세종대학교

Data Engineering

데이터엔지니어링

[해시 기반 데이터 엔지니어링(1)]
- HashSet CRUD -

HashSet 소개

학습내용

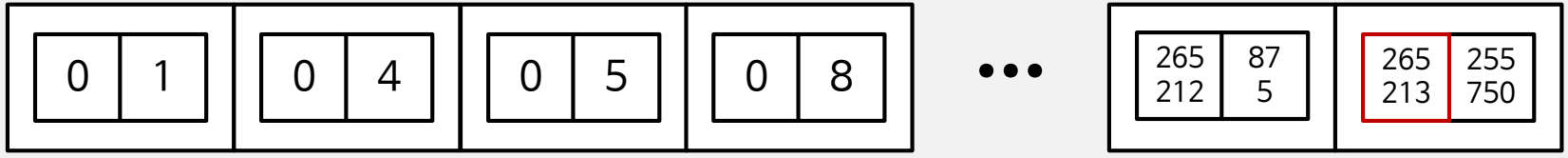
1. HashSet
2. HashSet을 구성하는 Interface

학습목표

- HashSet의 개념을 설명할 수 있다.
- HashSet을 구성하는 Interface의 연산을 수행할 수 있다.

특정 보낸사람 ID가 있는지 확인하기

List<Email>



ID 0이 있는가? 바로 종료

ID 8이 있는가? 중간까지 탐색

ID 265213이 있는가? 모든 데이터를 확인해야 함(최악의 경우)

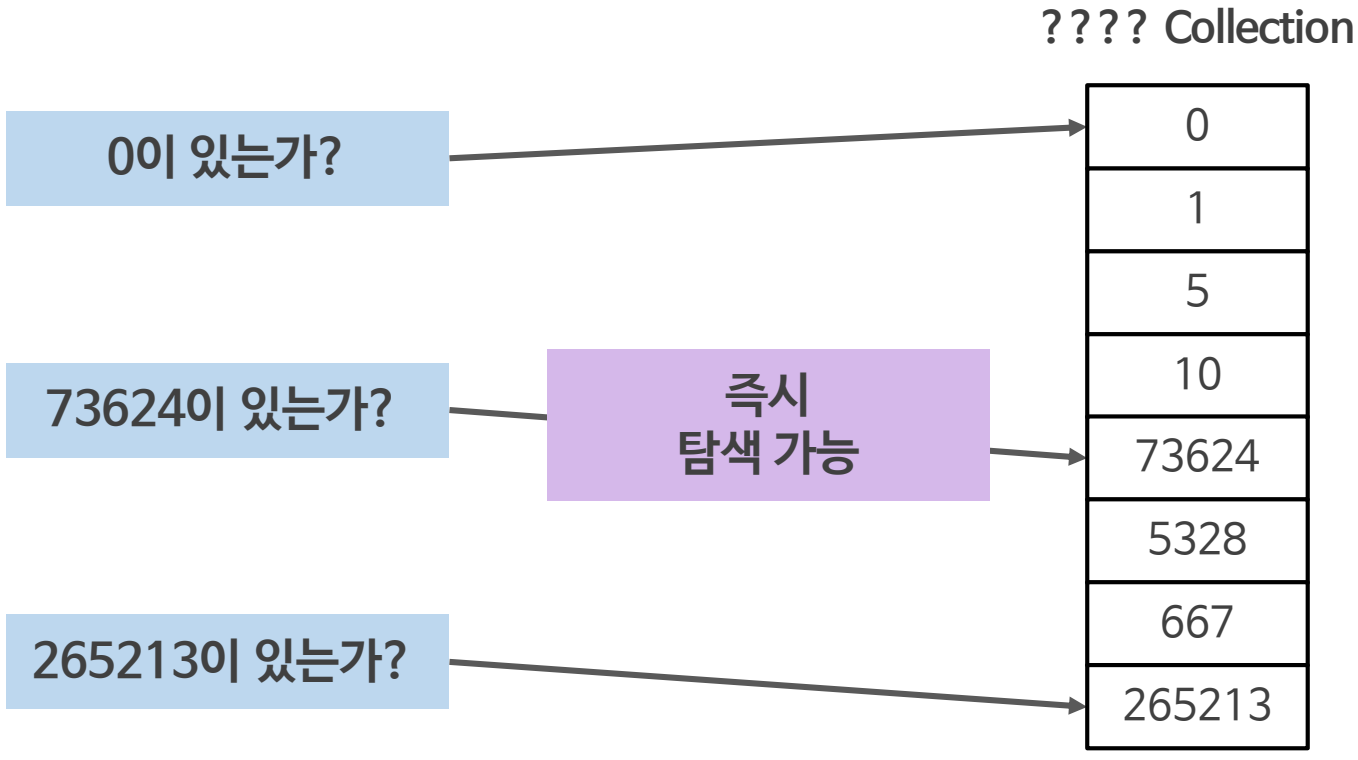
boolean isExist를 false로 시작하여
특정 보낸사람 ID가 발견되면 true로 만들고
종료하는 접근방법

