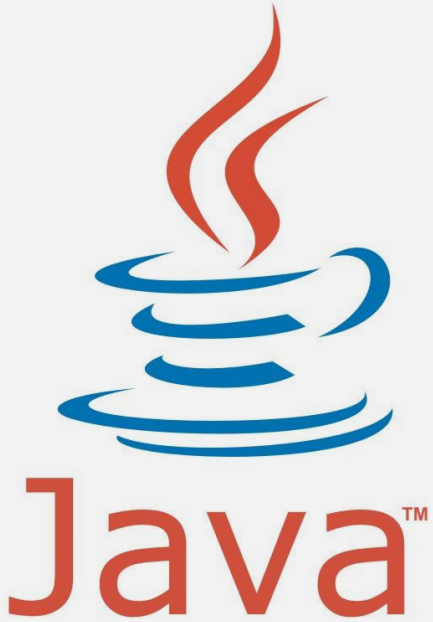
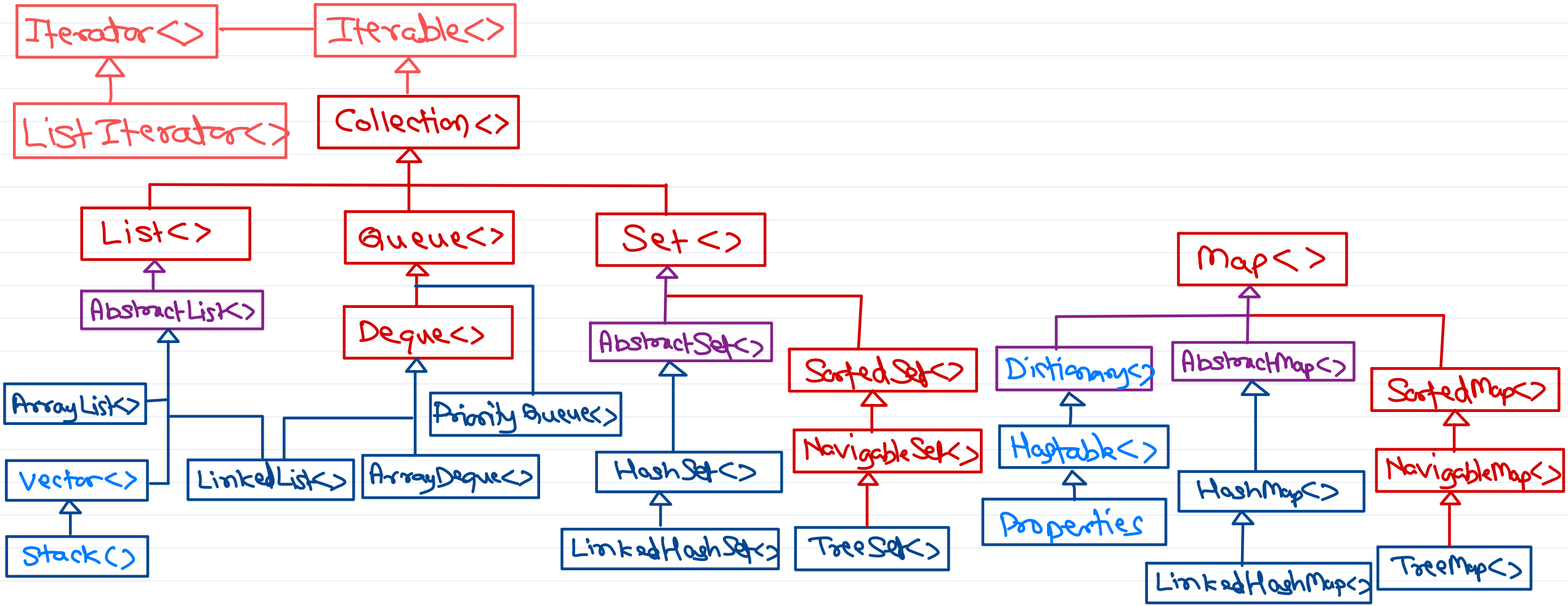


