

# Data Structures and Algorithms

## Hashing and Collisions

CS 225  
G Carl Evans

April 18, 2025



Department of Computer Science

# A Hash Table based Dictionary

## Client Code:

```
1 Dictionary<KeyType, ValueType> d;  
2 d[k] = v;
```

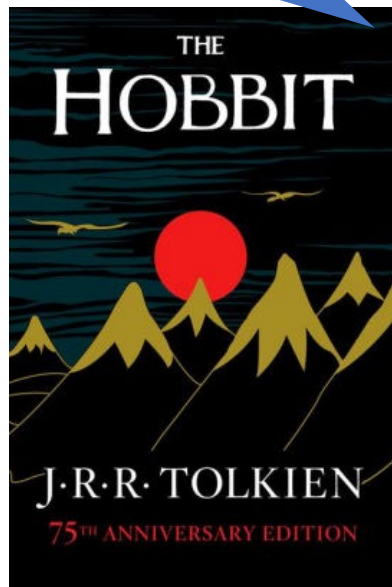
A **Hash Table** consists of three things:

1. A hash function
2. A data storage structure
3. A method of addressing *hash collisions*

# Hash Collision

A **hash collision** occurs when multiple unique keys hash to the same value

J.R.R. Tolkien = 30!



Jim Truth = 30!



...	...
27	∅
28	∅
29	∅
30	???
31	∅
...	...



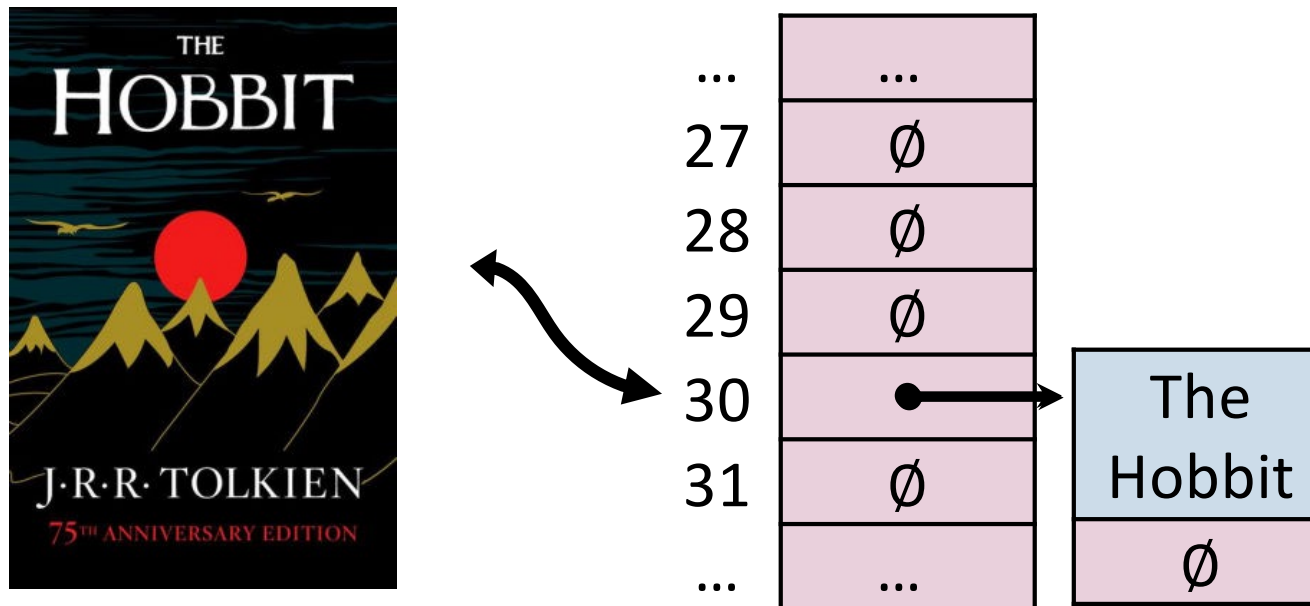
## Open vs Closed Hashing

Addressing hash collisions depends on your storage structure.

