세종대학교

Data Engineering

데이터엔지니어링

[해시 기반 데이터 엔지니어링(3)]
- MyHashSet 구현 및 활용 -

Set 인터페이스의 해시기반 자료구조 설계(1)

- 준비 및 생성자와 4가지 메소드를 통한 설계

학습내용

1. Set 인터페이스의 해시기반 자료구조 설계
2. Set 인터페이스의 해시기반 자료구조 구현

학습목표

- Set 인터페이스의 해시기반 자료구조를 설계할 수 있다.
- Set 인터페이스의 해시기반 자료구조를 구현할 수 있다.

HashSet

Set의 Hash 기반 구현

순서가 없고 중복을 허용하지 않는
빠른 탐색을 위한 수학적 집합의 Collection

	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
U	boolean	remove(Object o)	Set에 o라는 instance가 있다면 삭제
D	Iterator<E>	iterator()	Set을 순회할 수 있는 iterator를 반환
	void	clear()	Set을 비움
	<T> T[]	toArray(T[] a)	Set을 T타입의 배열에 담음
집합 연산	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

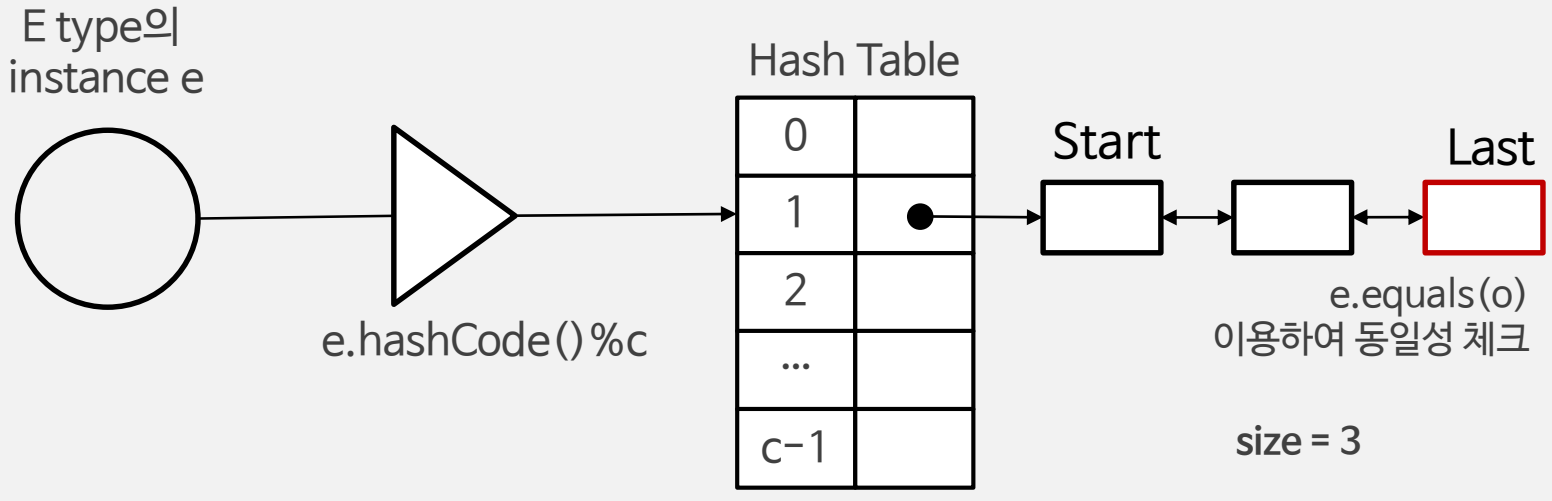
MyHashSet 설계 전략

Set<E> 구현

사용자가 HashTable 크기 capacity 지정 (디폴트 4096), rehashing 없음

E e는 Object.hashCode()%capacity의 LinkedList에 삽입됨

size를 추가적인 변수로 유지함



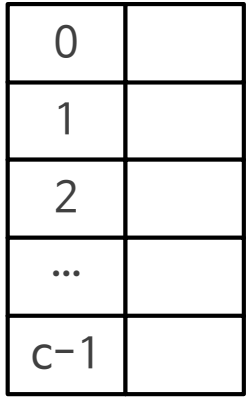
MyHashSet 구현 준비

Set〈E〉 인터페이스를 구현한 MyHashSet〈E〉 생성

LinkedList〈E〉[] : Hash Table

int size;