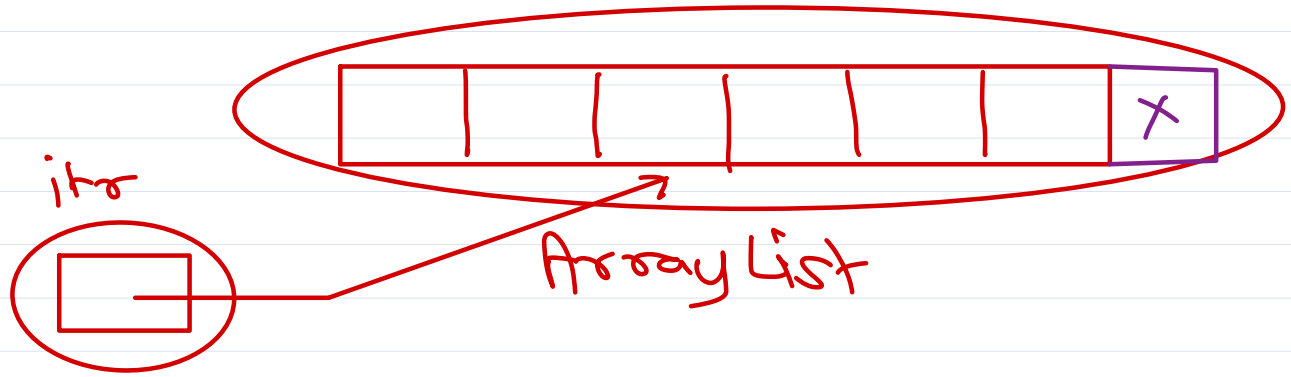
Core Java

Trainer: Nilesh Ghule

Java Collection Framework



Fail fast iterator



thread1

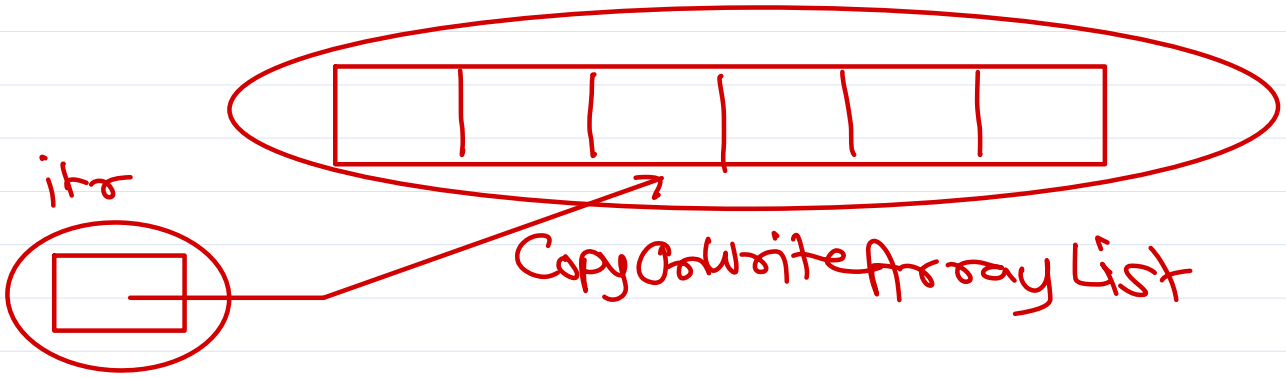
```
itr = list.iterator();
while(itr.hasNext()) {
    e = itr.next();
    Sysout(e);
}
```

