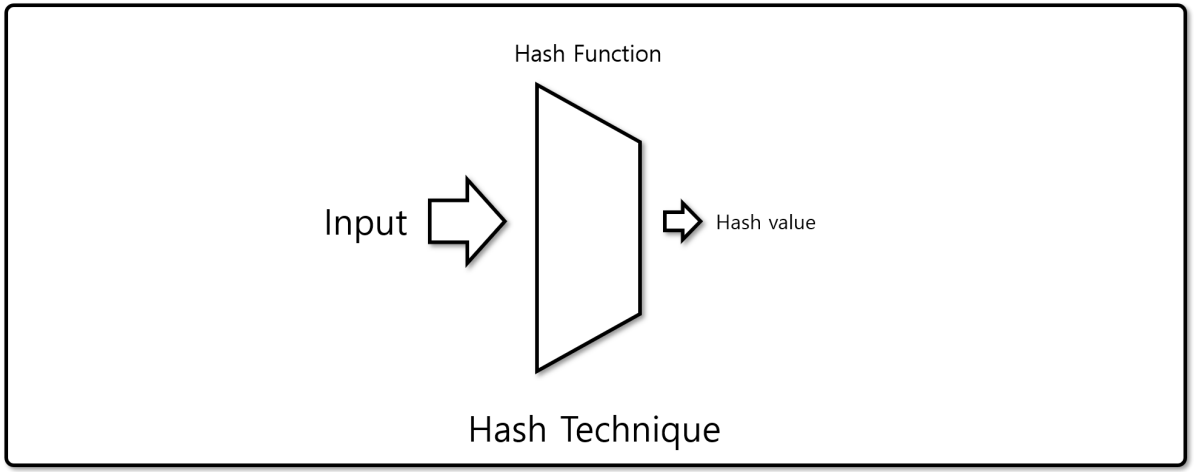
학습 키워드 - 해싱, 해시 함수

Chapter 01
해싱 이론

ZEROBASE SCHOOL

해싱 (Hashing)

- 넓은 범위의 값을 더 좁은 범위의 값으로 변환하는 기법
- 동일한 입력에 대해서 항상 동일한 출력을 보장



해시 함수 (Hash function)

- 해싱에 사용되는 함수로, 자주 사용되므로 빠른 동작이 요구된다.
- 다른 입력이 동일한 출력을 가질 수 있으며, 이것을 해시 충돌 (Hash collision)이라 한다.
- 파이썬에서는 hash() 함수가 내장 함수로 구현되어 있다.

02 해시 셋 이론

다음 챕터에서는 해싱을 이용하는 대표적인 자료구조인
해시 셋에 대해서 알아봅시다.

Chapter 01
해싱 이론

02 해시 셋 이론

해싱을 이용하는 대표적인 자료구조인 해시 셋에 대해 알아봅니다.

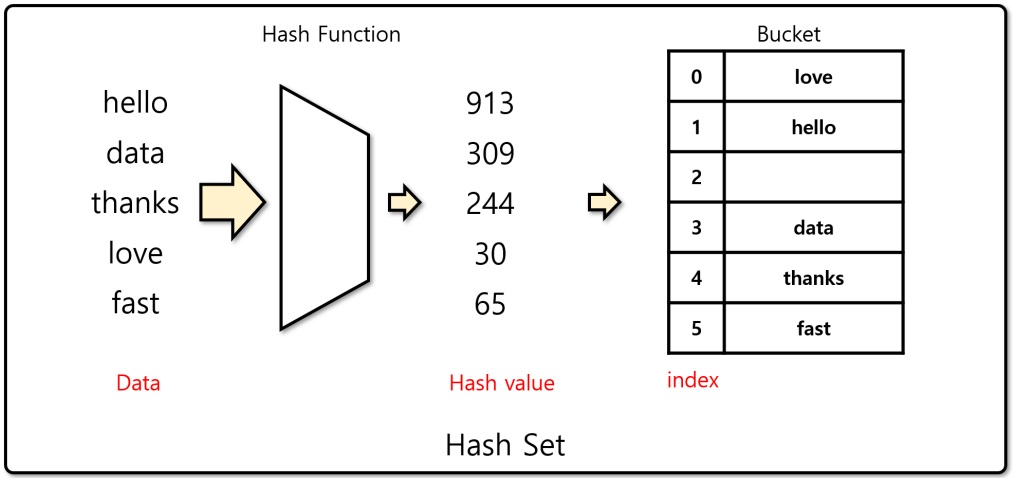
학습 키워드 - 해싱, 해시 셋, 집합, 셋

Chapter 02
해시 셋 이론

ZEROBASE SCHOOL

해시 셋 (Hash set)