- **Open Hashing:** store  $k, v$  pairs externally
- **Closed Hashing:** store  $k, v$  pairs in the hash table

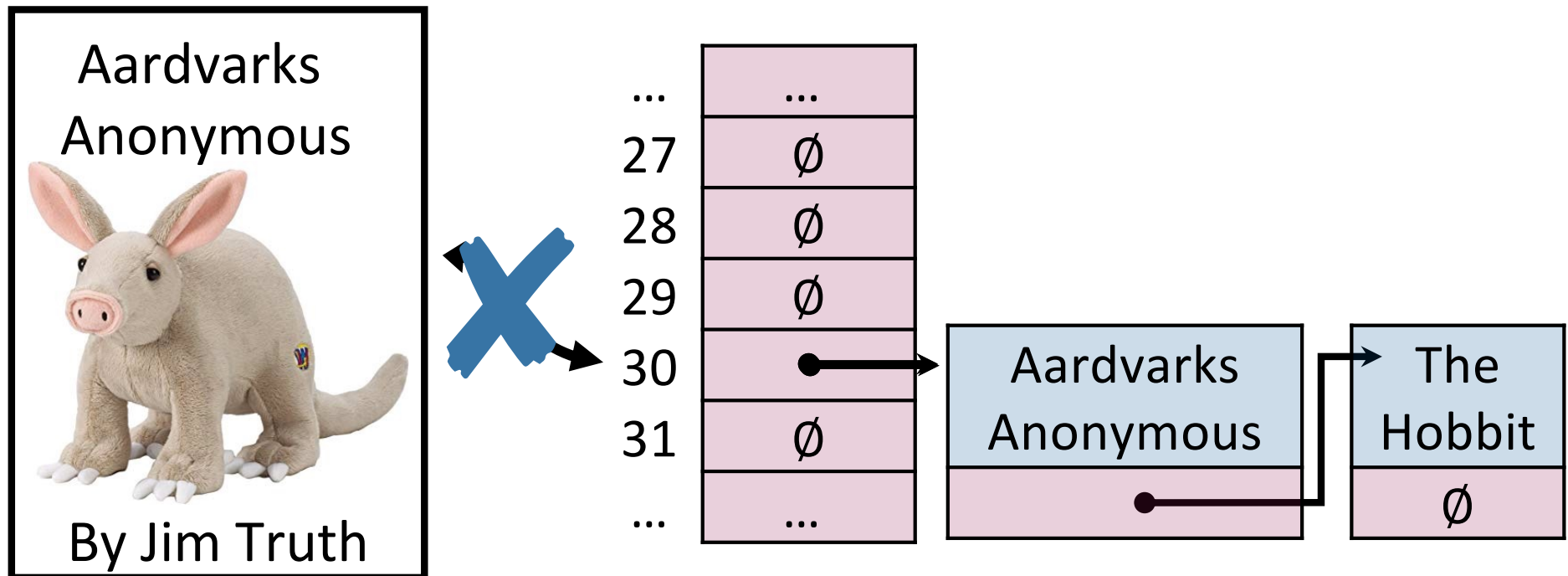
# Open Hashing

In an ***open hashing*** scheme, key-value pairs are stored externally (for example as a linked list).



# Hash Collisions (Open Hashing)

A **hash collision** in an open hashing scheme can be resolved by \_\_\_\_\_. This is called **separate chaining**.



