

# Data Structure

Trainer : Nisha Dingare

Email : nisha.dingare@sunbeaminfo.com



# HashTable :

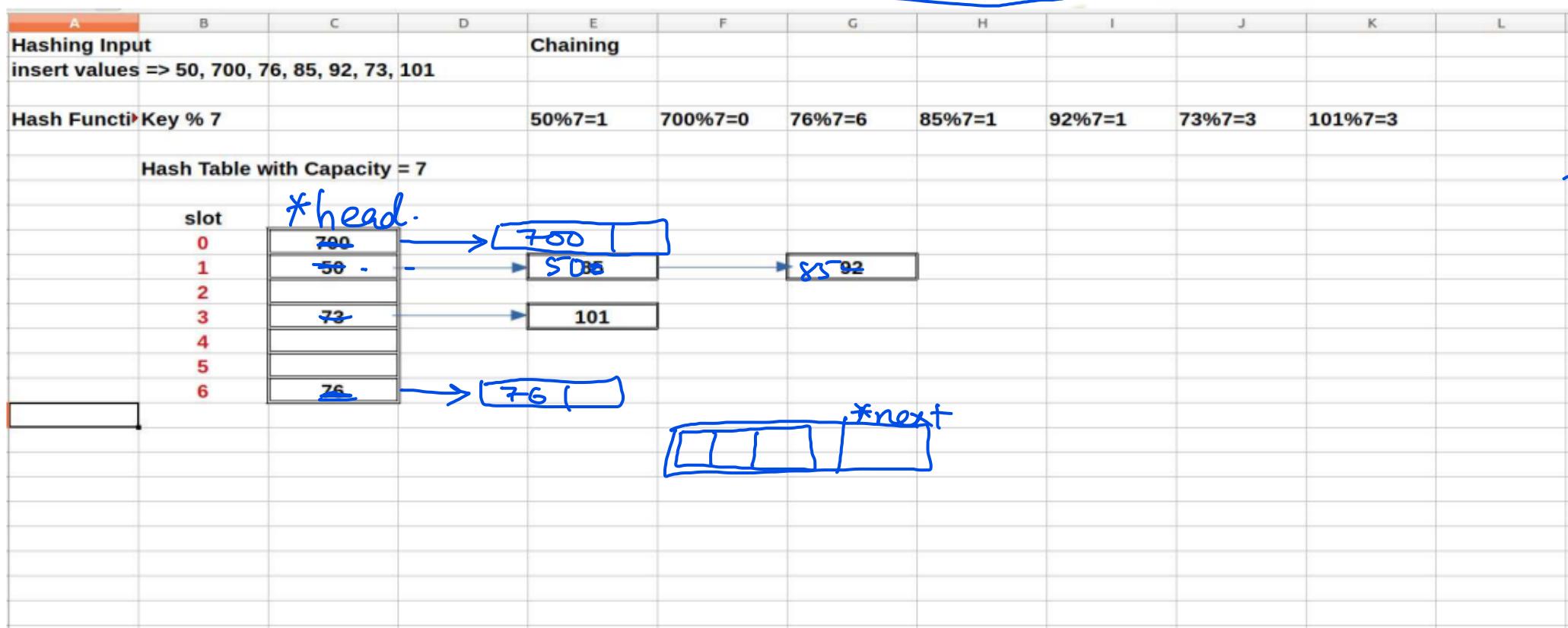
- **HashTable:** Hash table is an **associative** data structure in which data is stored in **key-value pairs** so that for the given key, value can be searched in minimal possible time. Ideal time complexity is O(1). Internally it is an array (table) where values can accessed by the index (slot) calculated from the key.
- **Hash Function:** It is mathematical function of the key that yields slot of the hash table where key-value is stored. Simplest example is:  $f(k) = k \% \text{ size}$ .
- **Collision:** There is possibility that two keys result in same slot. This is called collision and must be handled using some **collision handling technique**. It is handled by Open addressing or Chaining.

Hashing Input								
insert values => 50, 700, 76, 85, 92, 73, 101								
Hash Function	Key % 7	50%7=1	700%7=0	76%7=6	85%7=1	92%7=1	73%7=3	101%7=3
Hash Table with Capacity = 7								
slot 0 1 2 3 4 5 6	700 50 85 92 73 101 76				collision			

# Collision Handling Technique : chaining :

## 1. Chaining:

- The idea is to make each cell of hash table point to a linked list of records that have same hash function value. Chaining is simple, but requires additional memory outside the table.



# Collision Handling Technique : Open Addressing :

## 2. Open Addressing:

- In open addressing, all elements are stored in the hash table itself. Each table entry contains either a record or NIL. When searching for an element, we one by one examine table slots until the desired element is found or it is clear that the element is not in the table.
- Open Addressing is done following ways:

**A. Linear Probing:** In linear probing, if collision occurs next free slot will be searched/probed linearly.

$$(k+i) \% s$$

$$k \% s \rightarrow \text{collision}$$

**B. Quadratic Probing:** In quadratic probing, if collision occurs next free slot will be searched/probed quadratically.

**C. Double Hashing:** In double hashing, if collision occurs next free slot will be searched/probed by using another hash function, so two hash functions can be used to find next/probe next free slot.



# HashTable :

- **Load Factor =  $n / m$** 
  - $n$  = Number of key-value pairs to be inserted in the hash table  $\frac{7}{10}$ .
  - $m$  = Number of slots in the hash table  $10$
  - If  $n < m$ , then load factor  $< 1$  ✓
  - If  $n = m$ , then load factor = 1  $\frac{10}{10} =$
  - If  $n > m$ , then load factor  $> 1$   $\frac{15}{10}$
- **Limitations of Open Addressing**
  - Open addressing requires more computation.
  - Cannot be used if load factor is greater than 1 (i.e. number of pairs are more than number of slots in the table).

# Thank You