- 해싱 기법을 이용하는 대표적인 자료구조
- 한정된 크기를 가지는 버킷(bucket)을 이용해 자료를 저장
- 자료를 저장하는 인덱스는 해시 함수의 출력인 해시 값을 버킷의 크기로 나눈 나머지로 한다 (index = hash_value % bucket_size)



해시 셋의 특징

- 파이썬에는 set 자료형으로 해시 셋이 구현되어 있다.
- 자료의 중복을 허용하지 않으며, 빠르게 자료를 탐색할 수 있다.
- 버킷이 가득 차면 버킷의 크기를 증가시켜 재배치한다.

03 해시 테이블 이론

다음 챕터에서는 해싱을 이용하는 또 다른 자료구조인
해시 테이블에 대해서 알아봅시다.

Chapter 02
해시 셋 이론

ZEROBASE SCHOOL

03 해시 테이블 이론

해싱을 이용하는 대표적인 자료구조인 해시 테이블에 대해 알아봅니다.

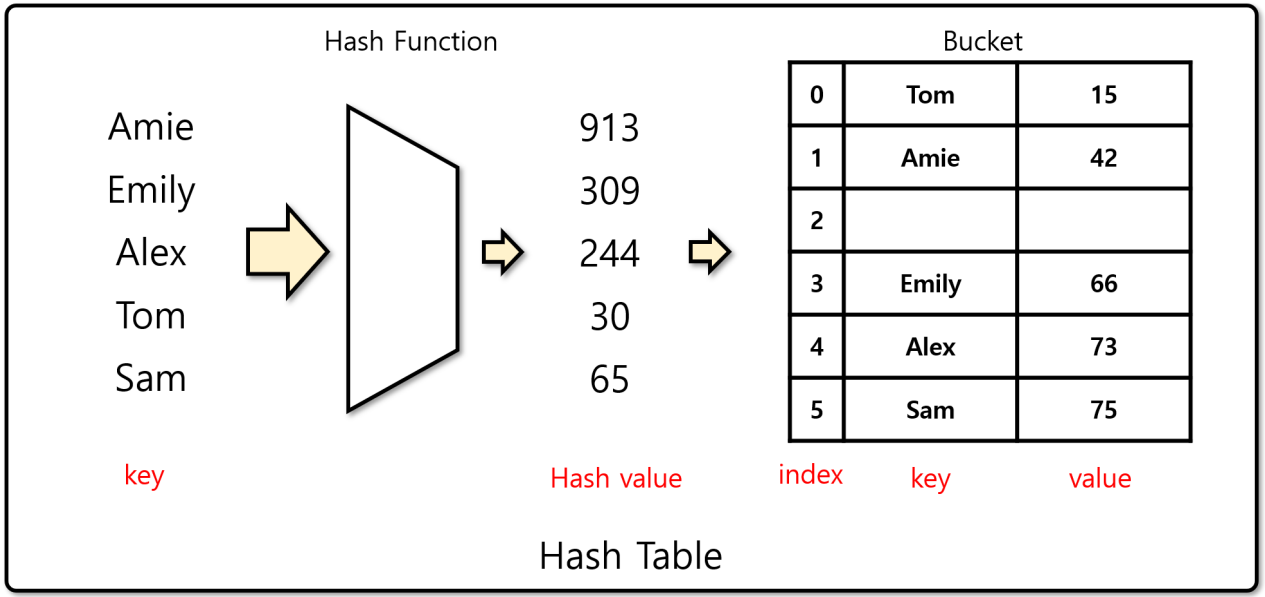
학습 키워드 - 해싱, 해시 테이블, 딕셔너리

Chapter 03
해시 테이블 이론

ZEROBASE SCHOOL

해시 테이블 (Hash table)