확인하고자 하는 ID에 따라 수행 속도 차이 발생

최악의 경우 모든 자료 탐색

다른 접근 방법은

검색 Key를 통해 저장된 위치를 바로 찾을 수 있는 데이터의 구성방법이 있다면?



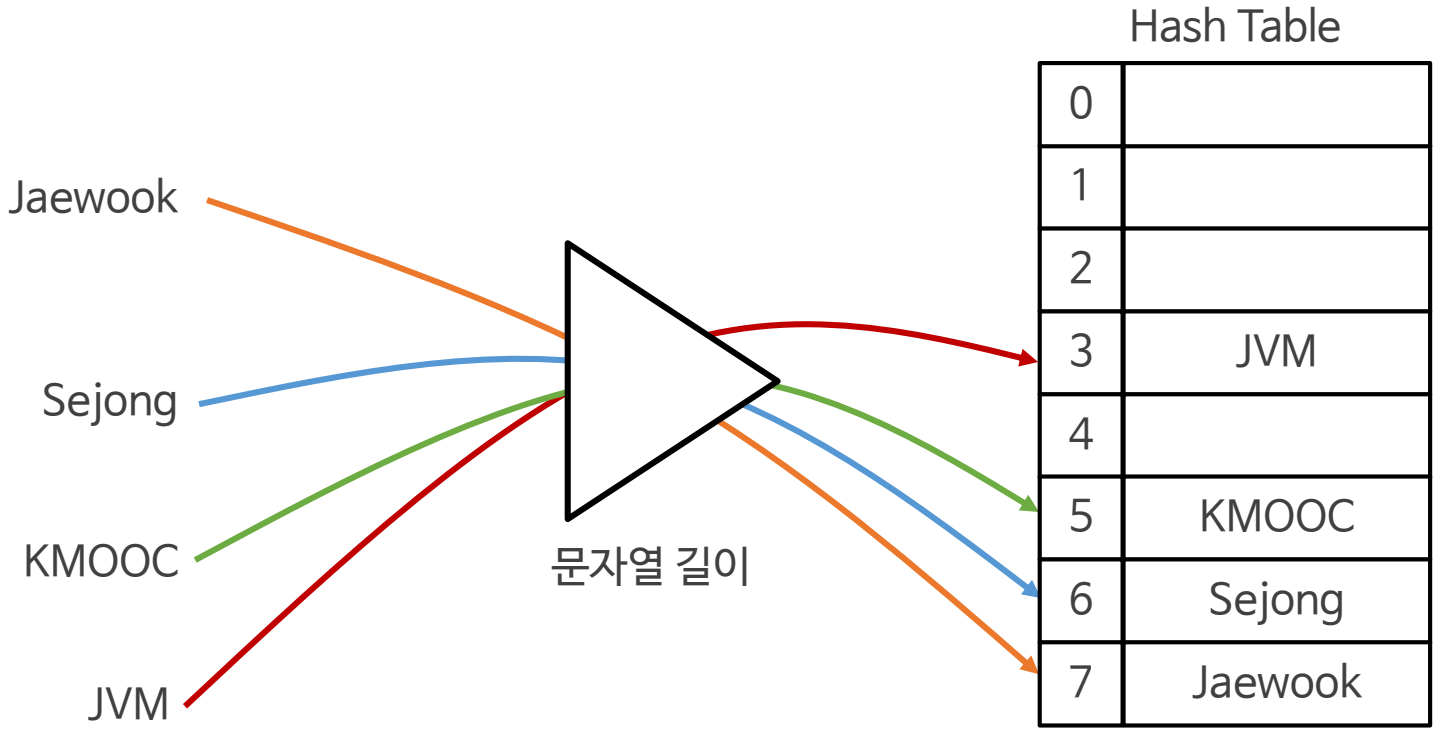
