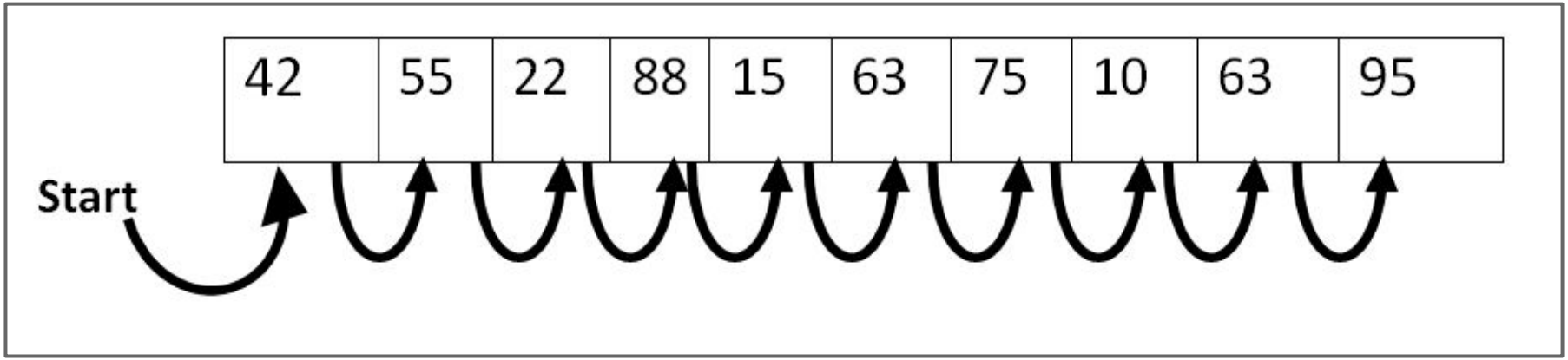
Data Structures

Hashing

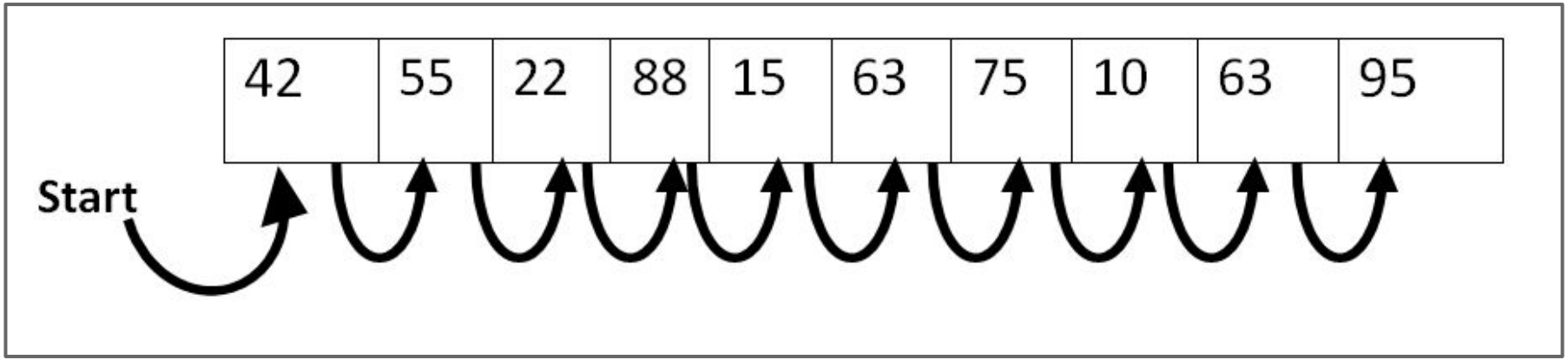
Credit: Prantik Paul, Lecturer, BRAC University

Linear Search



Problem?

Linear Insert/Delete



Problem?

Linear Search

What if it’s Binary Search?
On a sorted array?

