



Set 인터페이스의 해시 기반 자료구조 설계 (2)

- 4가지 메소드를 통한 설계

학습내용

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

학습목표

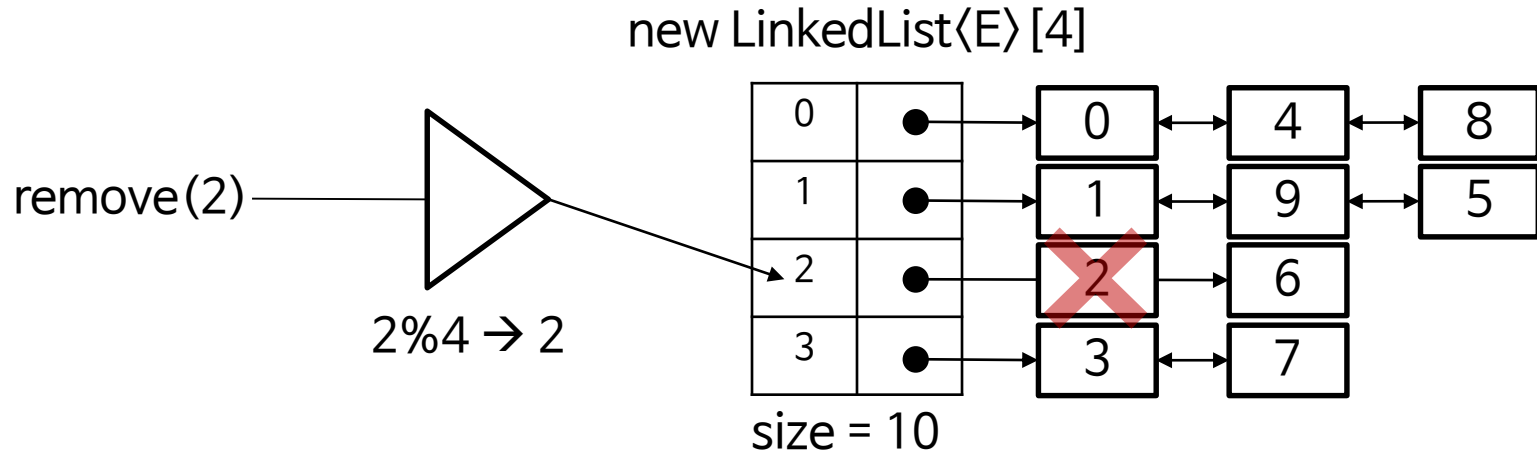
- Set 인터페이스의 해시기반 자료구조를 설계할 수 있다.
- Set 인터페이스의 해시기반 자료구조를 구현할 수 있다.

remove

Return Type	Method	Description
boolean	Remove (Object o)	Set에 o라는 instance가 있다면 삭제

remove

hashCode에 대한 LinkedList에 **remove(Object o)** 수행



LinkedList에서 2제거

[MyHashSet 구현]

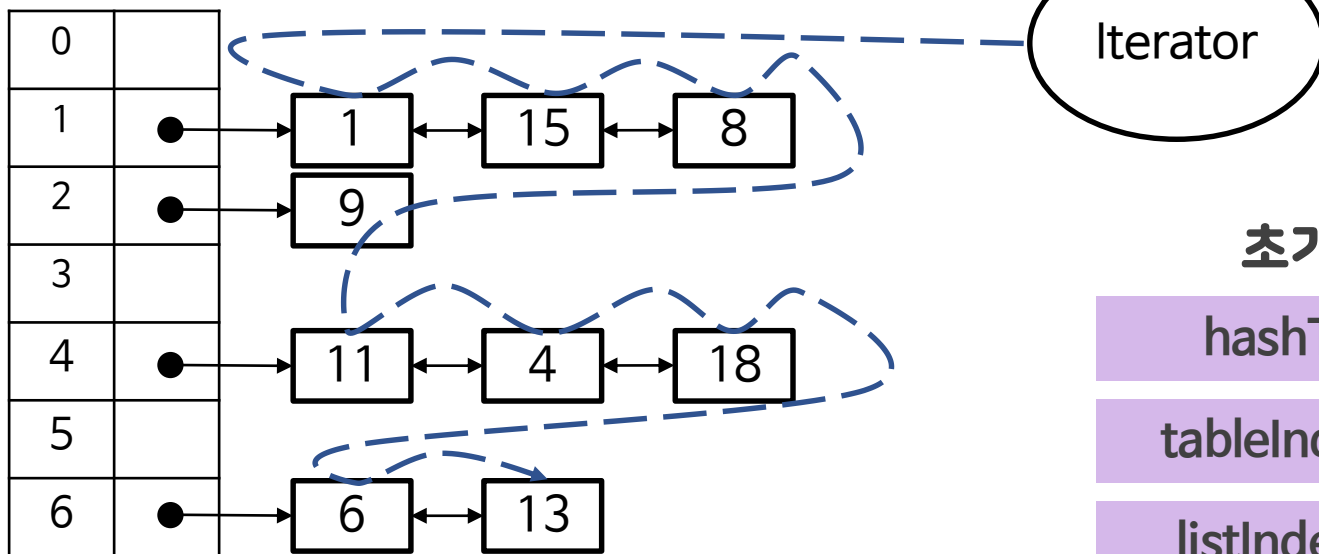
remove

iterator

Return Type	Method	Description
Iterator<E>	iterator()	Set을 순회할 수 있는 iterator를 반환

iterator

LinkedList<E> [7]



