

# ECAP770

ADVANCE DATA STRUCTURES

---

Ashwani Kumar  
Assistant Professor



# Learning Outcomes



After this lecture, you will be able to

- understand hash functions

# Hashing

- Hashing data structure is used to store and find data efficiently using array.
- It is a technique or process of mapping keys, values into the hash table by using a hash function. It is done for faster access to elements.
- The efficiency of mapping depends on the efficiency of the hash function used.

# Steps to implement hashing

- An element is converted into an integer by using a hash function. This element can be used as an index to store the original element, which falls into the hash table.
- The element is stored in the hash table where it can be quickly retrieved using hashed key.

# Steps to implement hashing

`hash = hash func(key)`

`index = hash % array_size`

- The hash is independent of the array size and it is then reduced to an index (a number between 0 and `array_size - 1`) by using the modulo operator (%).

# Hash function

- A hash function is any function that can be used to map a data set of an arbitrary size to a data set of a fixed size, which falls into the hash table.
- The values returned by a hash function are called hash values, hash codes, hash sums, or hashes

# Hash function

- An efficient hash function should be built such that the index value of the added item is distributed equally across the table.
- An effective collision resolution technique should be created to generate an alternate index for a key whose hash index corresponds to a previously inserted position in a hash table.

# Characteristics of hash function

- Uniform Distribution: For distribution throughout the constructed table.
- Fast: The generation of hash should be very fast, and should not produce any considerable overhead.
- Less collisions: Collisions occur when pairs of elements are mapped to the same hash value. These should be avoided.

# Hash function

## Methods for calculating the hash function

- Division method
- Mid square method
- Digit folding method / pairing method
- Multiplication Method