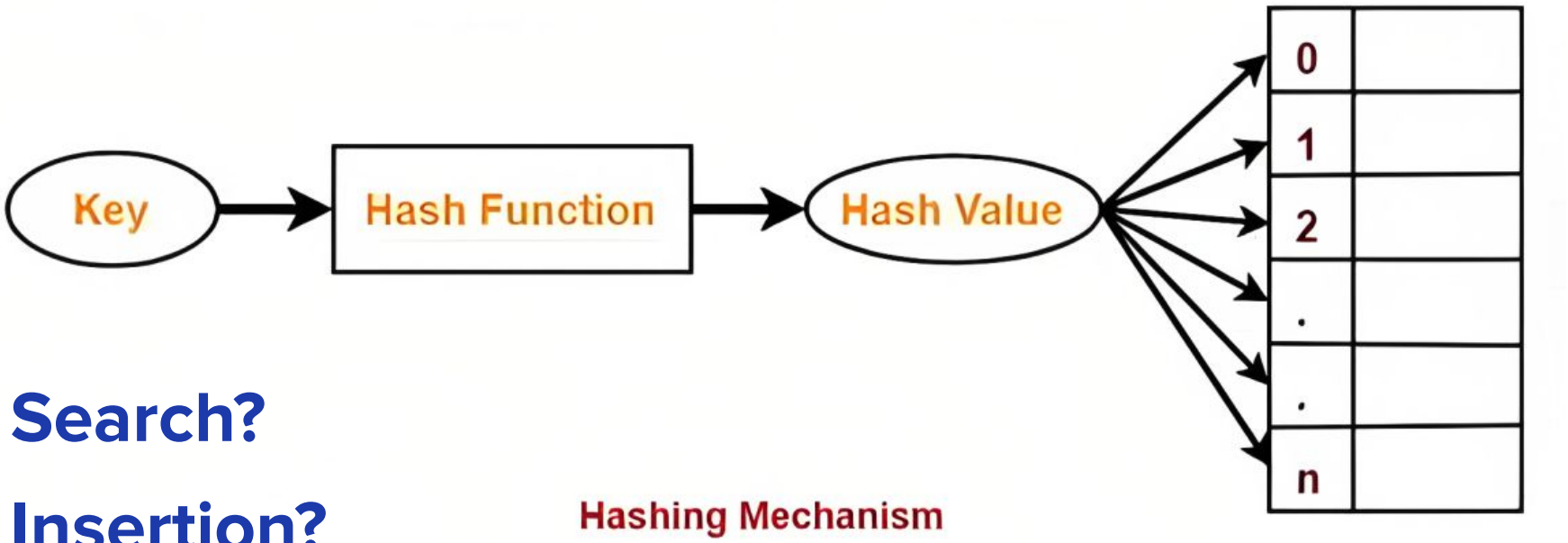
Linear Insert/Delete

What if it’s on a sorted array?

Search/Insert/Delete

What if it’s on a Linked List?

Solution? - Hashing



Hashing Mechanism

Search?

Insertion?

Deletion?