thread2

```
list.add("x");
```

ConcurrentModificationException

Fail safe iterator



thread1

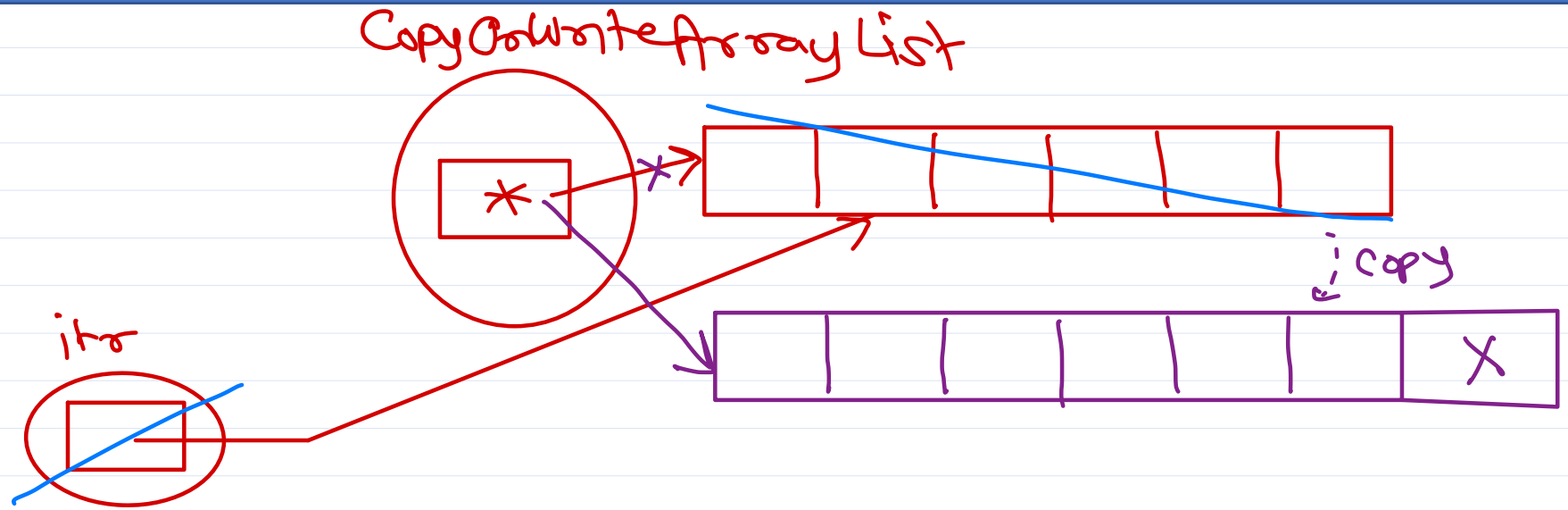
```
itr = list.iterator();
while(itr.hasNext()) {
    e = itr.next();
    Sysout(e);
}
```