- 해시 셋과 유사하나, 해시를 계산하는 key와 짝을 이루는 value를 함께 버킷에 저장하는 자료구조



해시 테이블의 특징

- 파이썬에는 dict 자료형으로 해시 테이블이 구현되어 있다.
- Key는 중복이 허용되지 않으나, value는 중복이 허용된다.
- 공간 복잡도를 희생하여 시간 복잡도를 낮추는 대표적인 자료구조

04 해시 충돌

다음 챕터에서는 해싱을 이용하는 자료구조에서 발생하는 문제인
해시 충돌에 대해서 알아봅시다.

Chapter 03
해시 테이블 이론

04 해시 충돌

해시 셋과 해시 테이블에서 발생할 수 있는 해시 충돌 문제를 알아보고, 대표적인 해결 방법을 학습합니다.

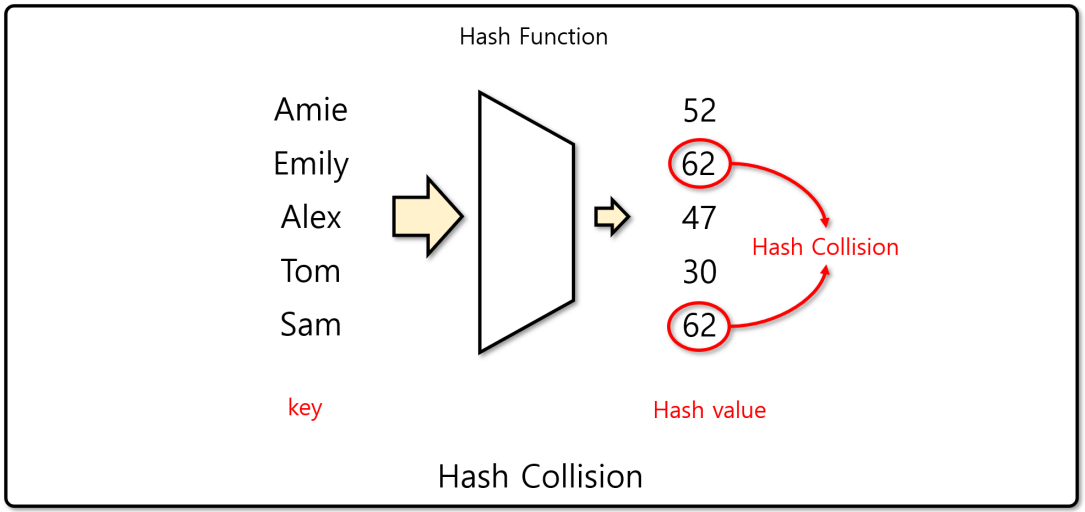
학습 키워드 - 해시 충돌, 선형 탐사, 체이닝

Chapter 04
해시 충돌

ZEROBASE SCHOOL

해시 충돌 (Hash collision)

- 해시 함수가 서로 다른 입력에 대해 동일한 출력을 내는 경우
- 해시 충돌이 100% 발생할 경우, 탐색 성능은 $O(1)$에서 $O(N)$으로 감소