Key-Value Pairs

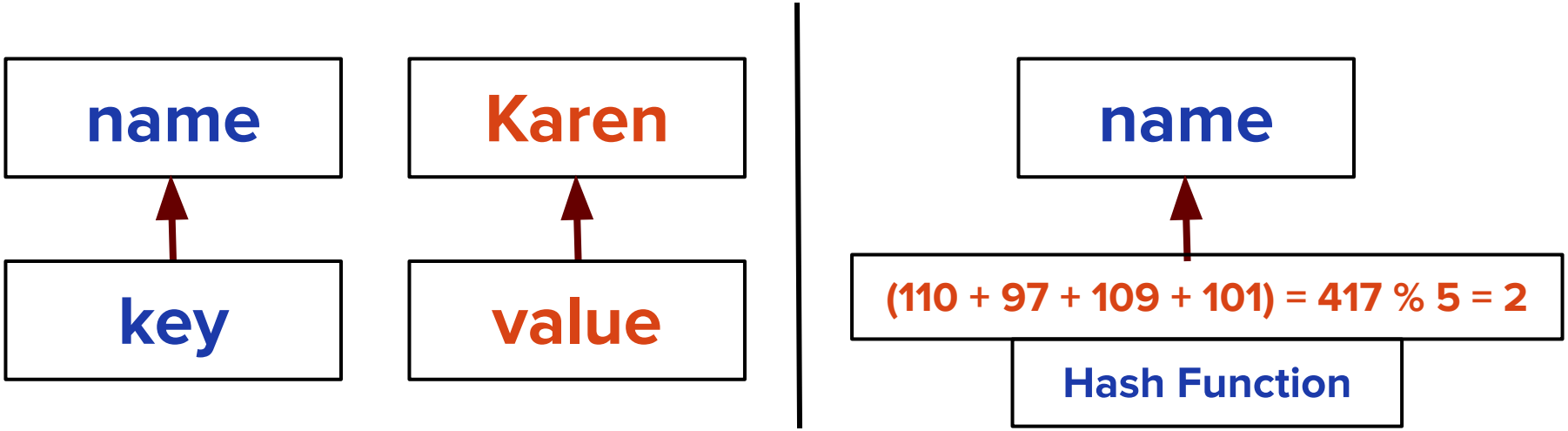
0	1	2	3	4
9341049	8828328	7100090	9889849	9651423

0	1	2	3	4
Mumit	Belal	Mukul	Muzil	Muhit

Key-Value Pairs

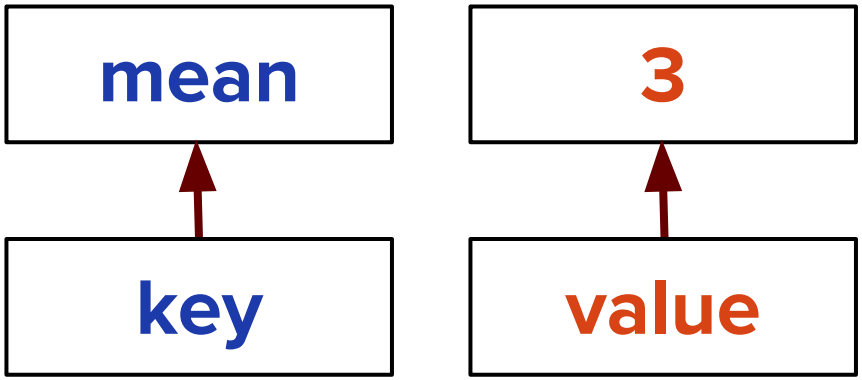
0	1	 700000	....800000...	9889849
null	null	Mr. X	Ms. Y	Muzil