thread2

```
list.add("x");
```

NO
ConcurrentModificationException

Fail safe iterator

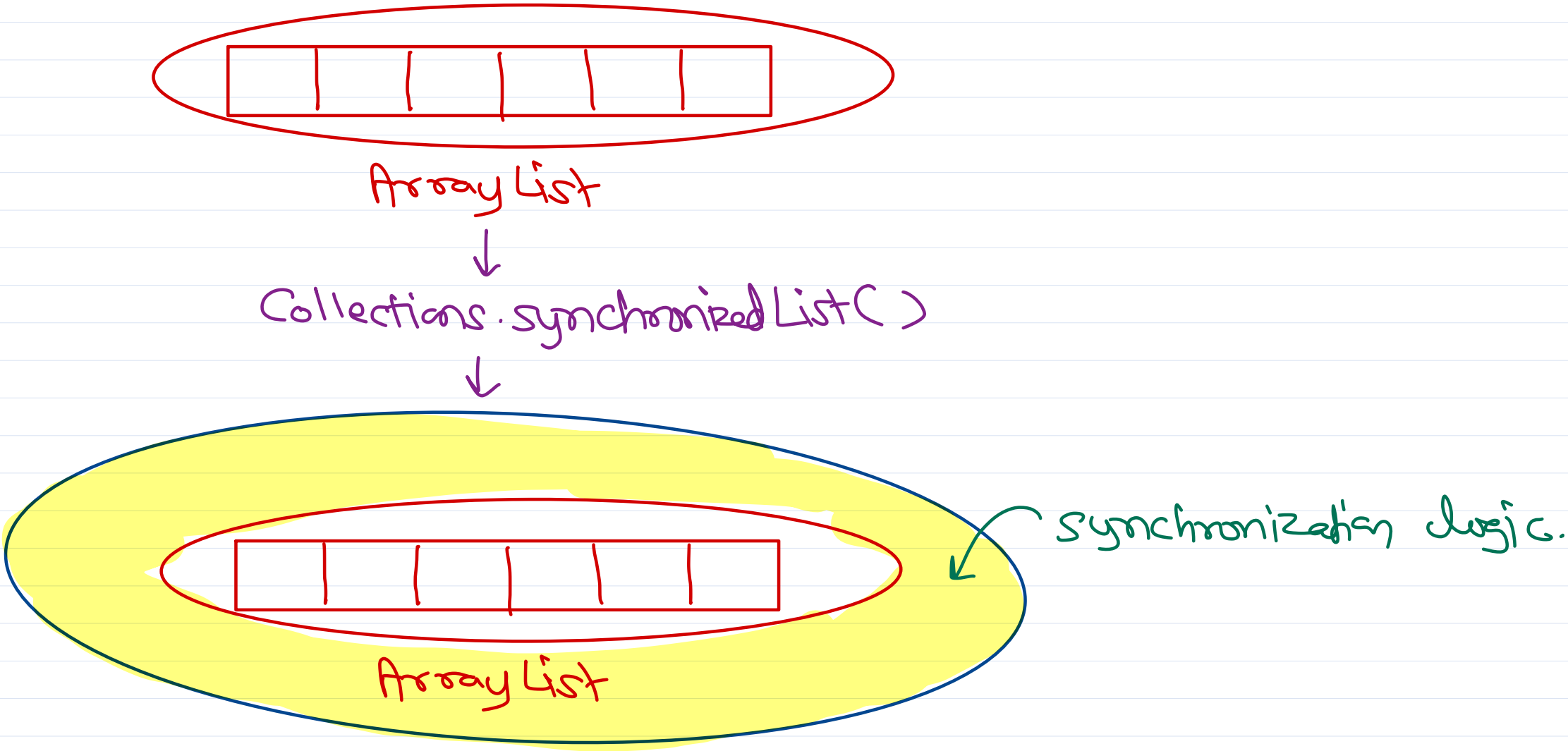


thread1

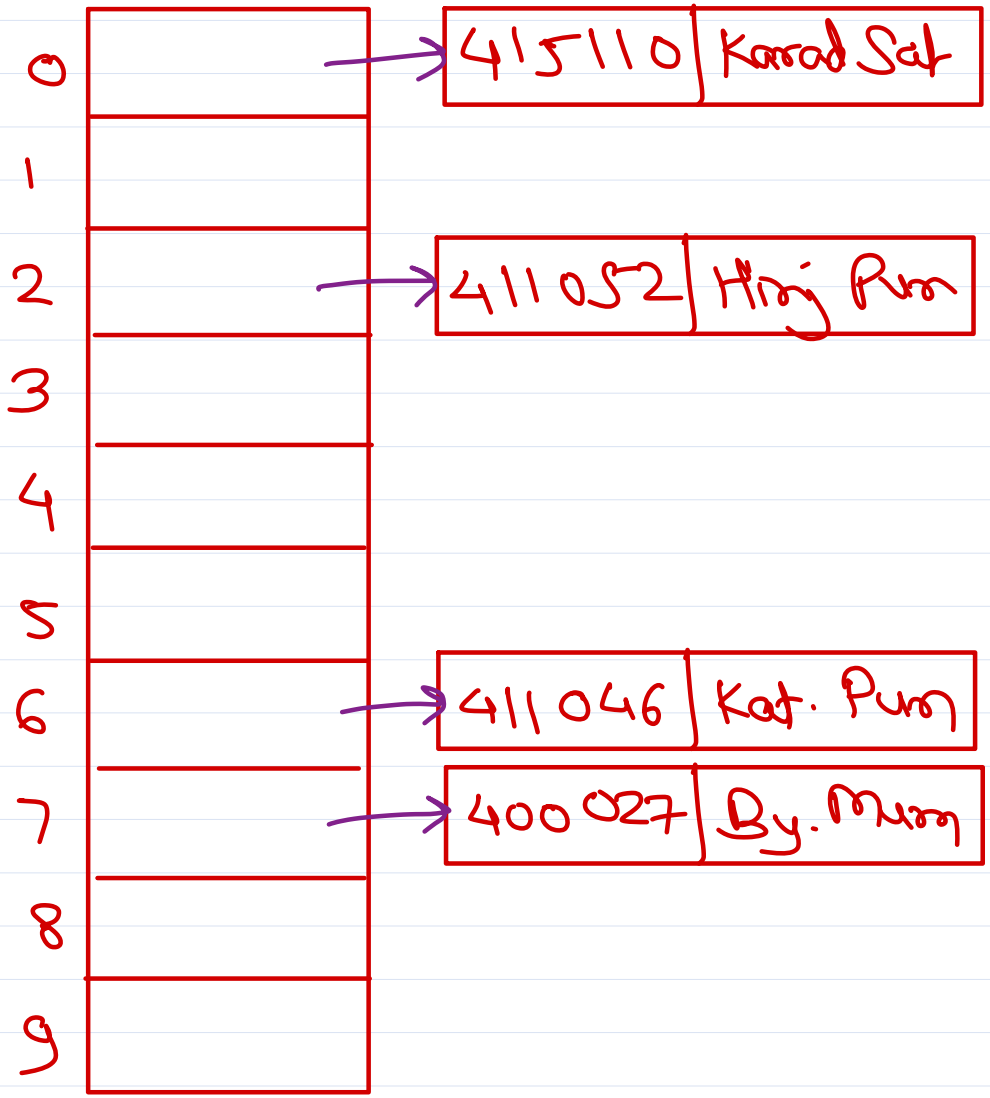
```
itr = list.iterator();
while(itr.hasNext()) {
    e = itr.next();
    Sysout(e);
}
```