# Division method

Hash function  
 $h(x) = x \% 10$

5  
10  
12  
8  
14  
19

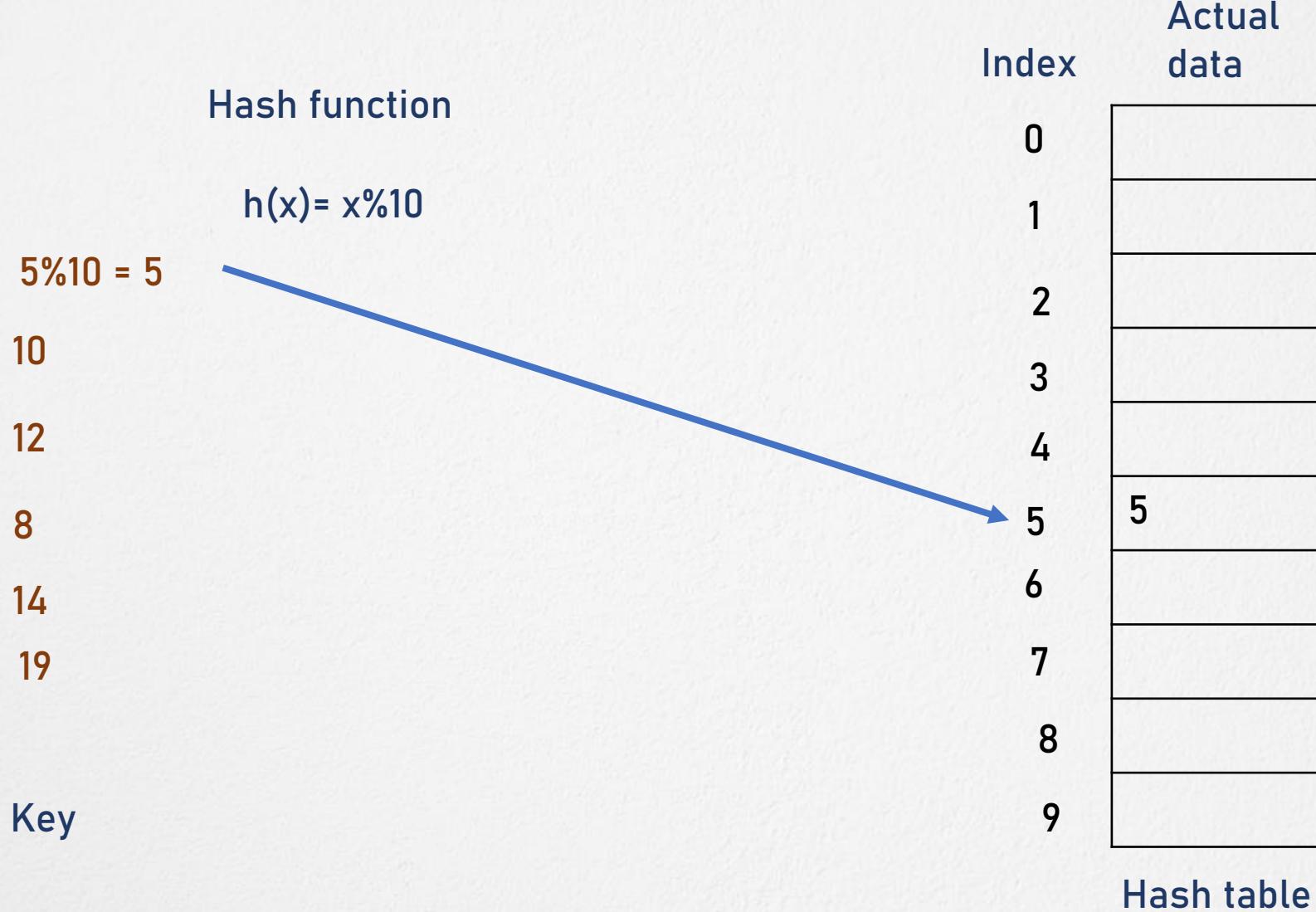
Key

Size of hash table

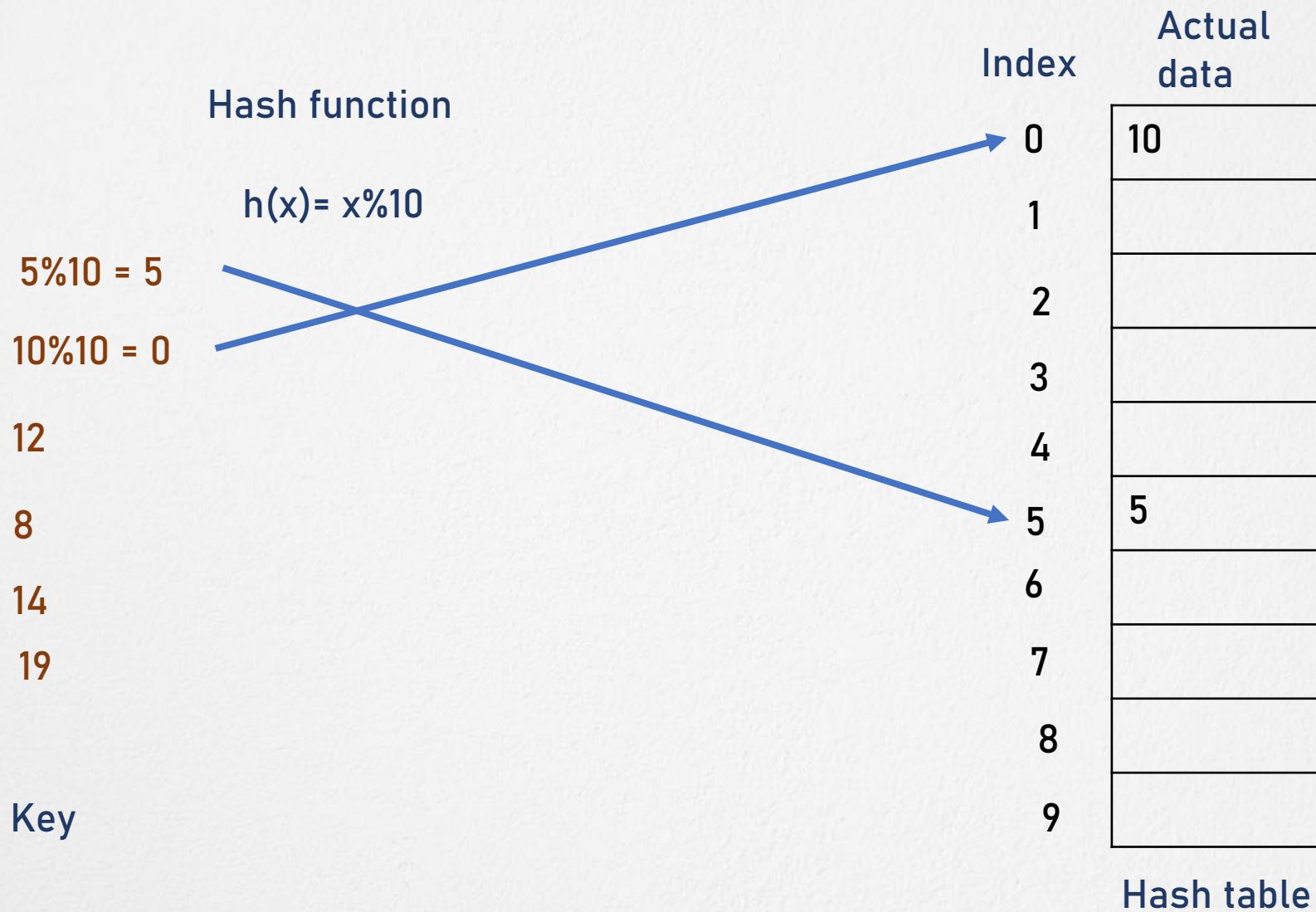
Index	Actual data
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Hash table