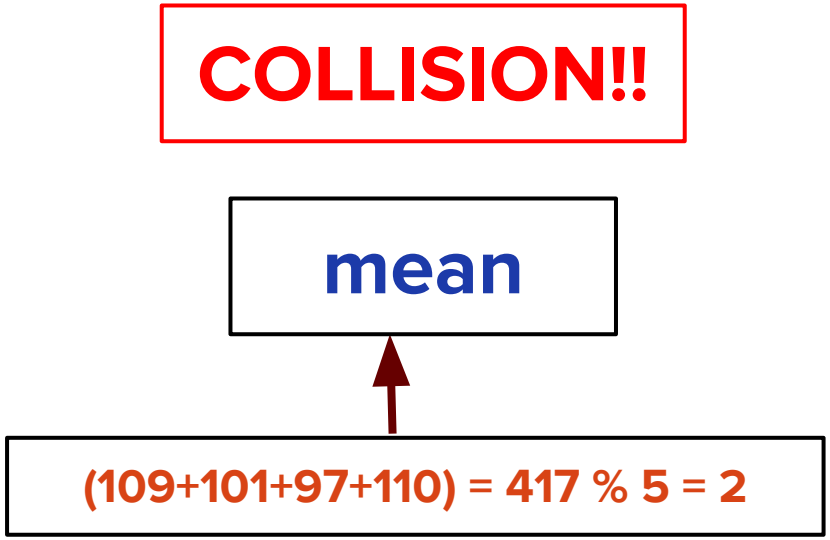
Hashing



		name		
		Karen		

Hashing





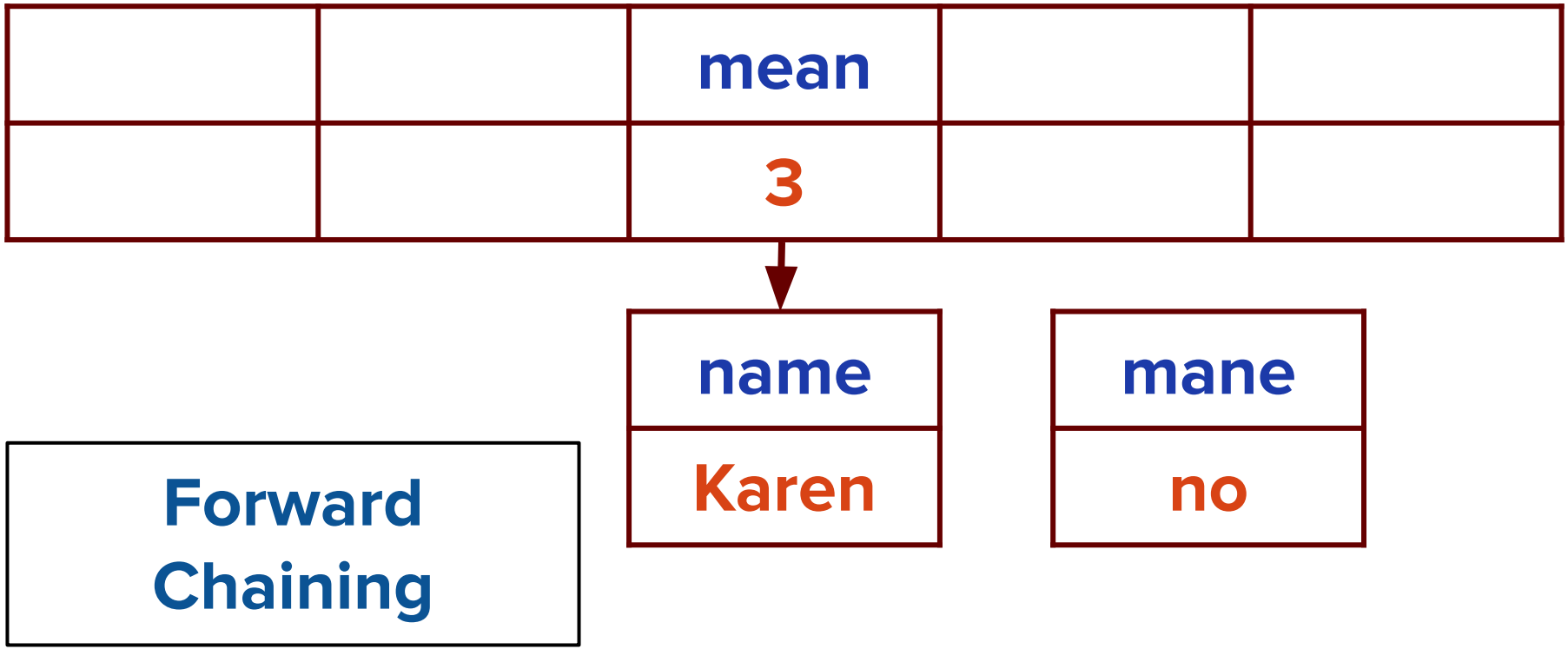