# Insertion (Separate Chaining)

`_insert("Bob")`

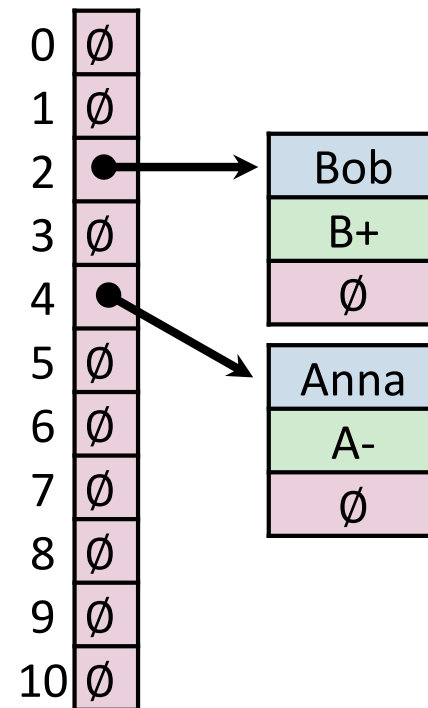
`_insert("Anna")`

Key	Value	Hash
Bob	B+	2
Anna	A-	4
Alice	A+	4
Betty	B	2
Brett	A-	2
Greg	A	0
Sue	B	7
Ali	B+	4
Laura	A	7
Lily	B+	7

0	∅
1	∅
2	∅
3	∅
4	∅
5	∅
6	∅
7	∅
8	∅
9	∅
10	∅

# Insertion (Separate Chaining) `_insert("Alice")`

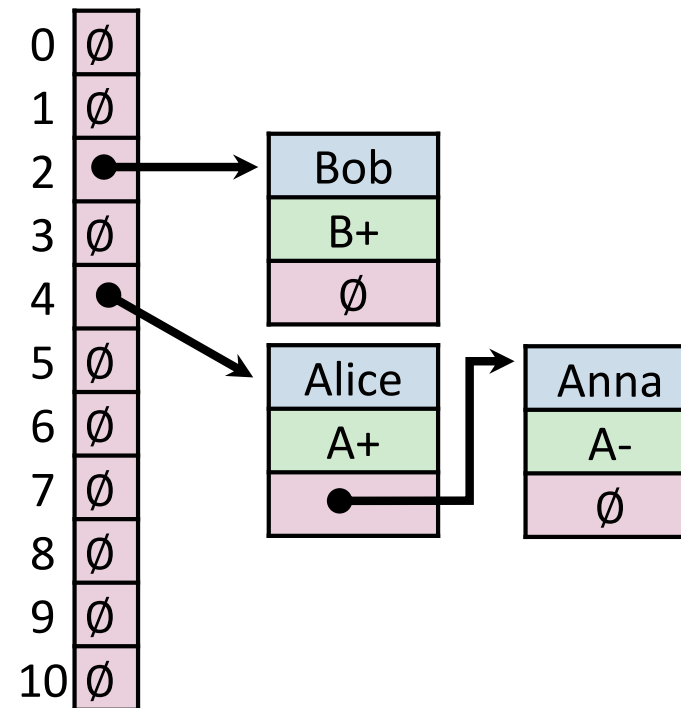
Key	Value	Hash
Bob	B+	2
Anna	A-	4
Alice	A+	4
Betty	B	2
Brett	A-	2
Greg	A	0
Sue	B	7
Ali	B+	4
Laura	A	7
Lily	B+	7