new LinkedList〈E〉[c]



size = 0

Size 계산을 위해
계속 Set을 순회하는 것은
비용이 큼

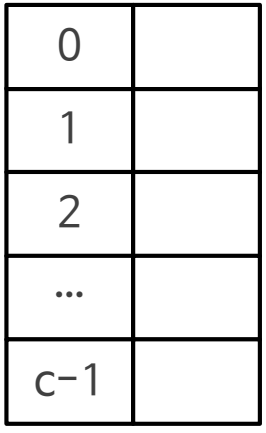
[MyHashSet 구현]

준비

생성자

Return Type	Method	Description
생성자	MyHashSet()	빈 MyHashSet을 Hash Table의 초기 크기 4096으로 설정해 생성
생성자	MyHashSet (int initialCapacity)	빈 MyHashSet을 Hash Table의 초기 크기를 설정해 생성

new LinkedList<E> [c]



size = 0

[MyHashSet 구현]

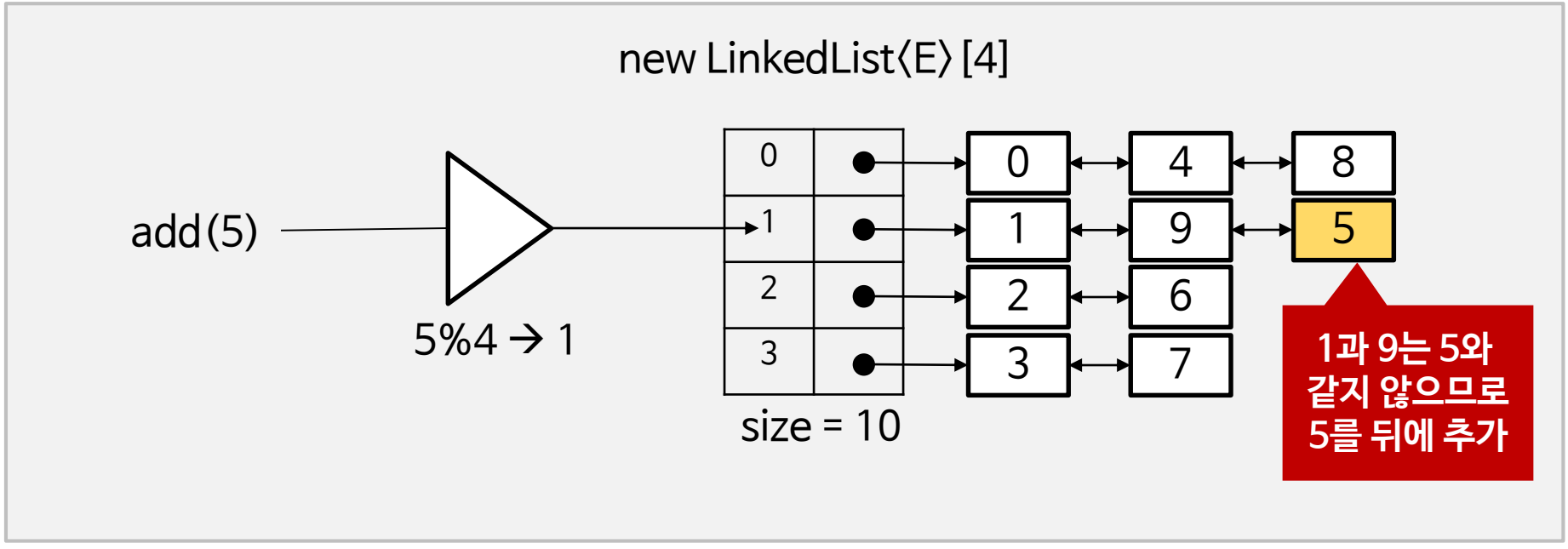
생성자

isEmpty, size, add(E e)

Return Type	Method	Description
boolean	isEmpty()	Set이 비어 있는지 확인
int	size()	Set의 크기를 반환
boolean	add(E e)	Set에 새로운 instance를 삽입