		name		
		Karen		

Hashing

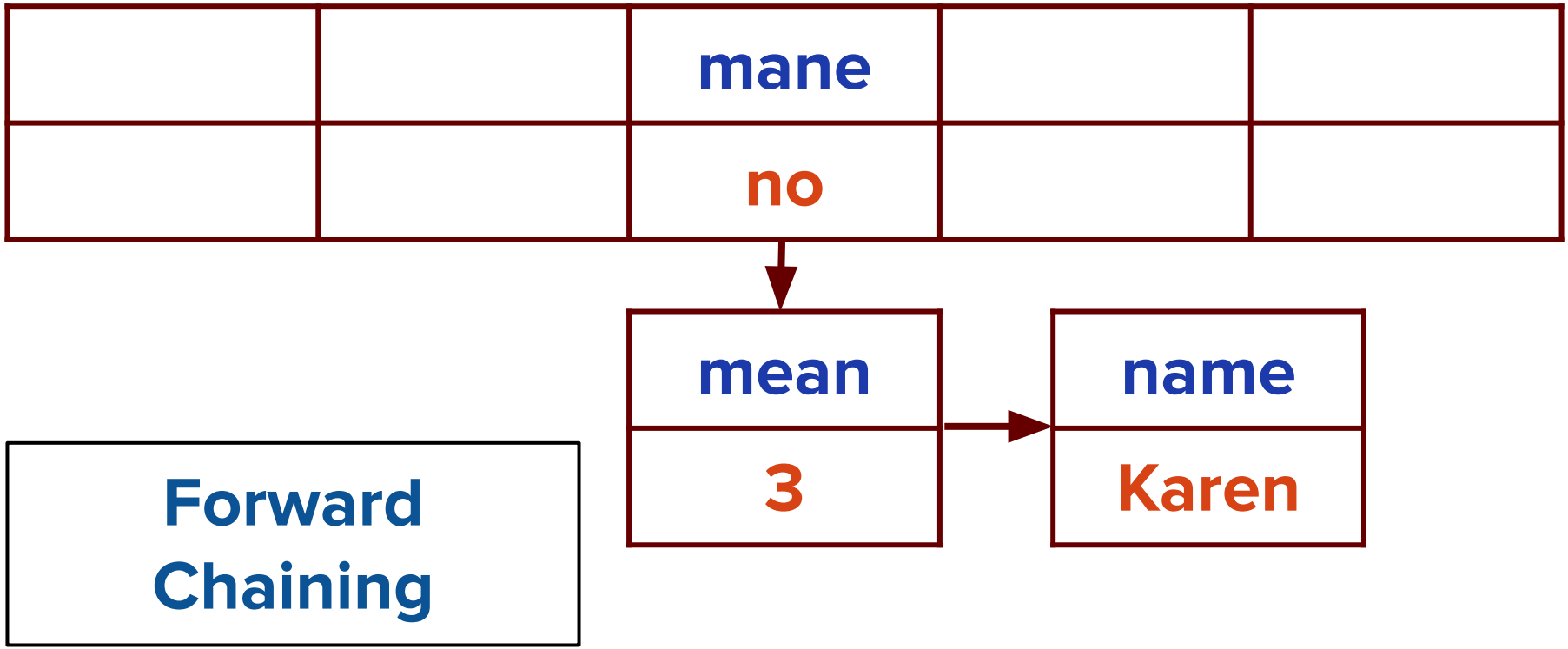
		name		
		Karen		

mean
3

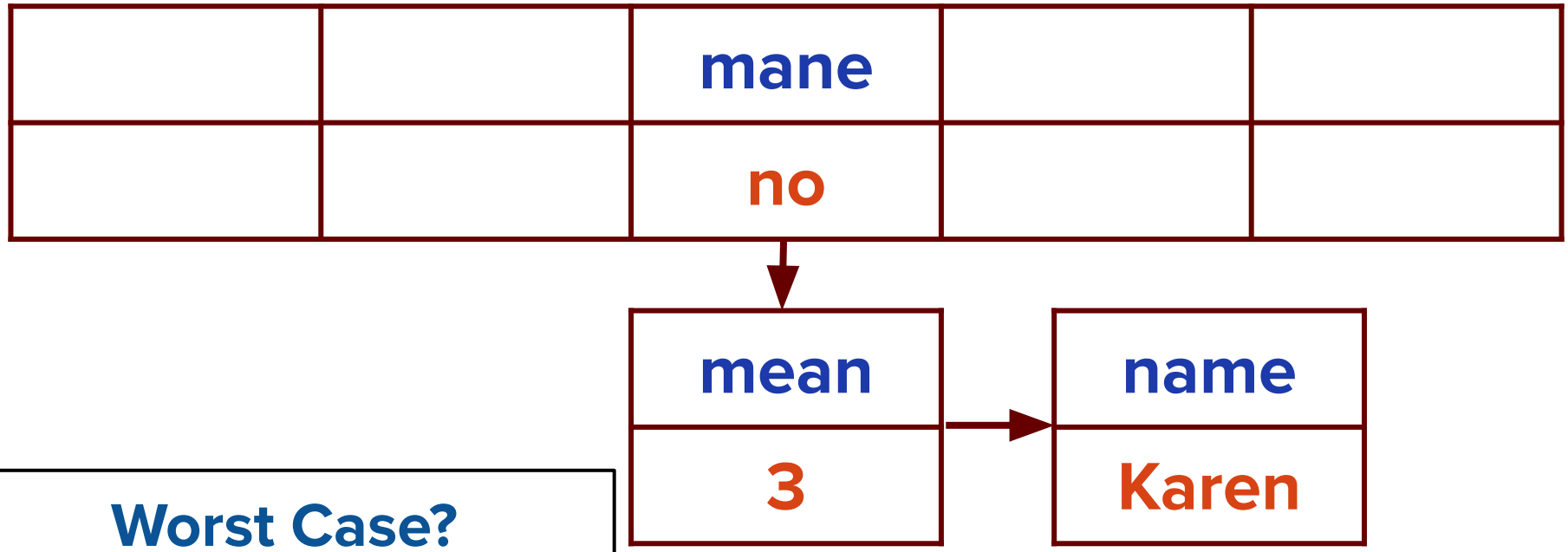