Data

Hash Table이라는
배열에 저장되어 있음

Hash

검색 키에 대한 데이터
저장 위치를 반환함

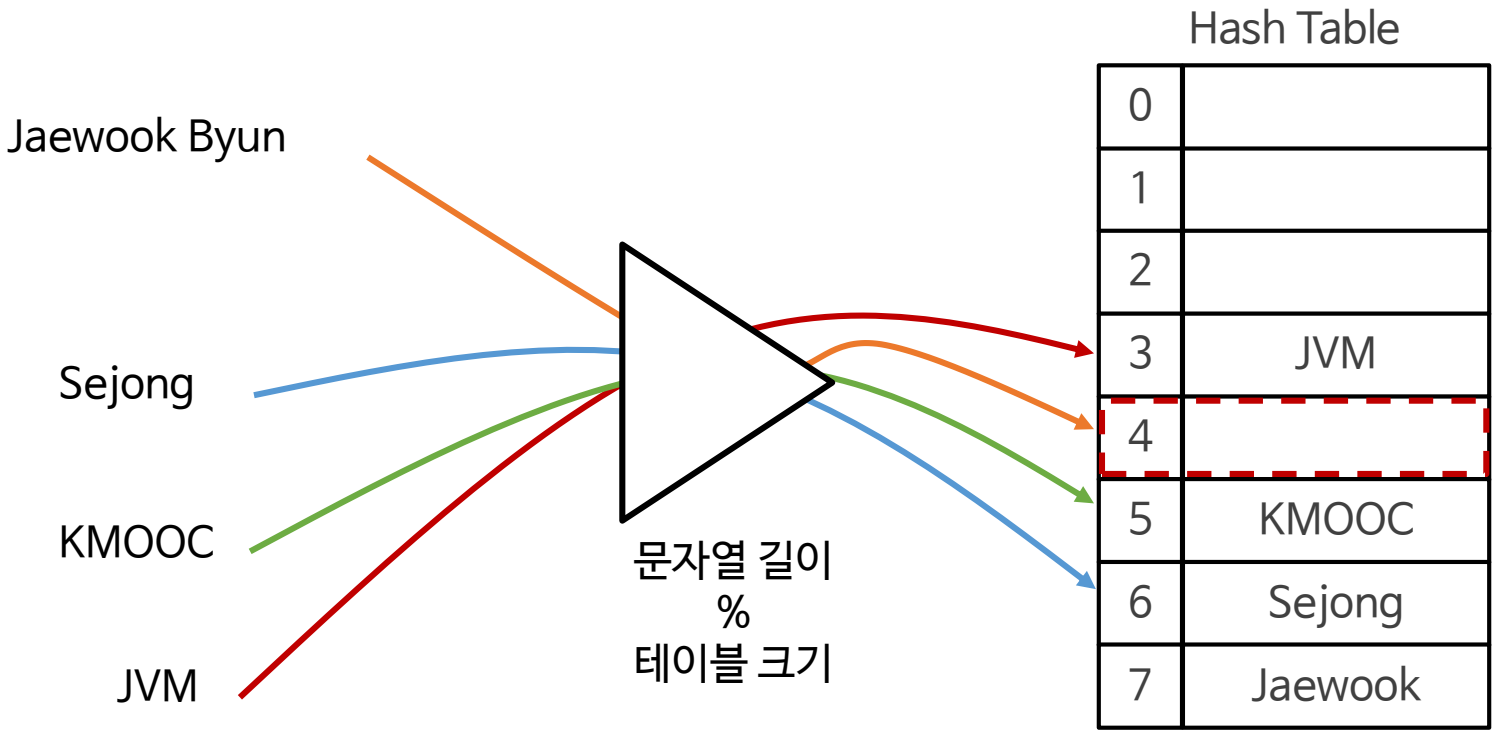
이름을 저장하고 있는 Hash Table



효율적이나 많은 고려가 필요함