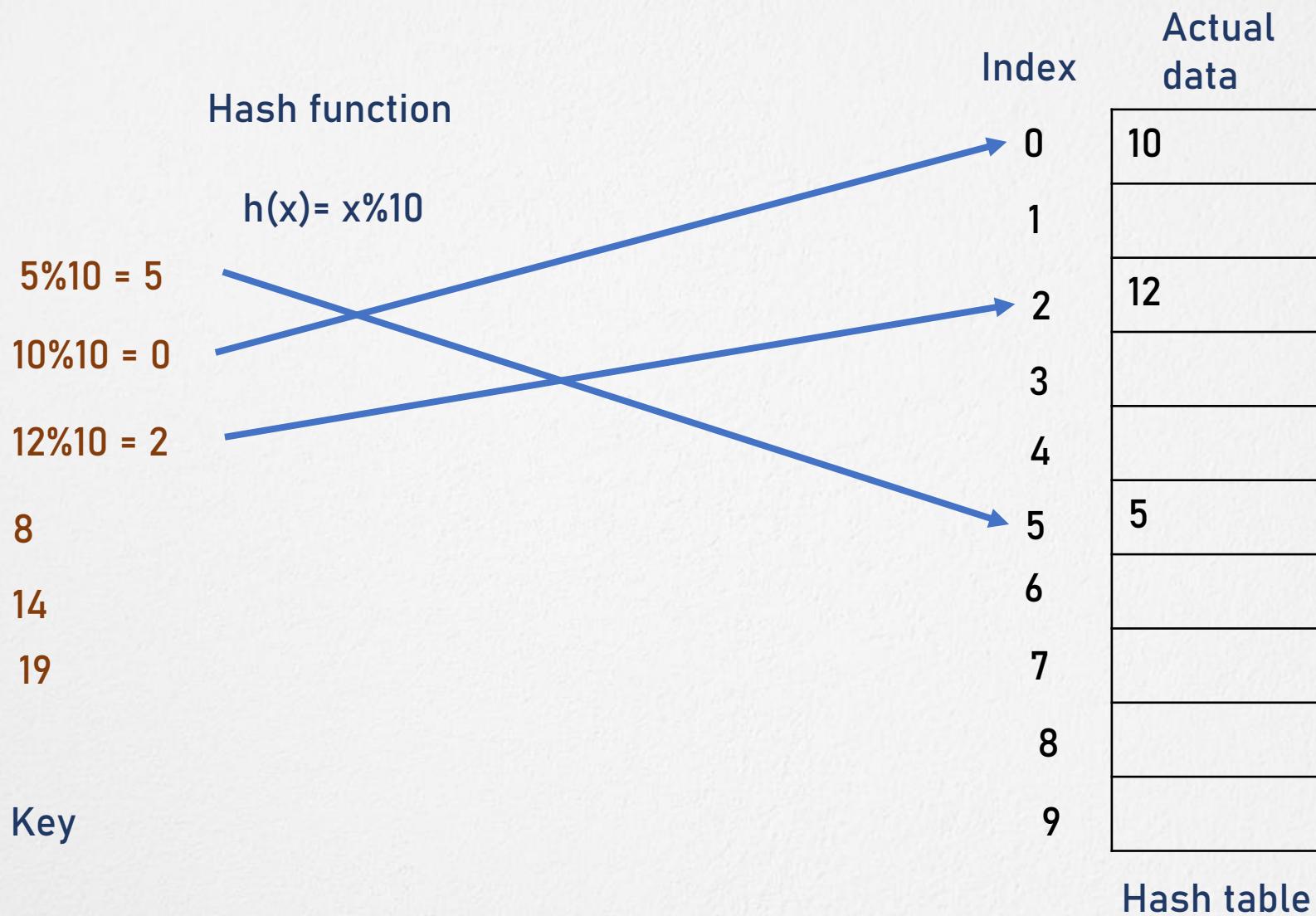
# Division method



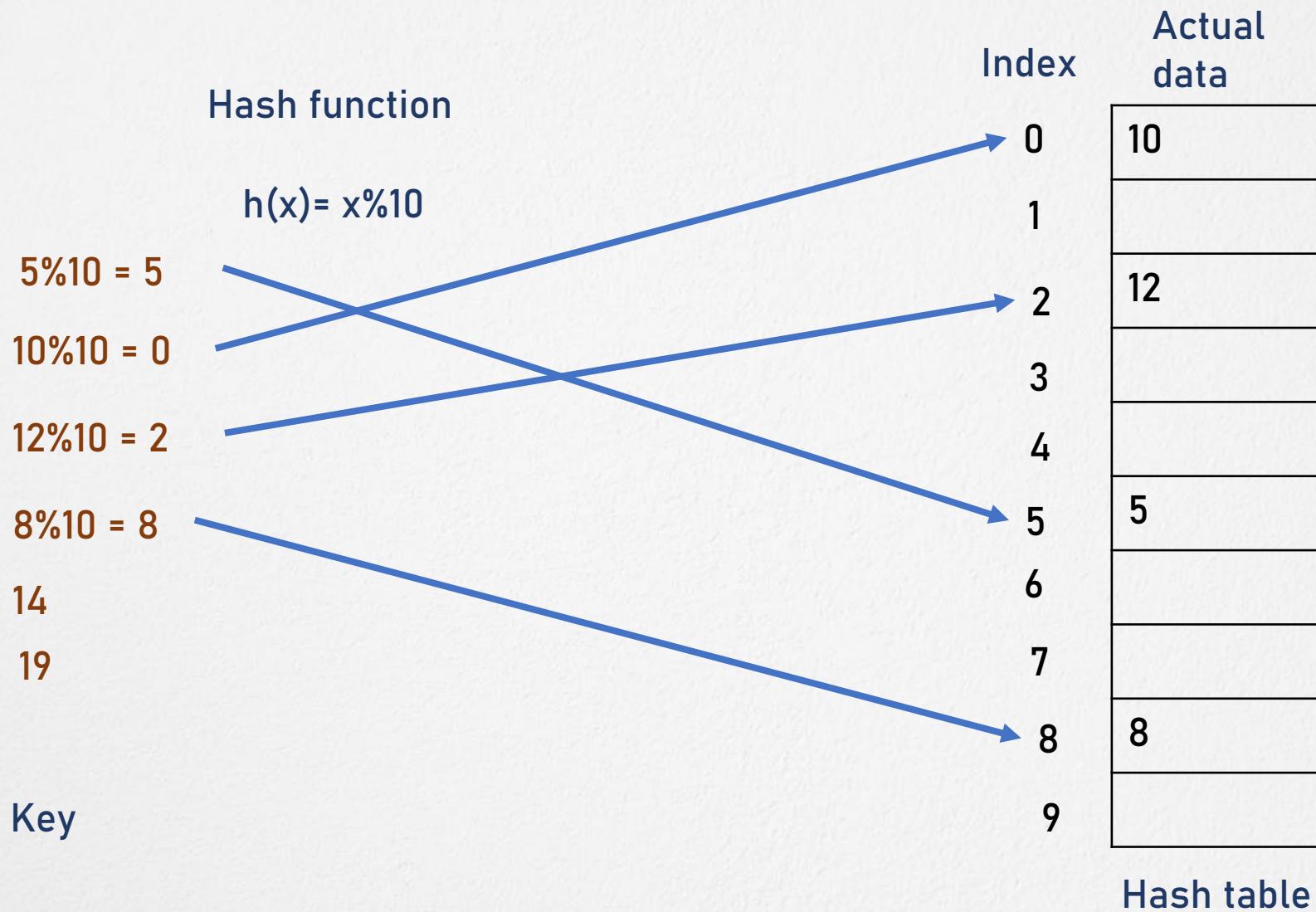
# Division method



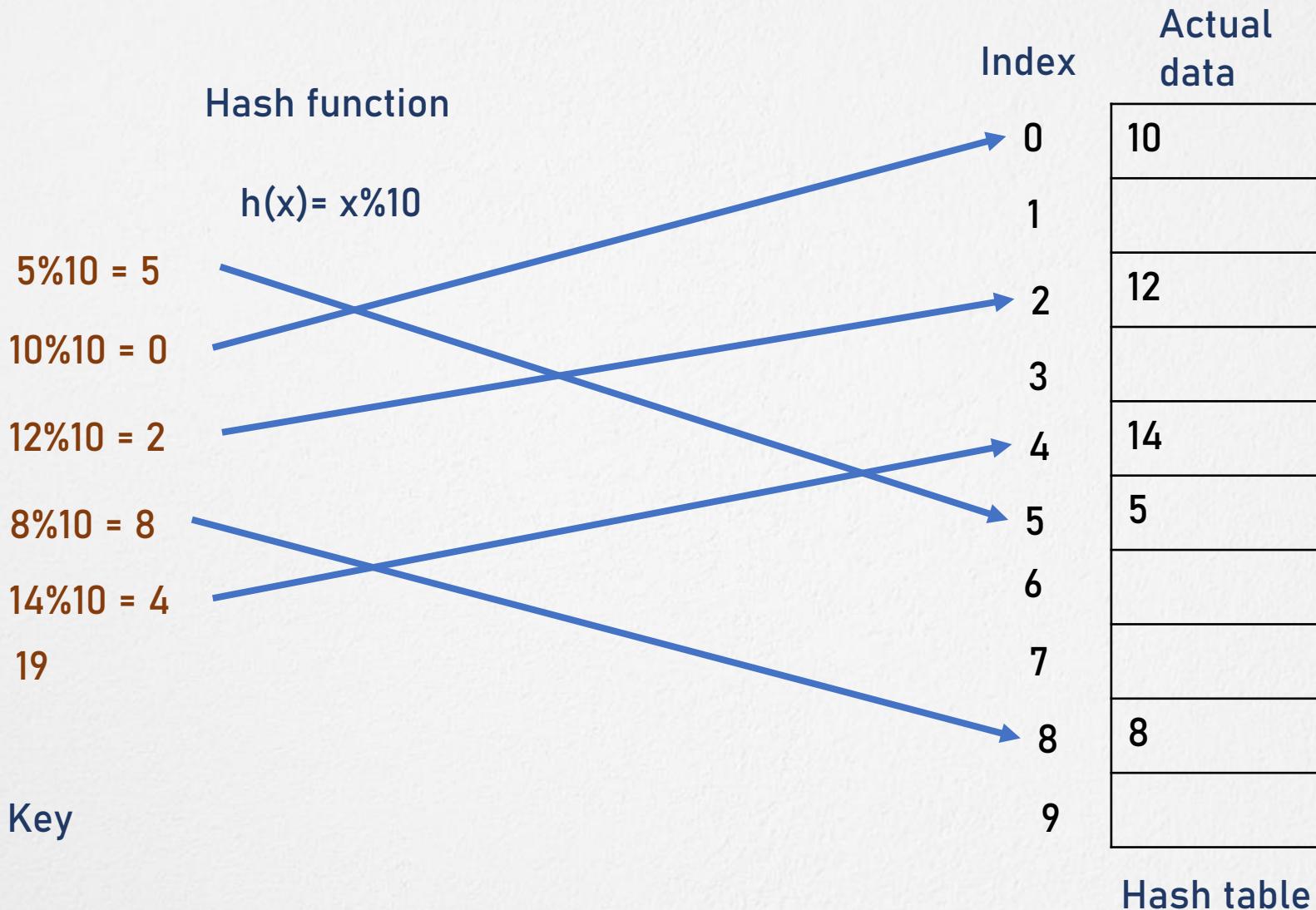