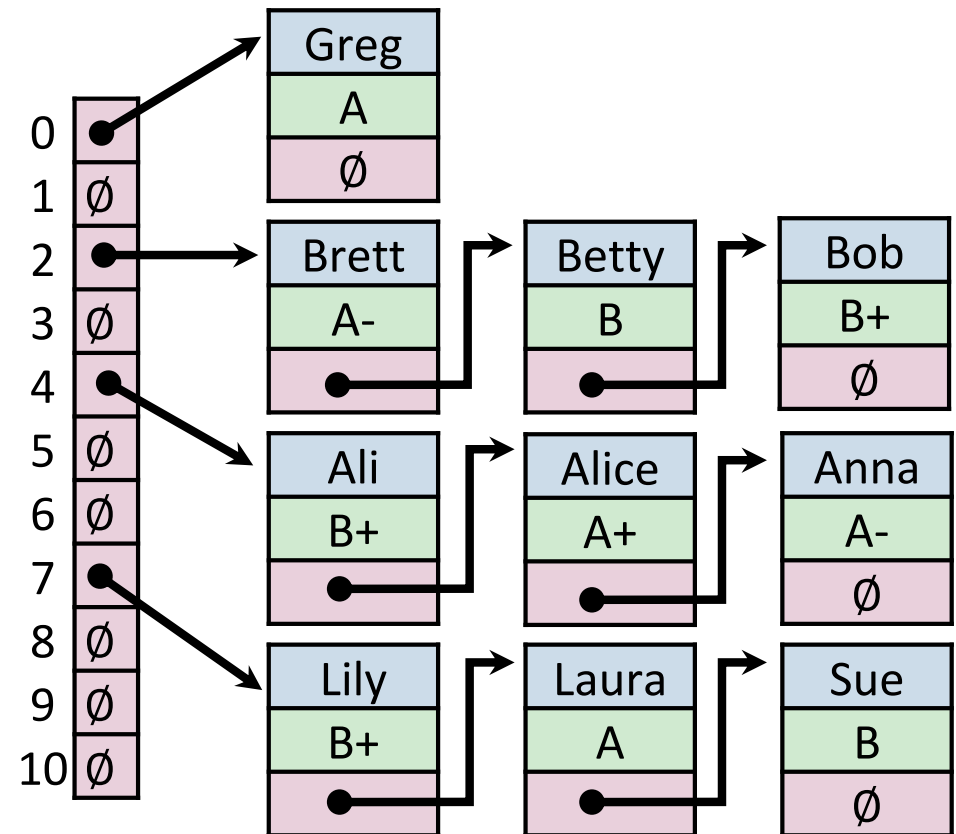
# Insertion (Separate Chaining)

Key	Value	Hash
Bob	B+	2
Anna	A-	4
Alice	A+	4
<b>Betty</b>	<b>B</b>	<b>2</b>
Brett	A-	2
Greg	A	0
Sue	B	7
Ali	B+	4
Laura	A	7
Lily	B+	7



# Insertion (Separate Chaining)

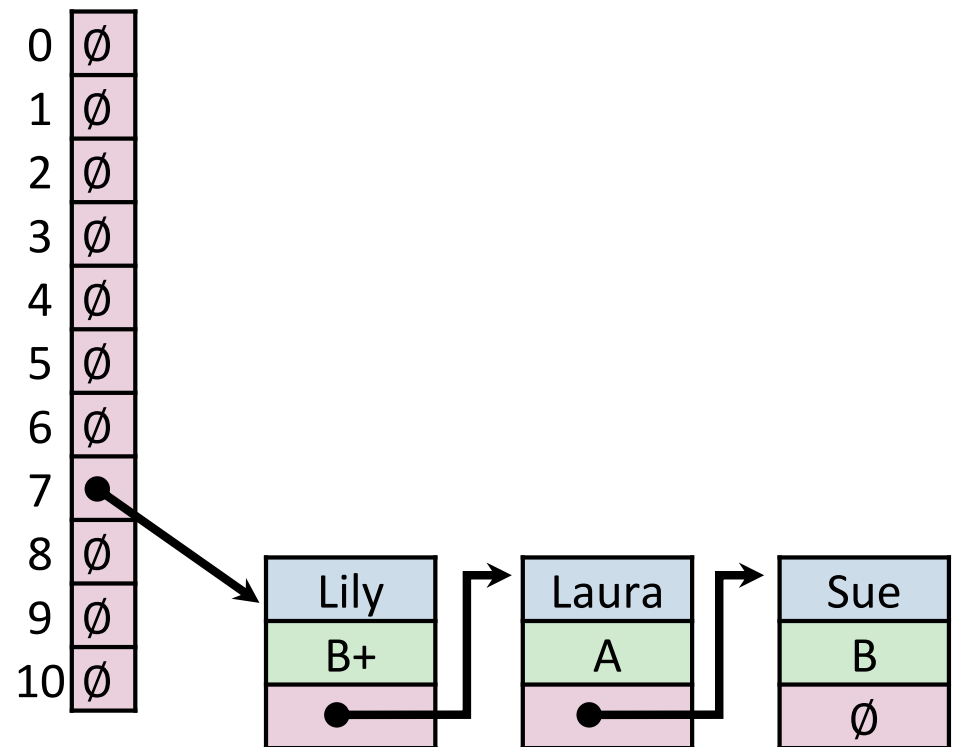
Key	Value	Hash
Bob	B+	2
Anna	A-	4
Alice	A+	4
Betty	B	2
Brett	A-	2
Greg	A	0
Sue	B	7
Ali	B+	4
Laura	A	7
Lily	B+	7