초기화

hashTable

tableIndex = 0

listIndex = 0;

LinkedList의 크기가 커질 경우, 비효율적으로 동작할 수 있음

[MyHashSet 구현]

Iterator

clear

Return Type	Method	Description
void	clear()	Set을 비움

clear

```
hashTable = new LinkedList[capacity];
```

LinkedList<E> [7]

0	
1	
2	
3	
4	
5	
6	

[MyHashSet 구현]

clear

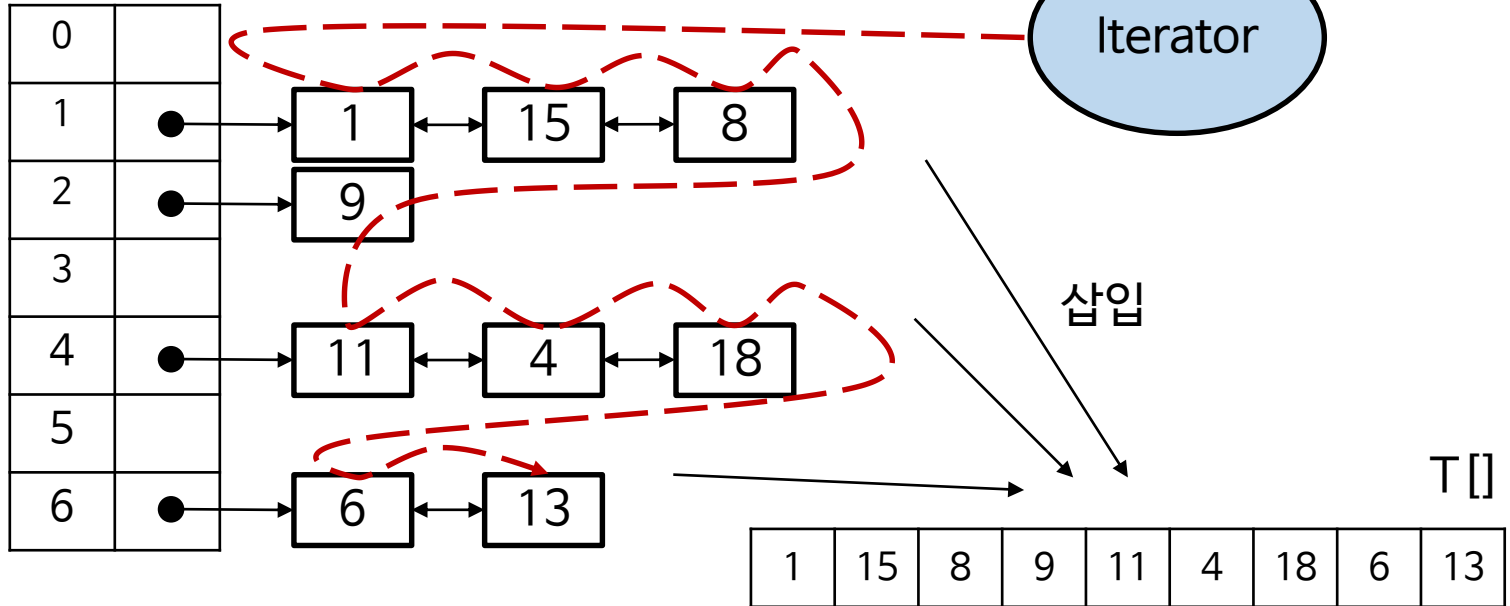
toArray

Return Type	Method	Description
〈T〉 T []	toArray(T [] a)	Set을 T타입의 배열에 담음

toArray

T[size] 생성 후 **iterator**를 이용하여 채워 넣기

LinkedList<E> [7]



[MyHashSet 구현]

toArray

Remind

Set 인터페이스의 해시 기반 자료구조 구현

Return Type	Method	Description
boolean	Remove(Object o)	Set에 o라는 instance가 있다면 삭제
Iterator<E>	iterator()	Set을 순회할 수 있는 iterator를 반환
void	clear()	Set을 비움
<T> T[]	toArray(T[] a)	Set을 T타입의 배열에 담음