

name	add(val)	contains(val)	remove(val)
ArrayList	$O(1)$	$O(n)$	$O(n)$
LinkedList	$O(1)$	$O(n)$	$O(n)$
TreeSet	$O(\lg n)$	$O(\lg n)$	$O(\lg n)$
HashSet	$O(1)$	$O(1)$	$O(1)$

... where  $n$  is the size() of the data structure

"fast random access"

int n = arr[0]

arr[10000]

value 42 → index 2

[10, 21, 42, 73, ...]

contains(42) // false

add(42)

contains(42) // true

# Hash Function

"function or method that takes a value and returns an index"

```
public int hash(int value) {  
    return value % table.length;  
}
```

hash table

	11			24 104			7		
0	1	2	3	4	5	6	7	8	9

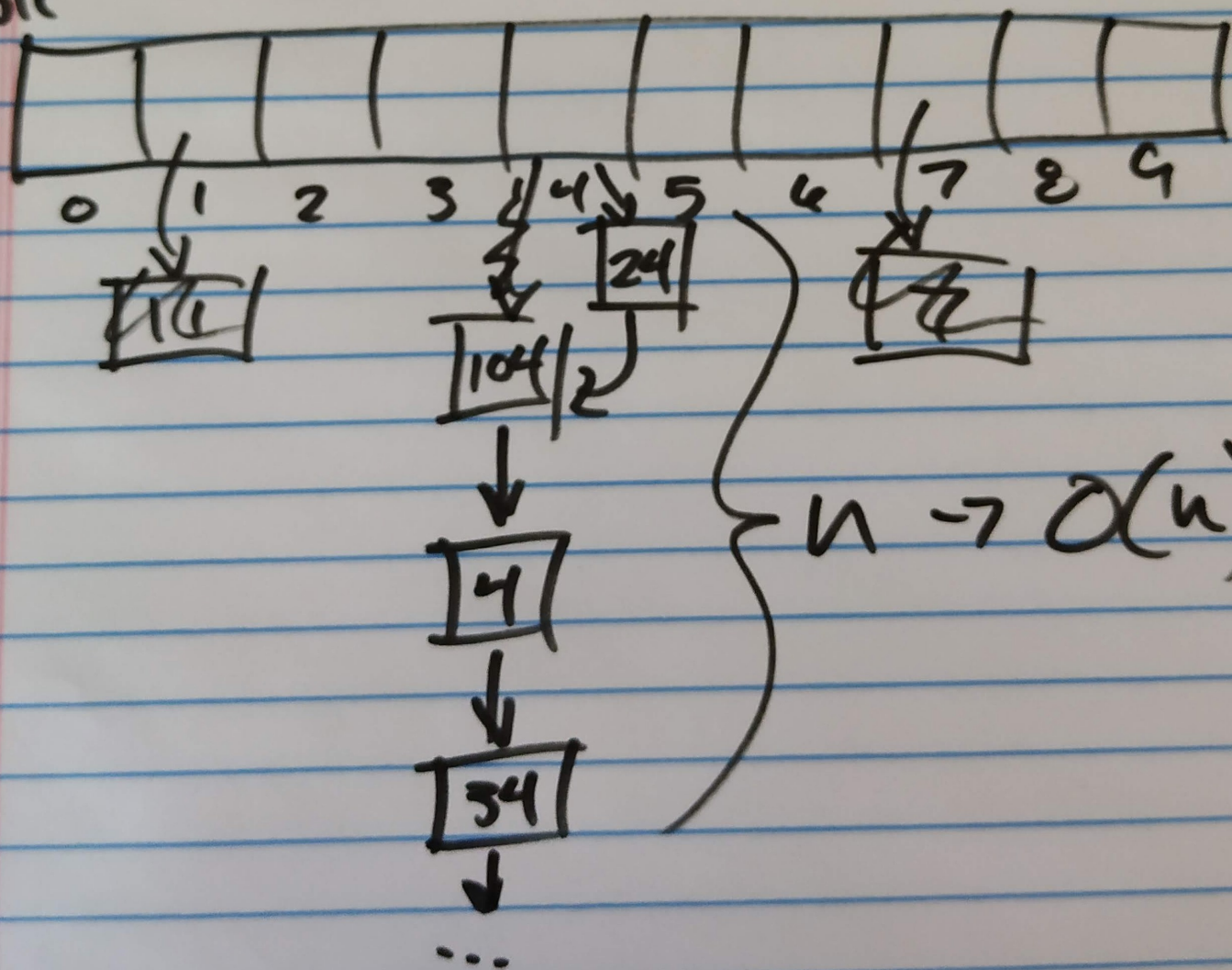
```
add(7)  
add(11)  
add(104)  
contains(104) // true  
add(24)  
contains(104) // false
```

# Collision Resolution

"what you do when 2 values hash to the same index"

## Separate Chaining

table



.hashCode()