

# DATA STRUCTURES (ITPC-203)

## Hashing



**Dr. Sanga Chaki**

**Department of Information Technology**

**Dr. B. R. Ambedkar National Institute of Technology, Jalandhar**



# Contents

1. Hashing
2. Hash Function
3. Collision Resolution Strategies
4. Applications.

# Hashing

1. Hashing is a technique or process of mapping keys, and values into the hash table by using a hash function.
2. Done for faster access to elements
3. Example: Consider the dictionary as an ADT – termed as the symbol table
4. we define the symbol table as a set of name (key) - attribute pairs.
5. The characteristics of the name and attribute vary according to the application.
6. For example, in a thesaurus, the name is a word, and the attribute is a list of synonyms for the word

# Hashing

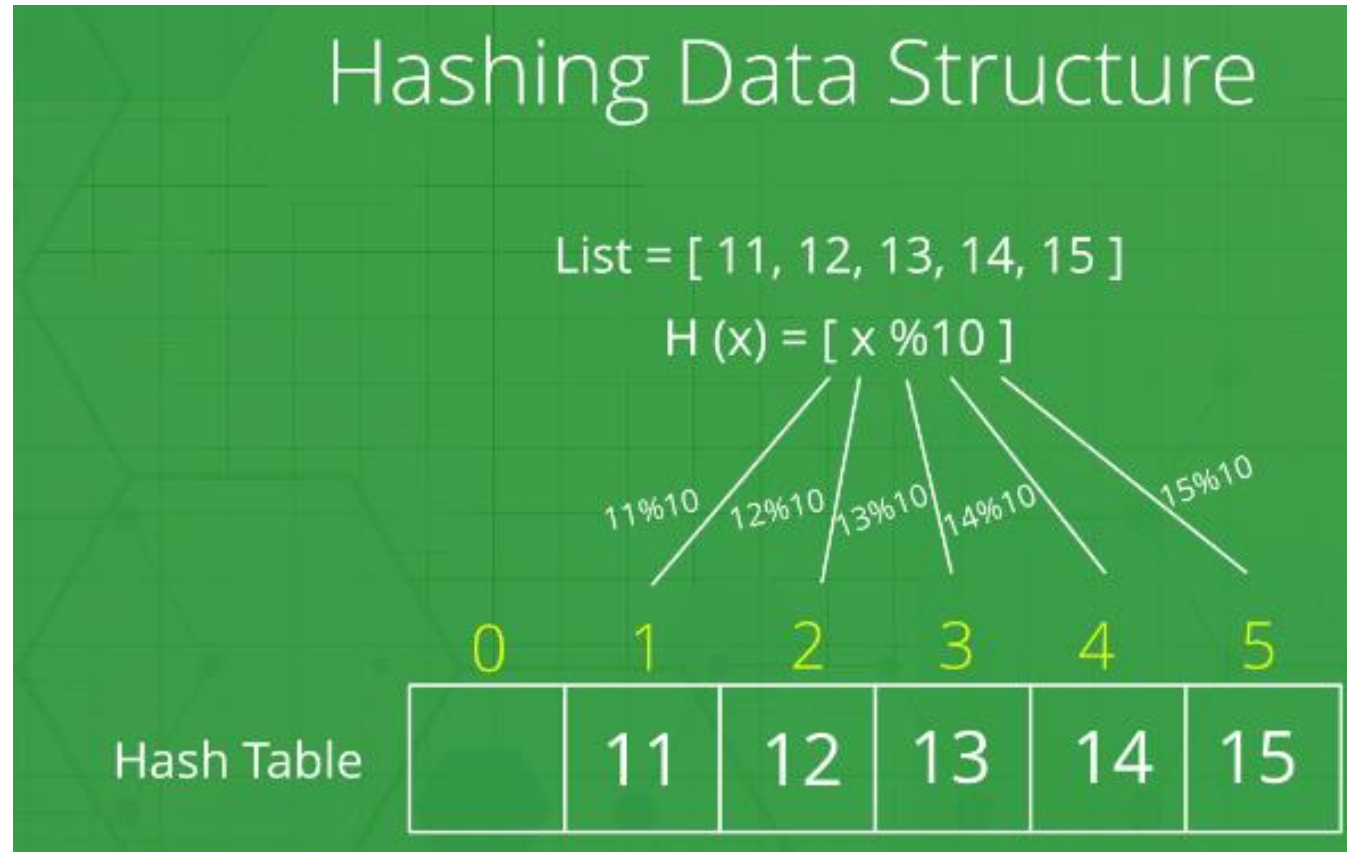
1. Generally we would want to perform the following operations on any symbol table:
  - a) determine if a particular name is in the table
  - b) retrieve the attributes of that name
  - c) modify the attributes of that name
  - d) insert a new name and its attributes

# Hash Table

1. We store the identifiers in a fixed size table called a **hash table**
2. We use an arithmetic function – or the **hash function**,  $f$ , to determine the address, or location, of an identifier,  $x$ , in the table.
3. Thus,  $f(x)$  gives the hash, or home address, of  $x$  in the table.
4. The hash table  $ht$  of size  $n$  is stored in sequential memory locations,  $ht[0], \dots$  ,  $ht[n - 1]$ .