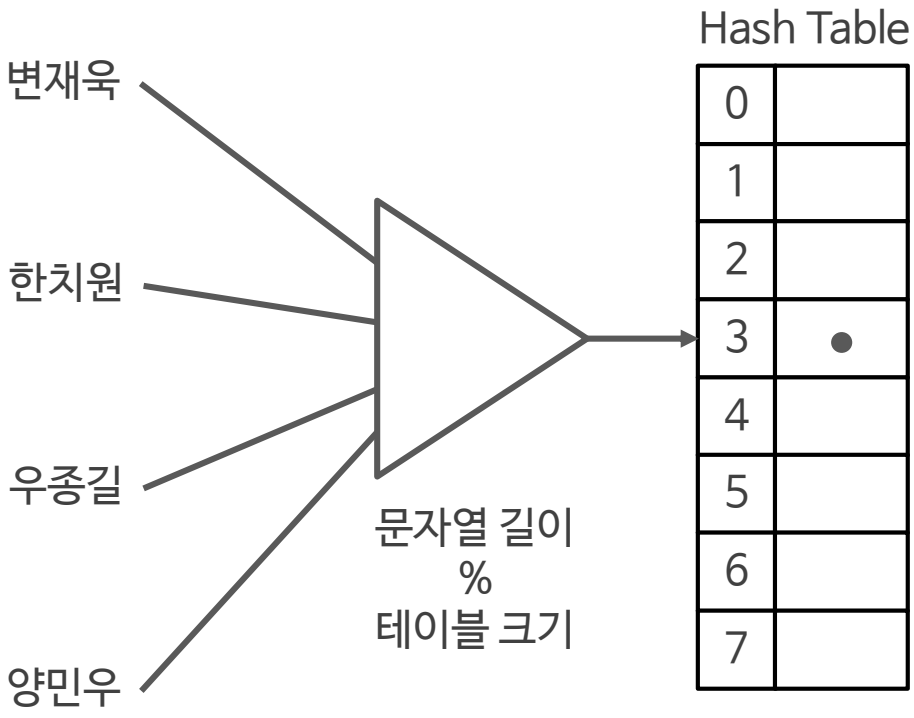
Hash 결과가 Hash Table의 크기를 초과한다면?

나머지 연산



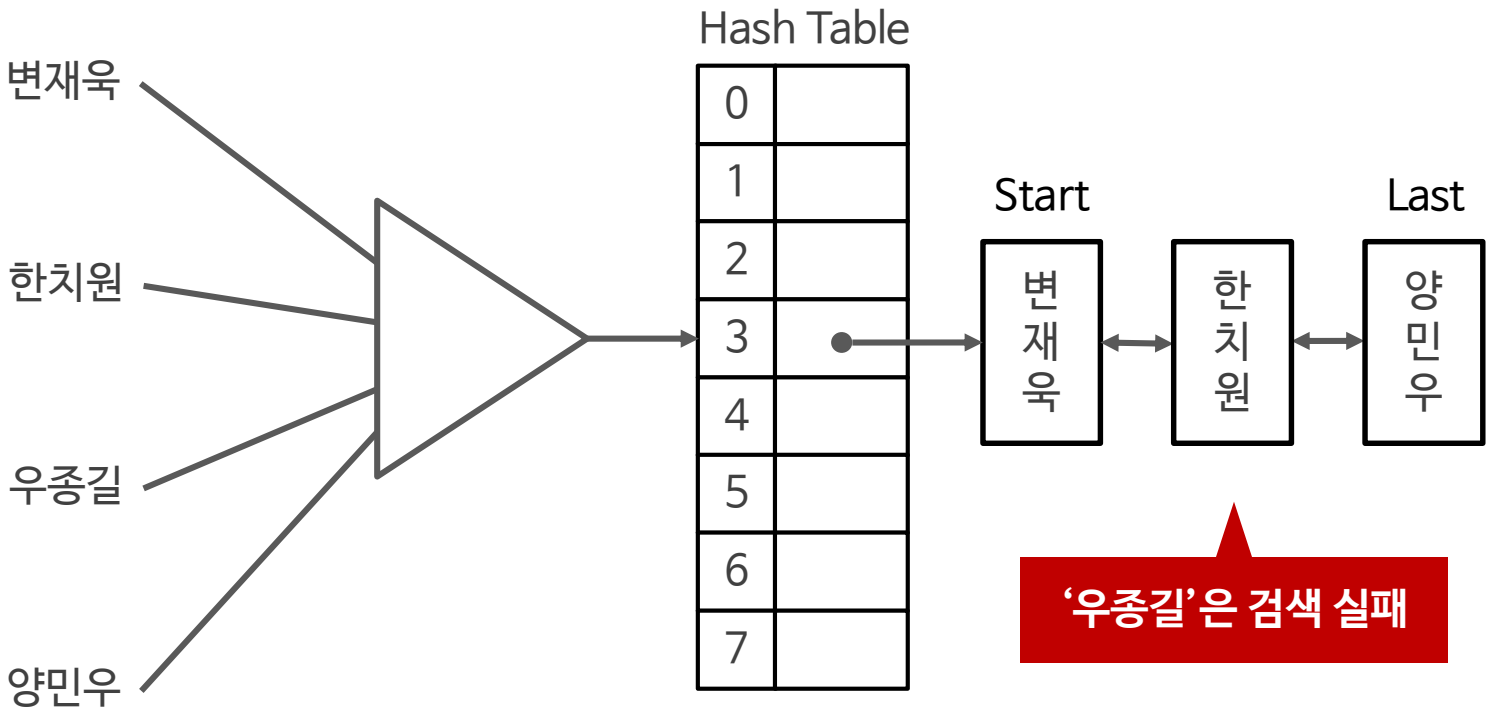
Hash 결과가 충돌(Collision)된다면?