thread2

```
list.add("x");
```


ArrayList
Collections.synchronizedList()
ArrayList
synchronization logic.

Hash table - basics



$h(k) = pin \% 10$

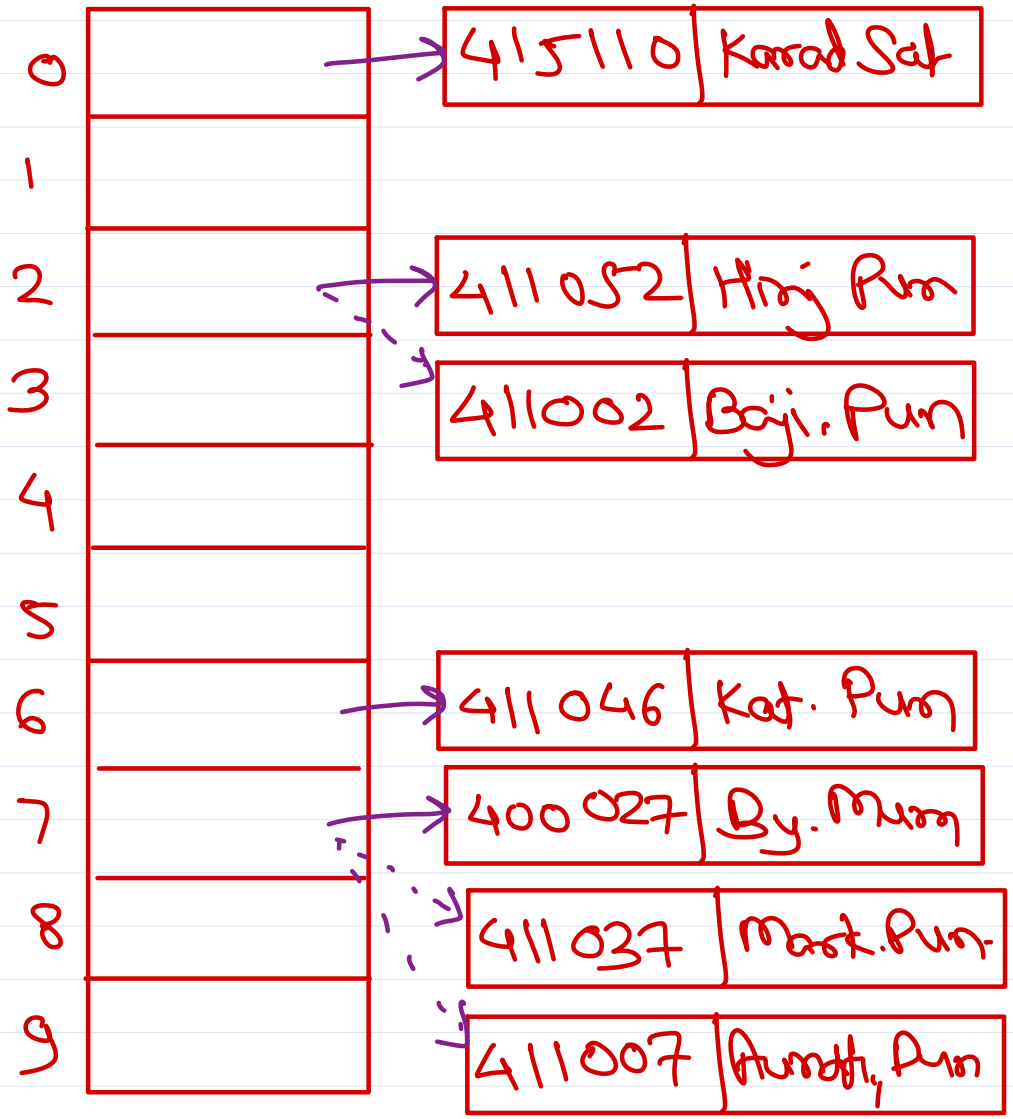
Hash fn is math fn of key that yields slot (index) of the table (array) in which elem is to be stored / searched.