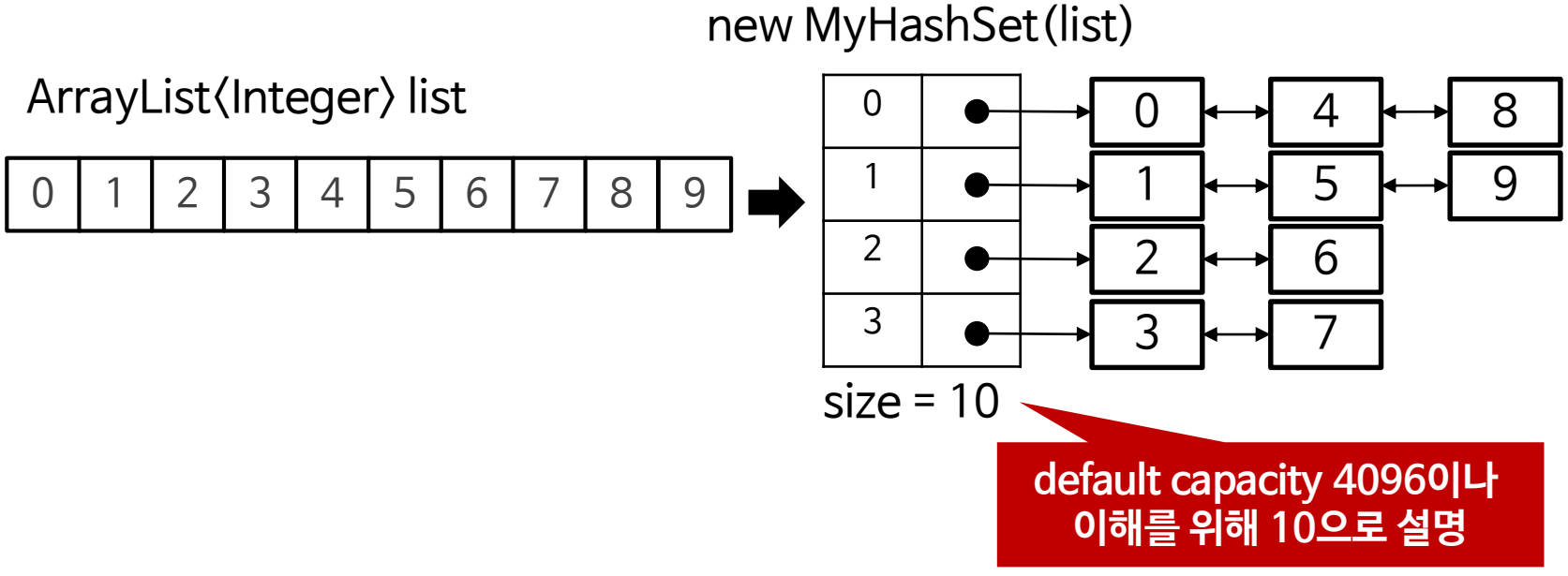
isEmpty, size, add(E e)

추가하려는 Key의 해시코드와 capacity를 이용하여 LinkedList를 찾기 → contains를 이용하여 기존에 존재하는 값인지 확인 → add를 통해 맨 뒤에 삽입



isEmpty, size, add(E e)

Return Type	Method	Description
생성자	MyHashSet(Collection<? extends E> c)	MyHashSet을 Collection c의 요소를 가져와 생성



[MyHashSet 구현]