리스트 유지(Close Addressing)

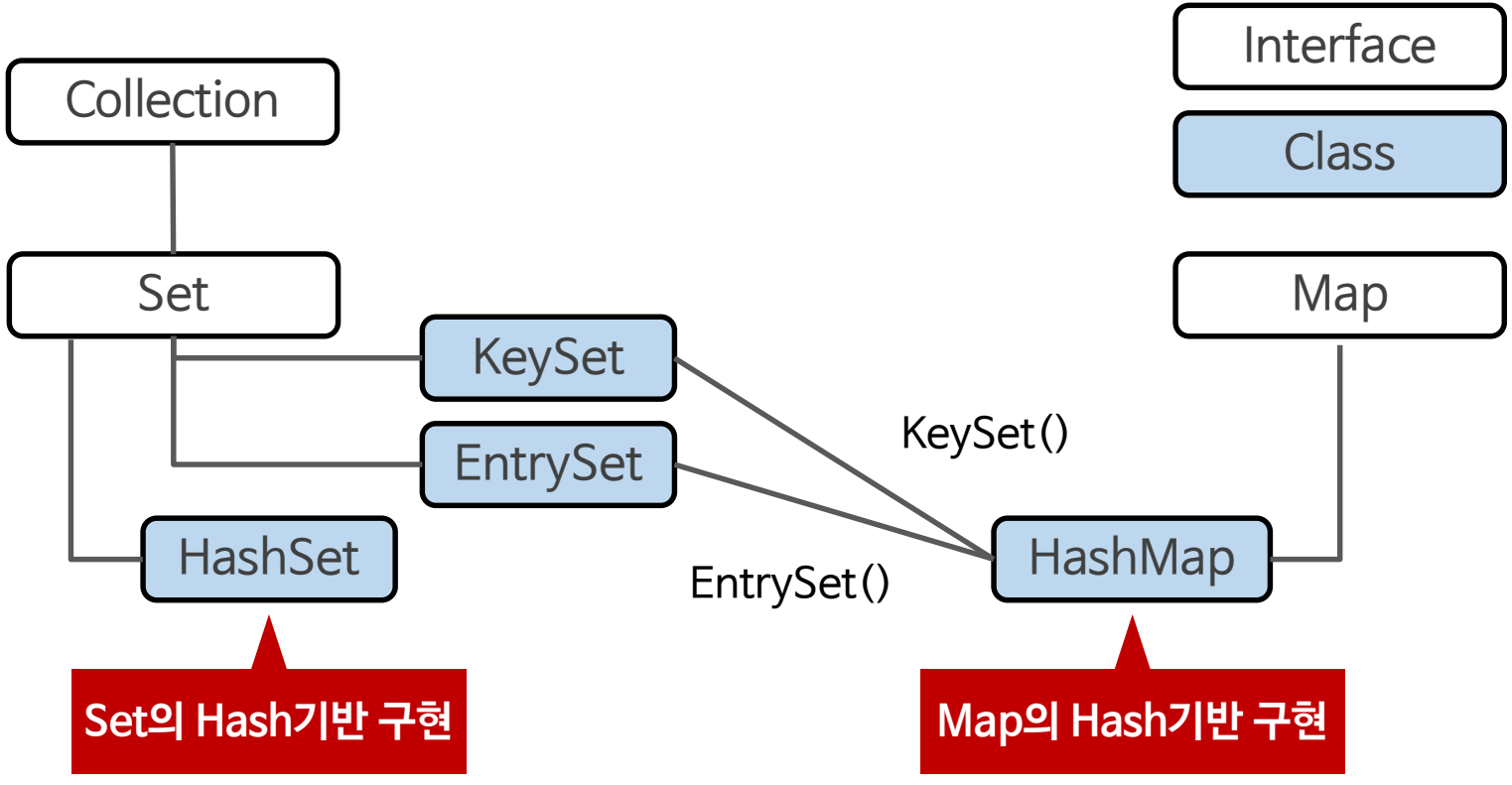


Hash 결과가 충돌(Collision)된다면?

리스트 유지(Close Addressing)



Java에서의 Hash 기반 구현

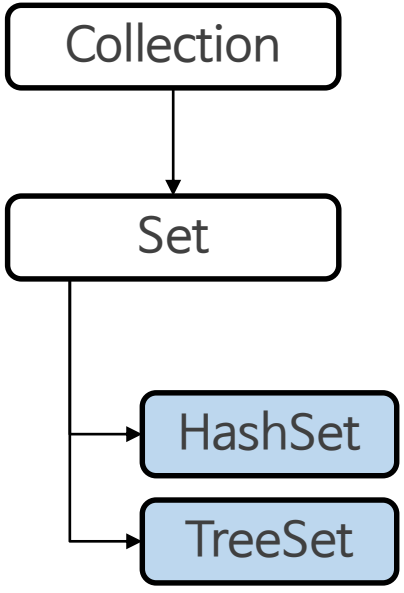


SET

순서가 없고 중복을 허용하지 않는 Collection

수학적 집합

SET

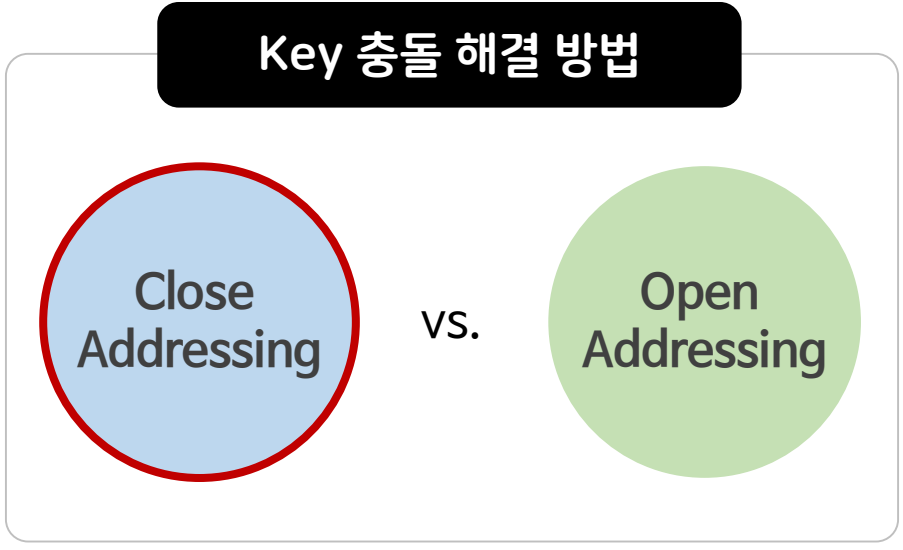


	Return Type	Method	Description
META	boolean	isEmpty()	Set이 비어 있는지 확인
	int	size()	Set의 크기를 반환
C	boolean	add(E e)	Set에 새로운 instance를 삽입
R	boolean	contains(Object o)	Set에 o라는 instance가 있는지 확인
D	boolean	remove(Object o)	Set에 o라는 instance가 있다면 삭제
T	Iterator<E>	iterator()	Set을 순회할 수 있는 iterator를 반환
	void	clear()	Set을 비움
	<T> T[]	toArray(T[] a)	Set을 T타입의 배열에 담음
집합연산	boolean	containsAll(Collection<?> c)	Set이 Collection c의 instance들을 모두 갖고있는지 확인 (부분집합)
	boolean	addAll(Collection<?> c)	Set에 Collection c의 instance들을 모두 추가함 (합집합)
	boolean	retainAll(Collection<?> c)	Set에서 Collection c의 instance인 것만을 남김 (교집합)
	boolean	removeAll(Collection<?> c)	Set에서 Collection c의 instance인 것은 지움 (차집합)
	Stream<E>	stream()	Set에 대한 Stream을 반환