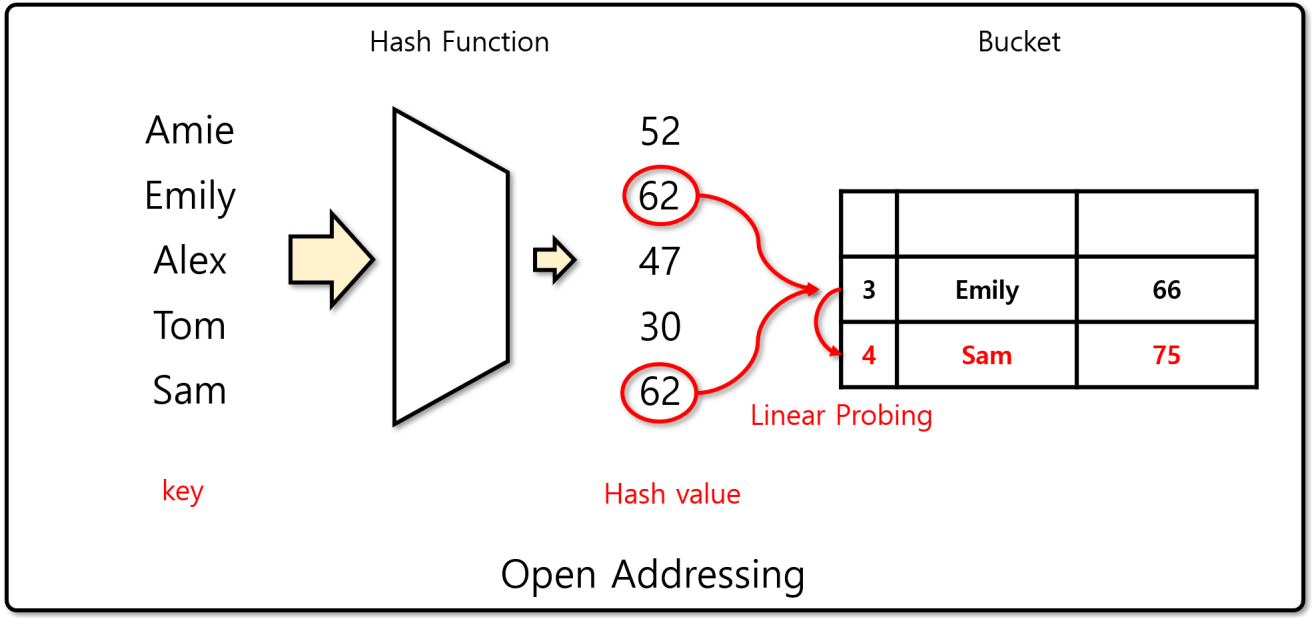


해시 충돌의 해결

- 개방 주소법 (Open addressing)
  - 버킷의 다른 index를 선택하여 자료를 저장하는 기법
  - 개방 주소법은 더 이상 해시 충돌이 발생하지 않을 때 까지 반복

- 개방 주소법의 종류
  - 선형 탐사 (Linear probing) – index를 1씩 이동시키는 방법
  - 이차 탐사 (Quadratic probing) – index를 $1^2, 2^2, 3^2, \cdots$ 씩 이동
  - 이중 해싱 (Double hashing) – 별도의 해시 함수로 이동 간격을 계산

선형 탐사 (Linear probing)