# Division method



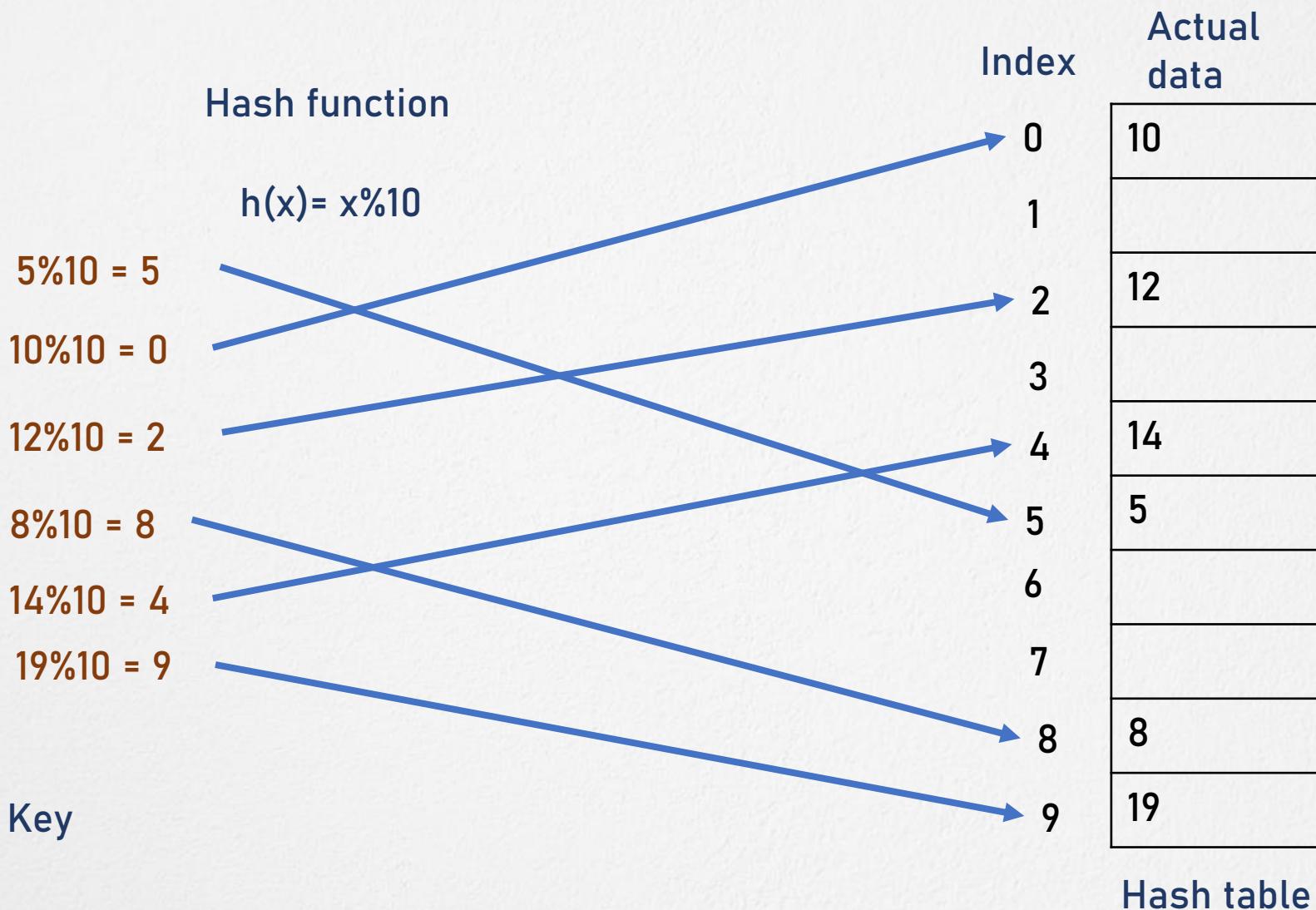
# Division method



# Division method



# Division method



# Mid square method

- In this technique, squares the key value. Then, some digits from the middle are extracted.
- These extracted digits form a number which is taken as the new number for address.

## Limitation

- The size of  $\text{key}^2$  is too large.