Hashing



Hashing



Hashing



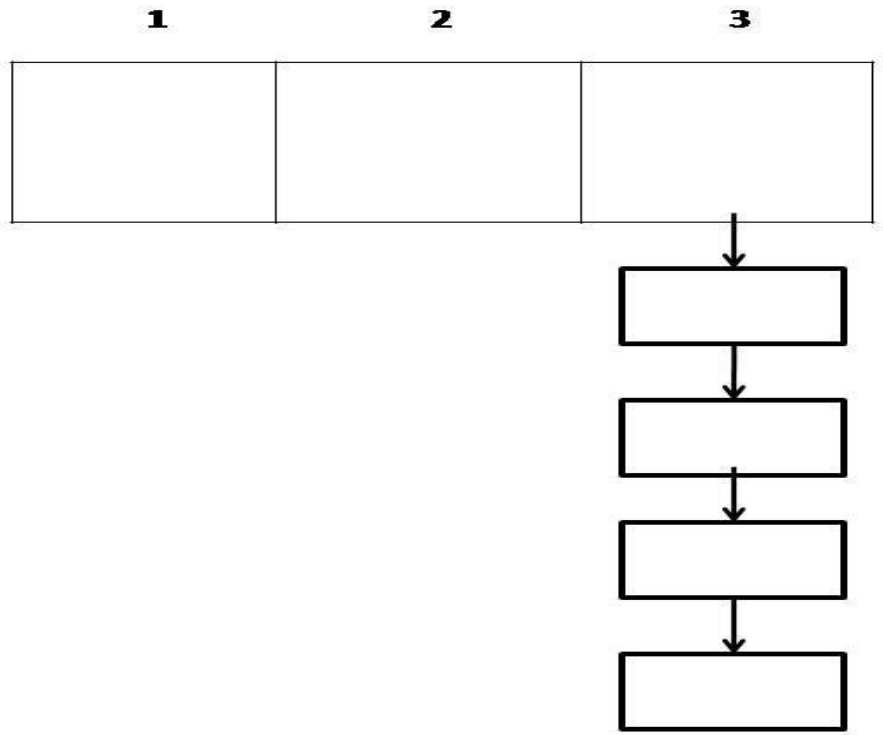
Worst Case?
Search/Insert/Delete?