# Find (Separate Chaining)

`_find("Sue")`

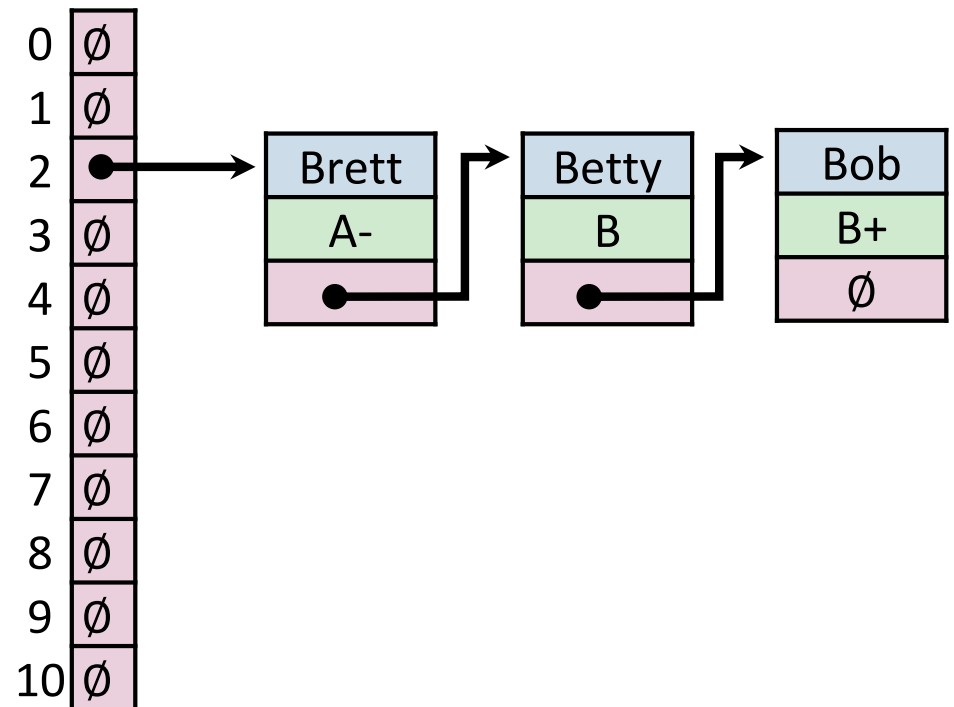
Key	Hash
Sue	7



## Remove (Separate Chaining)

`_remove("Betty")`

Key	Hash
Betty	2



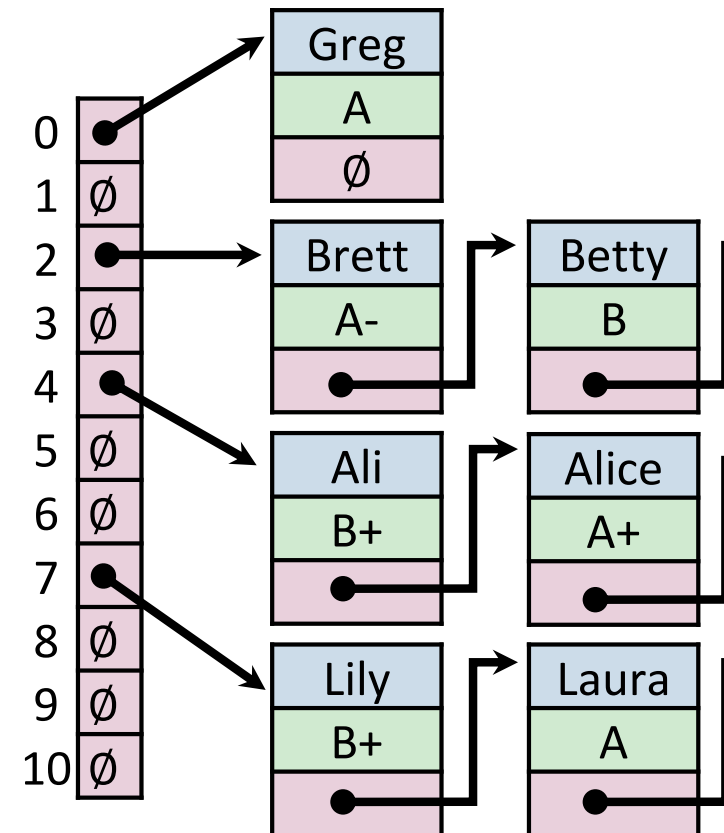
# Hash Table (Separate Chaining)

For hash table of size  $m$  and  $n$  elements:

Find runs in: \_\_\_\_\_

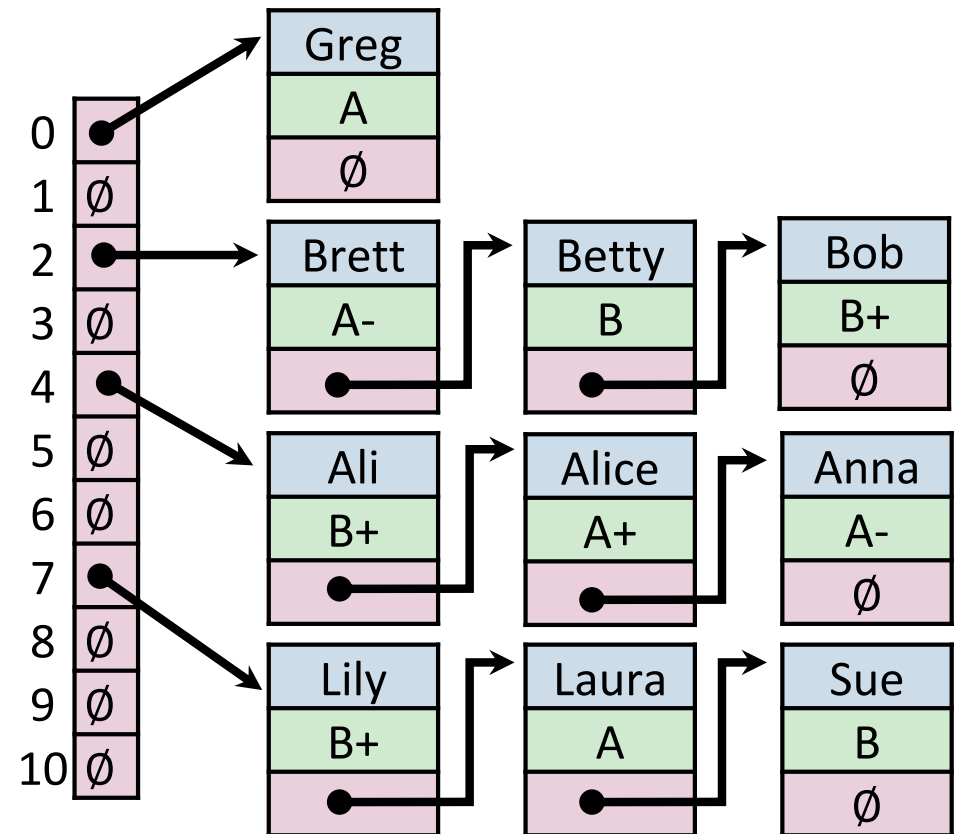
Insert runs in: \_\_\_\_\_

Remove runs in: \_\_\_\_\_



# Hash Table (Separate Chaining) – Open Hashing

Key	Value	Hash
Bob	B+	2
Anna	A-	4
Alice	A+	4
Betty	B	2
Brett	A-	2
Greg	A	0
Sue	B	7
Ali	B+	4
Laura	A	7
Lily	B+	7