E.g.  $2025 * 2025 = 4100625$

# Mid square method

$$H(key) = key^2$$

$$10 \quad 10 * 10 = 100$$

15

12

22

14



## Hash table

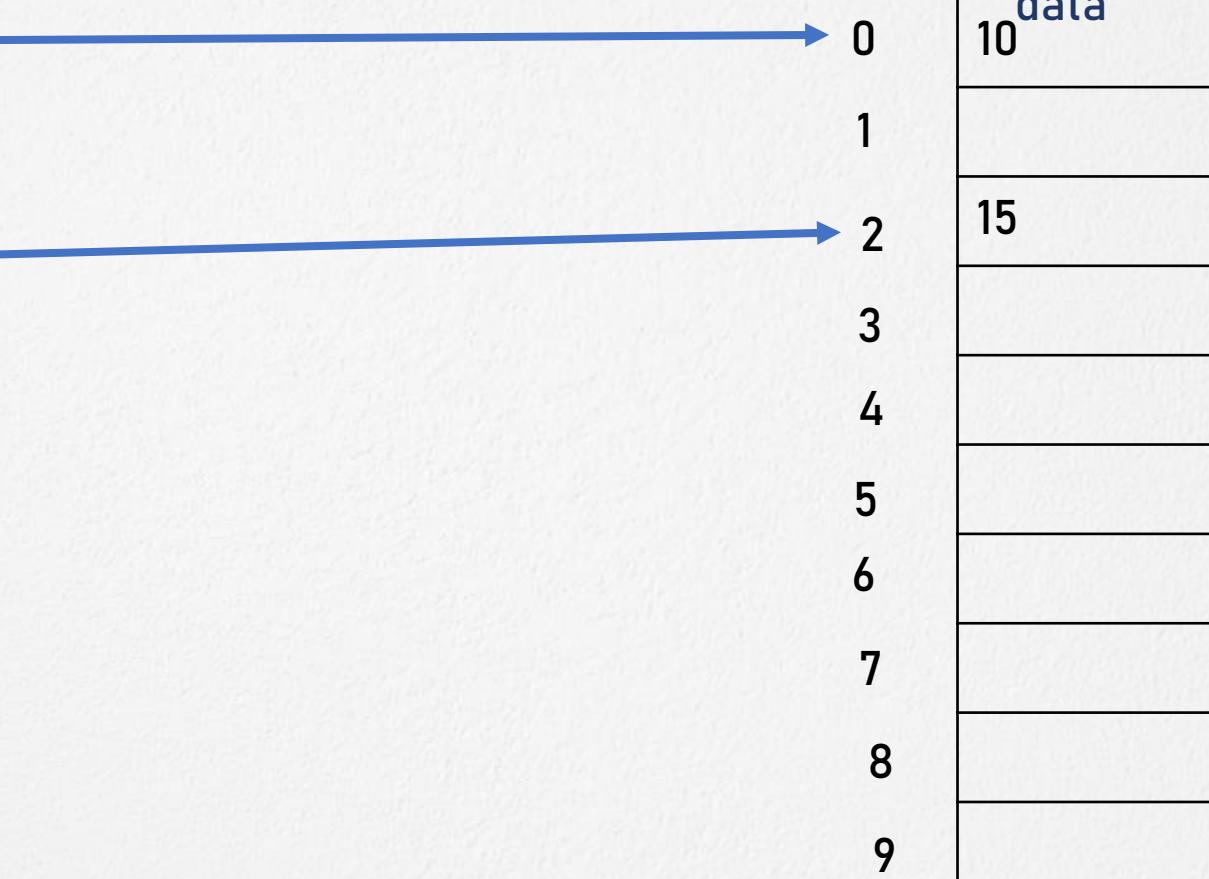