Ideal hash fn is one that gives diff slot of diff key.

Hash fns are always consistent i.e. Same key always result in same slot.

Duplicate keys are not allowed in hashtable.

Hash table - collisions



$h(k) = pin \% 10$

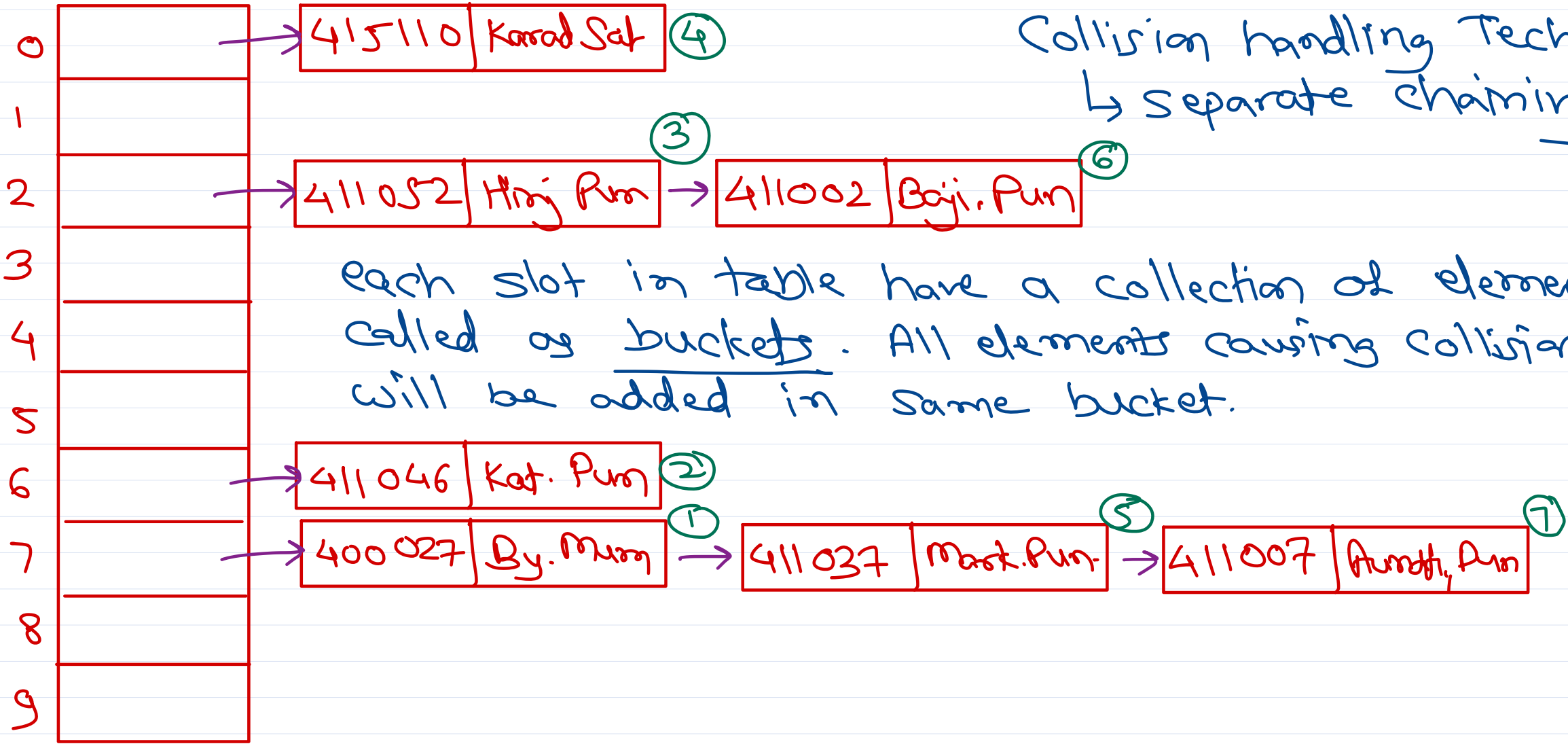
If diff keys results in same slot of the table - by hash fn - then it is called as collision.