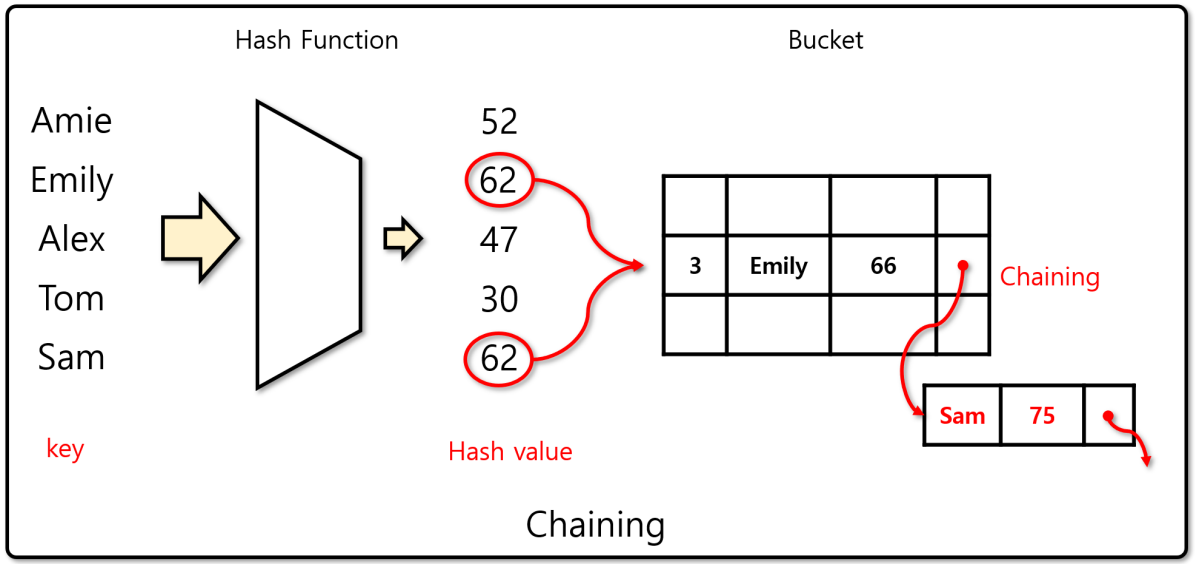


Chapter 04
해시 충돌

ZEROBASE SCHOOL

체이닝 (Chaining)

- 해시 충돌이 발생할 경우 버킷에 자료를 연결 리스트로 추가해 나가는 기법



05 해시의 활용

다음 챕터에서는 해시 셋과 해시 테이블을 활용하는
기초 유형 문제를 풀이합니다.

Chapter 04
해시 충돌

05 해시의 활용

해시 테이블의 대표적인 활용 예를 기초 유형 문제 풀이를 통해 학습합니다.

학습 키워드 - 해시 셋, 해시 테이블, 셋, 딕셔너리, 구현

Chapter 05
해시의 활용

ZEROBASE SCHOOL

```python
# Prob1.
# 리스트의 중복을 제거하고 출력하시오.
# 단, 출력 리스트의 순서는 오름차순으로 정렬하시오.
#
# 입출력 예시)
# arr = ["hello", "zero", "base", "buy", "zero", "hello"]
# 결과: ["base", "buy", "hello", "zero"]


def solution(arr):
    pass


if __name__ == '__main__':
    arr = ["hello", "zero", "base", "buy", "zero", "hello"]
    sol = solution(arr)
    print(sol)
```

Chapter 05
해시의 활용

ZEROBASE SCHOOL

```python
# Prob2.
# 정수로 이루어진 리스트 arr에서,
# 총 합이 0이 되는 부분 리스트가 있는지 여부를 출력하시오.
# 부분 리스트란, 리스트의 일부 연속된 부분을 잘라 만든 리스트를 말한다.
#
# 예시 입출력
# arr = [3, 4, -7, 3, 1, 3, 1, -4, -2, -2]
# 출력: True
#


def solution(arr):
    pass


if __name__ == '__main__':
    arr = [3, 4, -7, 3, 1, 3, 1, -4, -2, -2]
    sol = solution(arr)
    print(sol)
```

Chapter 05
해시의 활용

ZEROBASE SCHOOL

```
# Prob3.
# 철수와 영희는 눈을 가리고 잡기 놀이를 하기로 했다.
# 철수는 도망치는 역할을 맡았으며, (x1, y1) 좌표에서 출발한다.
# 매 1초마다 철수는 동/서/남/북 중 한군데를 임의로 선택하여 이동한다.
# 영희는 철수를 잡는 역할을 맡았으며, (x2, y2) 좌표에서 출발한다.
# 매 1초마다 영희는 북서/북동/남서/남동 중 한군데를 임의로 선택하여 이동한다.
# 모든 경우 중에서, 영희가 철수를 가장 빨리 잡는 경우 몇 초의 시간이 걸리는지
출력하시오.
#
# 입력설명
# -1000 <= x1 <= 1000
# -1000 <= y1 <= 1000
# -1000 <= x2 <= 1000
# -1000 <= y2 <= 1000
#
# 입출력 예시
# x1 = 2
# y1 = 4
# x2 = 5
# y2 = -3
# 출력: 4
```

```
def solution(x1, y1, x2, y2):
    pass


if __name__ == '__main__':
    x1 = 2
    y1 = 4
    x2 = 5
    y2 = -3
    sol = solution(x1, y1, x2, y2)
    print(sol)
```