# Mid square method

$$H(key) = \text{key}^2$$

10  
15  
12  
22  
14

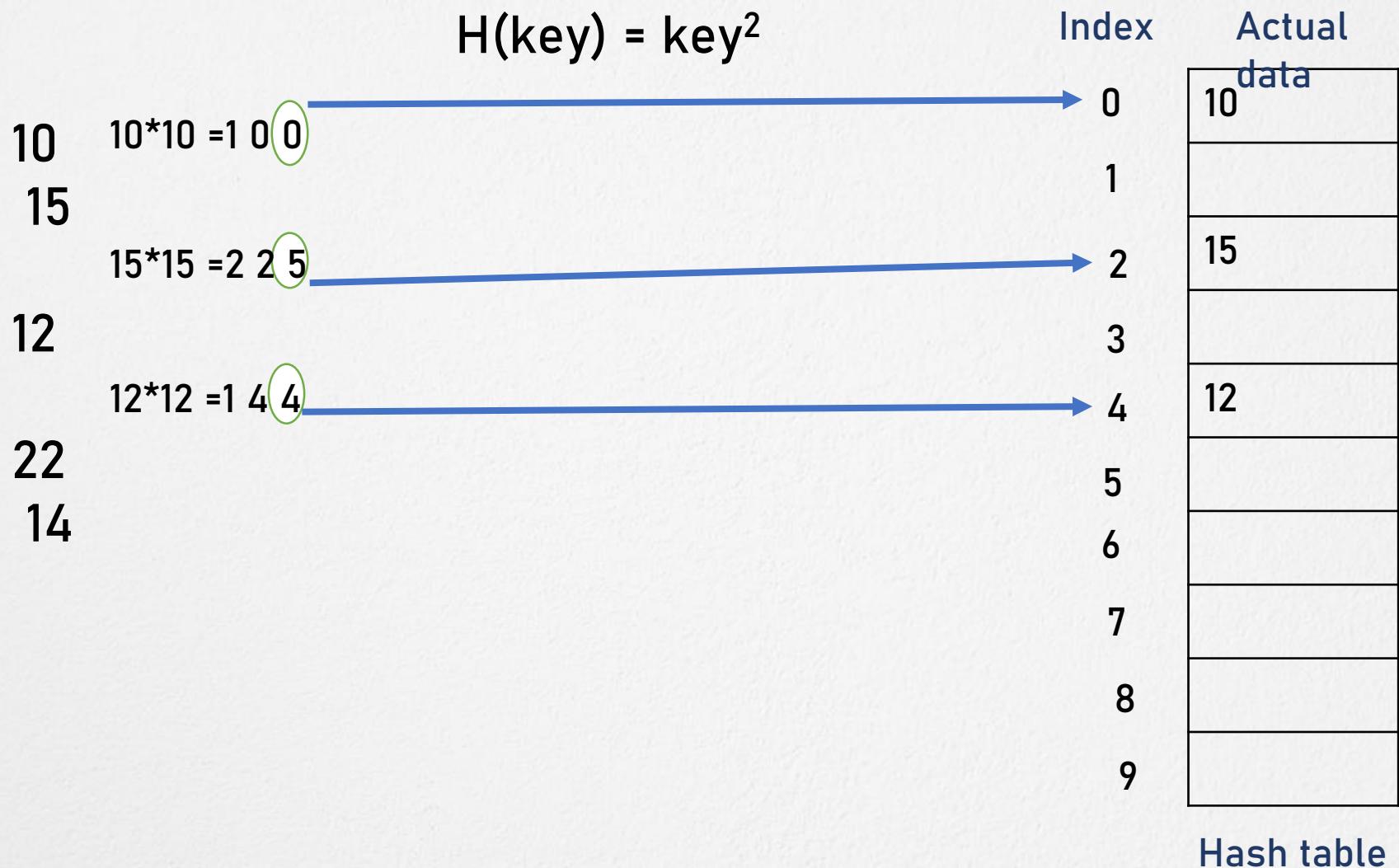
$$10*10 = 100$$

$$15*15 = 225$$