Collision handling Techniques.

- open addressing
- separate chaining

Hash table - Separate chaining

$h(k) = pin \% 10$

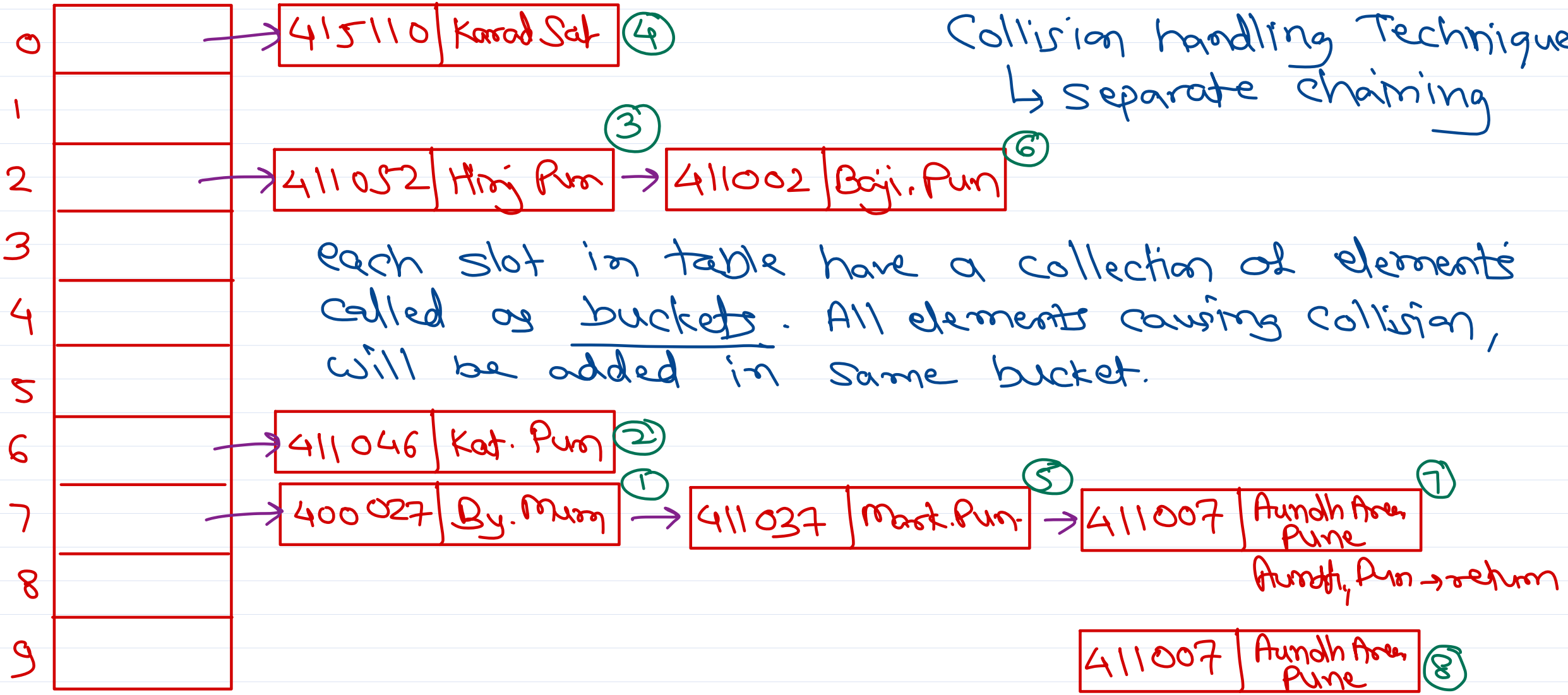


Collision handling Techniques
↳ separate chaining

Each slot in table have a collection of elements called as buckets. All elements causing collision, will be added in same bucket.