# Closed Hashing

- Store the Keys and Values in the table

Index	Key	Value
0	Bob	B+
1	Greg	A
2	Alice	A+
3		
4	Brett	A-
5	Ali	B+
6		
7		
8	Laura	A
9	Anna	A-

*(Example of closed hashing)*

# Collision Handling: Probe-based Hashing

$S = \{ 1, 8, 15 \}$

$h(k) = k \% 7$

$|S| = n$

$|Array| = m$

0	
1	
2	
3	
4	
5	
6	



(Example of closed hashing)


# Collision Handling: Linear Probing

$S = \{ 16, 8, 4, 13, 29, 11, 22 \}$

$|S| = n$

$h(k) = k \% 7$

$|Array| = m$



0	
1	
2	
3	
4	
5	
6	

$h(k, i) = (k + i) \% 7$

Try  $h(k) = (k + 0) \% 7$ , if full...

Try  $h(k) = (k + 1) \% 7$ , if full...

Try  $h(k) = (k + 2) \% 7$ , if full...

Try ...

*(Example of closed hashing)*

## Collision Handling: Linear Probing

$S = \{ 16, 8, 4, 13, 29, 11, 22 \}$        $|S| = n$

$h(k, i) = (k + i) \% 7$        $|Array| = m$

find(29)

0	22
1	8
2	16
3	29
4	4
5	11
6	13

*(Example of closed hashing)*

## Collision Handling: Linear Probing

$S = \{ 16, 8, 4, 13, 29, 11, 22 \}$

$|S| = n$

$h(k, i) = (k + i) \% 7$

$|Array| = m$

\_remove(16)

0	22
1	8
2	16
3	29
4	4
5	11
6	13



# A Problem w/ Linear Probing

**Primary clustering:**

0	
1	<b>1<sub>1</sub></b>
2	<b>1<sub>2</sub></b>
3	<b>3<sub>1</sub></b>
4	<b>1<sub>3</sub></b>
5	<b>3<sub>2</sub></b>
6	
7	
8	
9	

**Description:**

**Remedy:**

(Example of closed hashing)

## Collision Handling:

$S = \{ 16, 8, 4, 13, 29, 11, 22 \}$

$|S| = n$

$h_1(k) = k \% 7$

$|Array| = m$

$h_2(k) = 1$

0	
1	8
2	16
3	
4	4
5	
6	13

$h(k, i) = (h_1(k) + i * h_2(k)) \% 7$

Try  $h(k) = (k + 0 * h_2(k)) \% 7$ , if full...

Try  $h(k) = (k + 1 * h_2(k)) \% 7$ , if full...

Try  $h(k) = (k + 2 * h_2(k)) \% 7$ , if full...

Try ...

(Example of closed hashing)

# Collision Handling: Double Hashing

$S = \{ 16, 8, 4, 13, 29, 11, 22 \}$        $|S| = n$

$$h_1(k) = k \% 7$$

$$|\text{Array}| = m$$

$$h_2(k) = 5 - (k \% 5)$$

$$h(k, i) = (h_1(k) + i * h_2(k)) \% 7$$

Try  $h(k) = (k + 0 * h_2(k)) \% 7$ , if full...

Try  $h(k) = (k + 1 * h_2(k)) \% 7$ , if full...

Try  $h(k) = (k + 2 * h_2(k)) \% 7$ , if full...

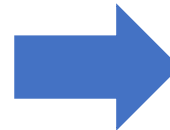
Try ...

0	
1	8
2	16
3	
4	4
5	
6	13

What to do when the table is full?

$S = \{ 16, 8, 4, 13, 29, 11, 22, 42 \}$

42



0	22
1	8
2	16
3	29
4	4
5	11
6	13



# Resizing a hash table

How do we resize?







# Running Times

	Hash Table	AVL	Linked List
Find			
Insert			
Storage Space			