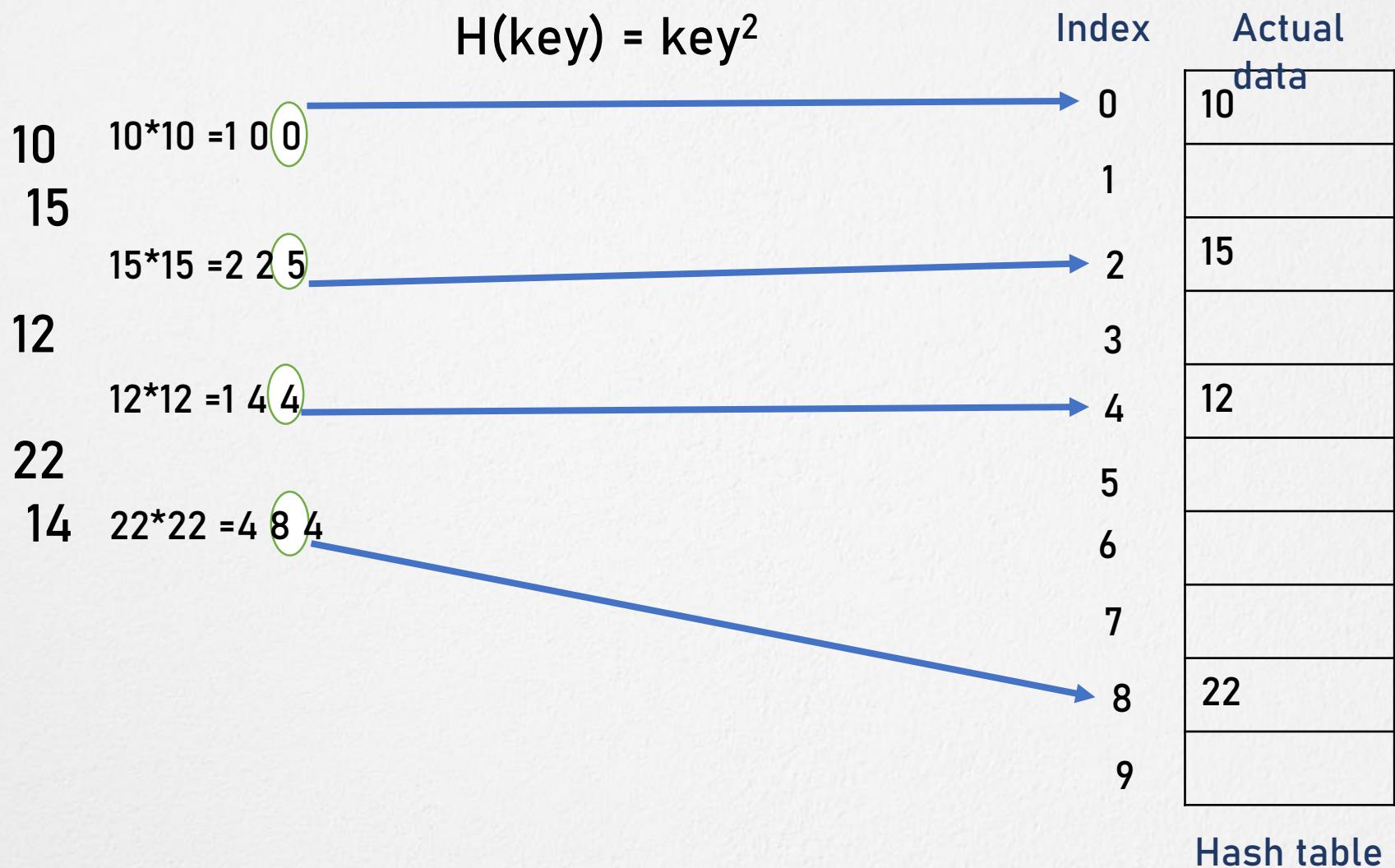


Hash table

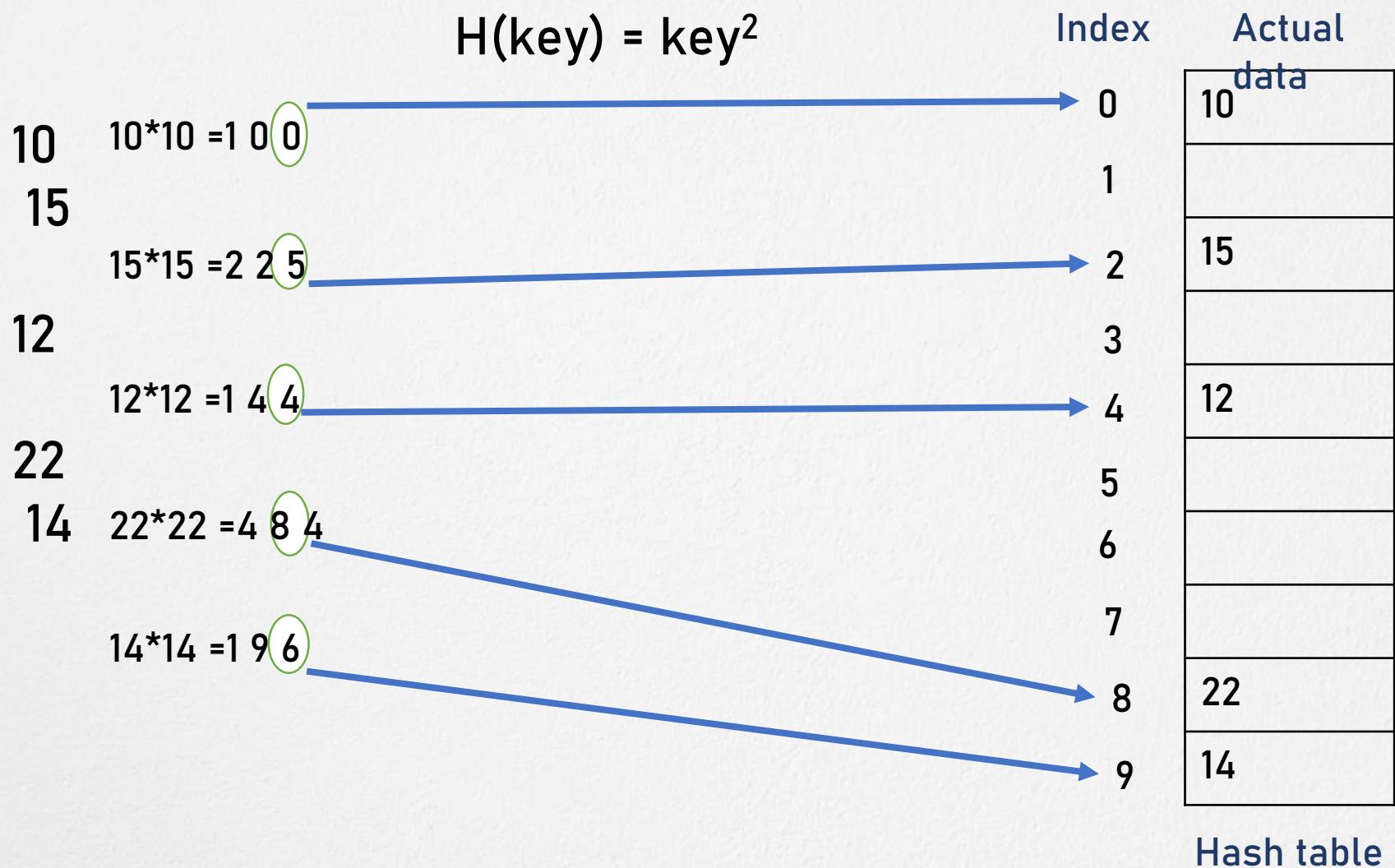
# Mid square method



# Mid square method



# Mid square method



That's all for now...