Hash table - put() - duplicate key

$h(k) = pin \% 10$



Collision handling Techniques
↳ separate chaining

Each slot in table have a collection of elements called as buckets. All elements causing collision, will be added in same bucket.

Aundh, Pun → return

Hash table - Load factor

$$\text{Load factor} = \frac{\text{Number of entries}}{\text{Number of buckets}}$$

e.g.

Num of entries	Num of buckets	Load factor
7	10	0.7
10	10	1
15	10	1.5

Sets vs Maps

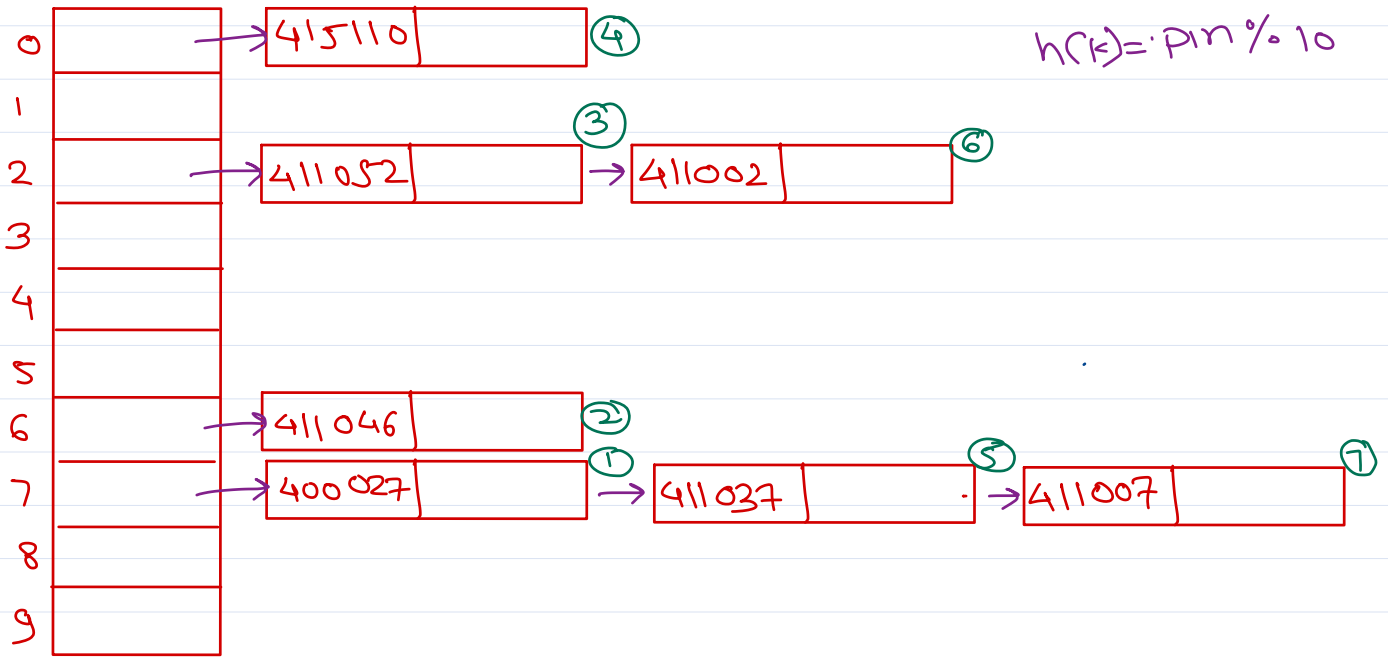
HashSet<K> = HashMap<K, null>

LinkedHashSet<K> = LinkedHashMap<K, null>

TreeSet<K> = TreeMap<K, null>

```
Set<Integer> s = new HashSet<>();
```

```
s.add(400027);
s.add(411046);
s.add(411052);
s.add(415110);
  :
```



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>