isEmpty

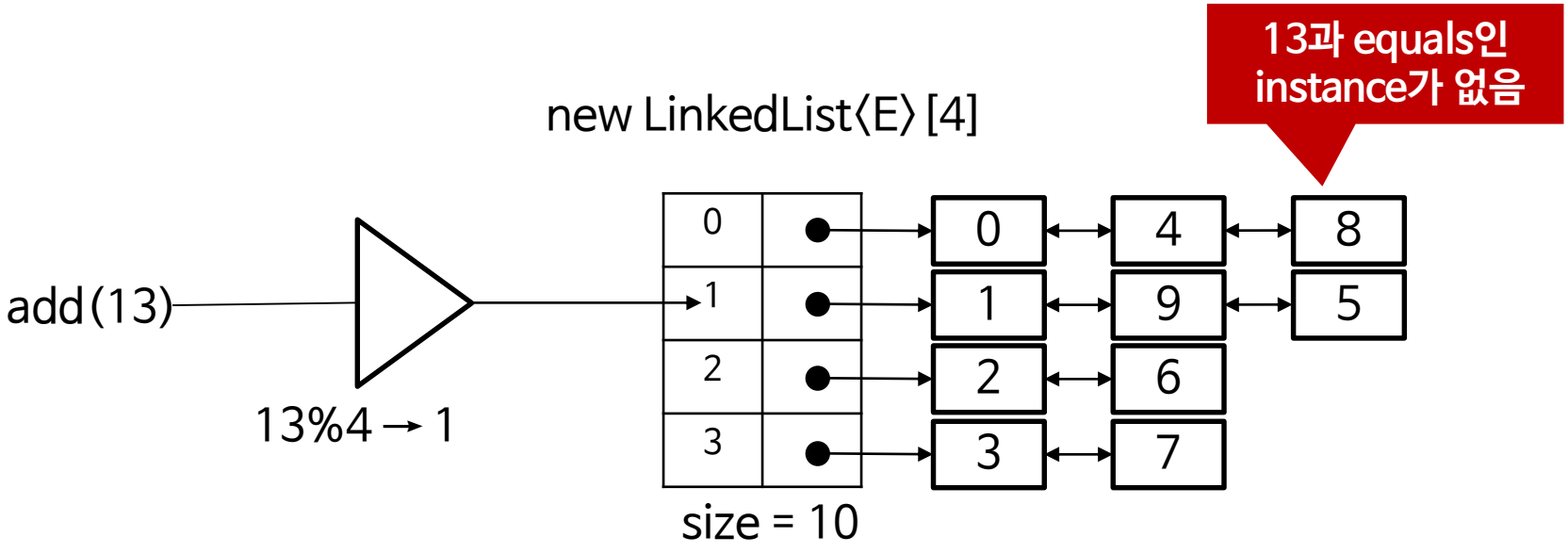
size

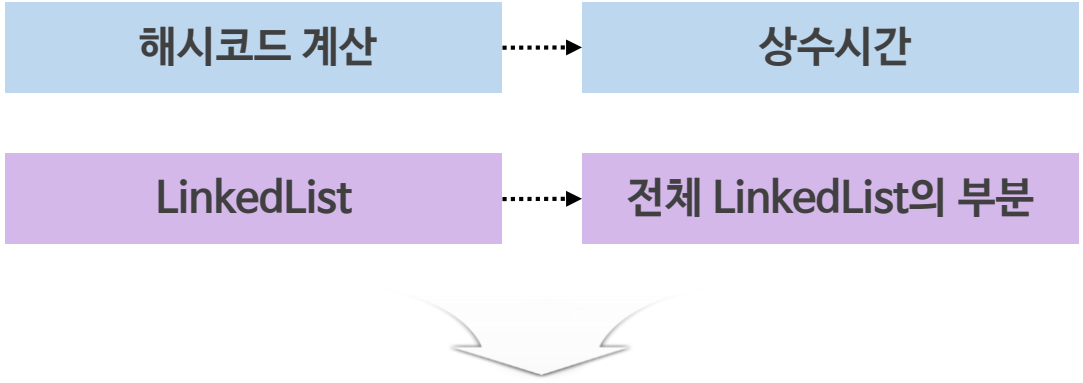
add

contains

Return Type	Method	Description
boolean	Contains (Object o)	Set에 o라는 instance가 있는지 확인

contains


13과 equals인
instance가 없음
new LinkedList<E> [4]
add(13)
13%4 → 1
0
1
2
3
0
4
8
1
9
5
2
6
3
7
size = 10


해시코드 계산
상수시간
LinkedList
전체 LinkedList의 부분
빠른 탐색 가능

[MyHashSet 구현]

contains

Remind

Set 인터페이스 해시기반 자료구조 설계 준비

Return Type	Method	Description
생성자	MyHashSet()	빈 MyHashSet을 Hash Table의 초기 크기 4096으로 설정해 생성
생성자	MyHashSet (int initialCapacity)	빈 MyHashSet을 Hash Table의 초기 크기를 설정해 생성
boolean	isEmpty()	Set이 비어 있는지 확인
int	size()	Set의 크기를 반환
boolean	add(E e)	Set에 새로운 instance를 삽입
boolean	Contains(Object o)	Set에 o라는 instance가 있는지 확인