add가 중복된 요소를 저장하지 않는 등 다르게 동작

HashSet

Set의 Hash기반 구현



HashSet

Hash는 상수시간의 빠른 탐색을 위해 존재

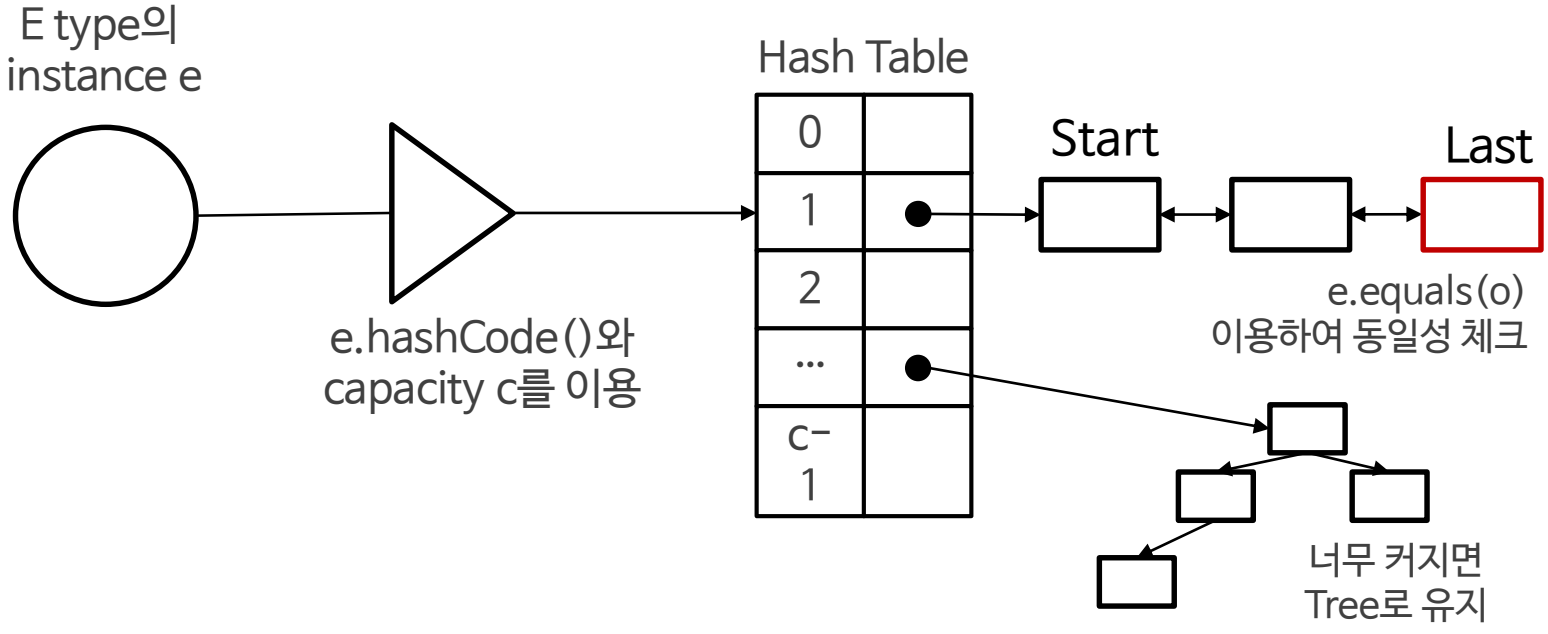
Hash Table의 크기가 작으면

자료 탐색의 시간 증가

HashSet

빠른 탐색을 위한 Hash Table 재구성
(Rehashing)

Hash Table의 크기와 임계 비율 활용



Remind

HashSet의 소개

HashSet의 개념

HashSet을 구성하는 Interface