Forward Chaining

How big should the array be?

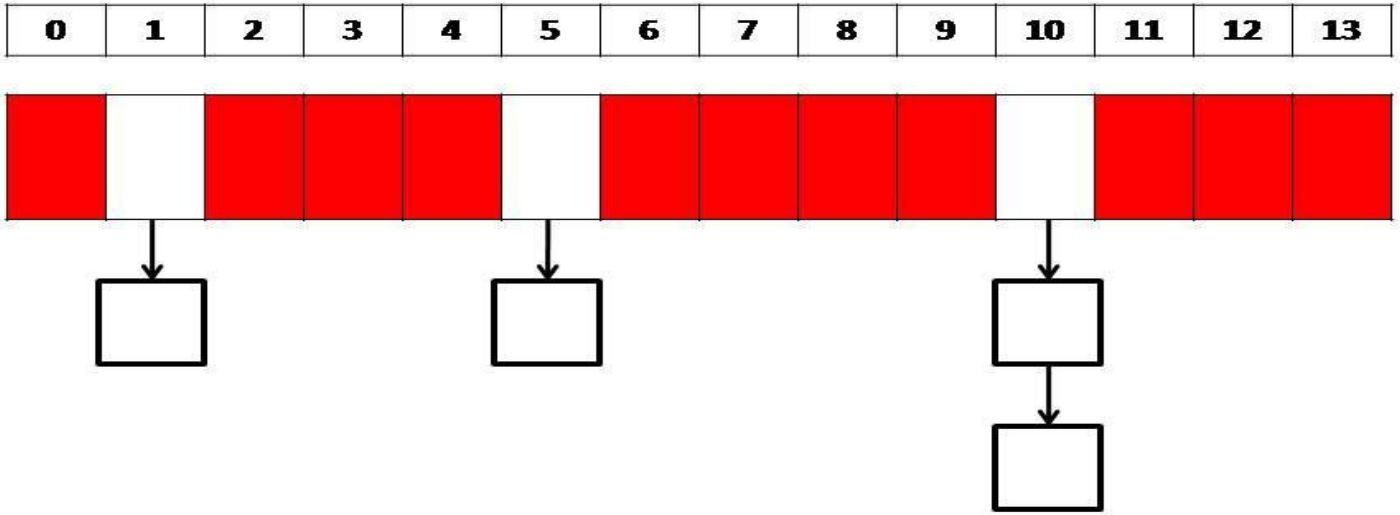
Forward Chaining

Too small



Forward Chaining

Too big



Forward Chaining (Insert)

```
def Forward_Chaining_Insert(arr):
    hashtable = [None] * size

    for element in arr:
        hash_value = hash_func(elem)
        if( hashtable[hash_value] == None):
            hashtable[hash_value] = Node(elem, None)
        else:
            current = hash_table[hash_value]
            hash_table[hash_value] = Node(elem, current)
```

Forward Chaining (Search)

```
def Forward_Chaining_Search(elem):
    hash_value = hash_func(elem)
    temp = hash_table[hash_value]

    while (temp!=None):
        if(temp.elem == elem):
            return True
        temp = temp.next

    return False
```

Forward Chaining (Delete)

Do It Yourself

Forward Chaining

hash(key) = key % 5

```
Insert (Key: 12 Value: "Apple")
Insert (Key: 5  Value: "Orange")
Insert (Key: 17 Value: "Banana")
Insert (Key: 10 Value: "Grapes")
Insert (Key: 22 Value: "Watermelon")
Insert (Key: 15 Value: "Pineapple")
```