# Hash Function

1. Let a hash function  $H(x)$  maps the value  $x$  at the index  $x\%10$  in an Array.
2. If the list of values is  $[11, 12, 13, 14, 15]$  it will be stored at positions  $\{1, 2, 3, 4, 5\}$  in the array or Hash table respectively.

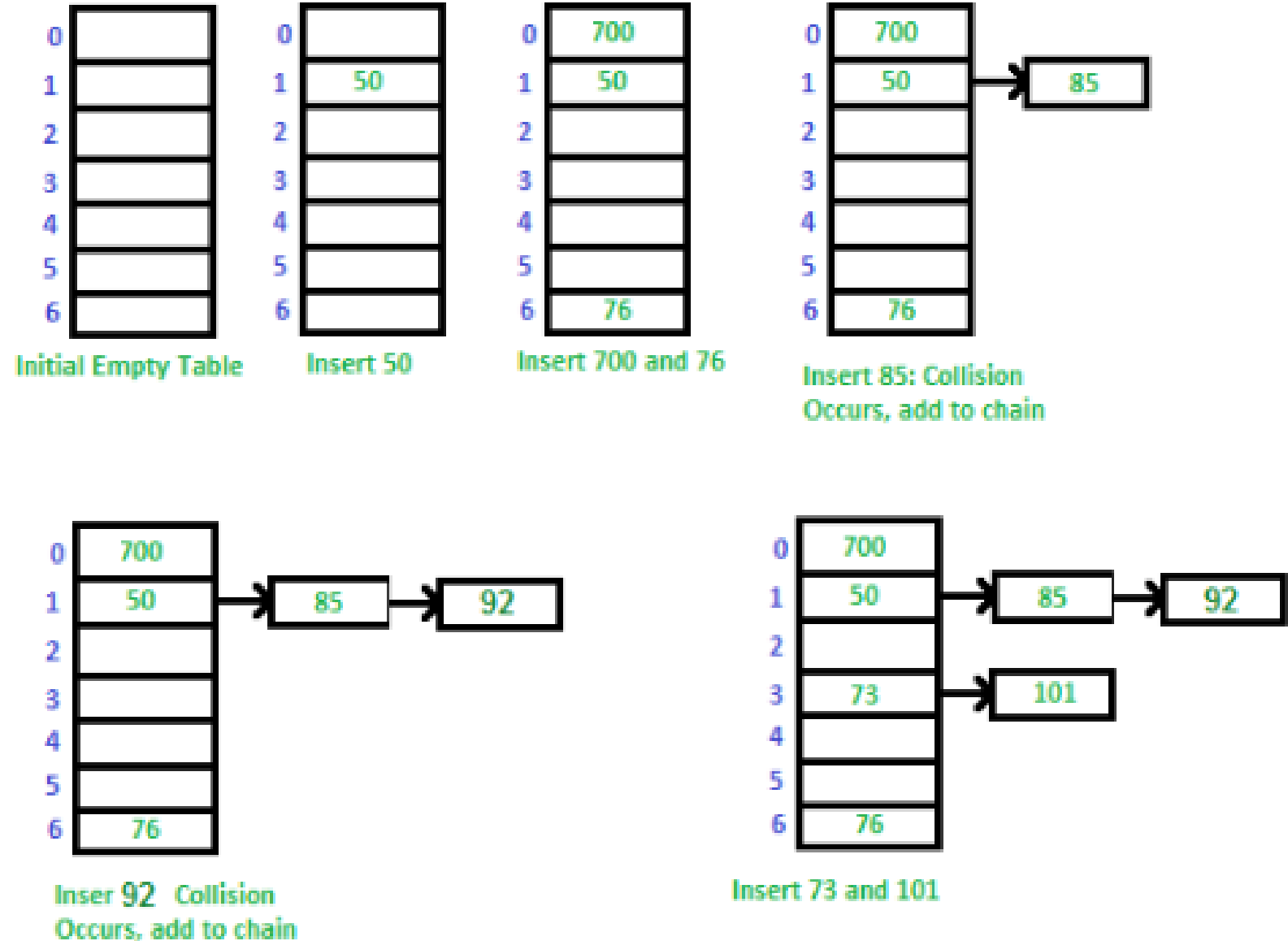


# Collision and Collision Handling

1. Since a hash function maps a larger value (key) into a comparatively small number (address), there is a possibility that two keys result in the same value.
2. Example: In the previous example, if the array was {11, 12, 13, 14, 15, 21}, then both 11 and 21 would be hashed to the array index 1.
3. This situation where a newly inserted key maps to an already occupied slot in the hash table is called collision and must be handled using some collision handling technique.
4. There are mainly two methods to handle collision:
  - a) Separate Chaining
  - b) Open Addressing/Closed hashing - all elements are stored in the hash table itself. So at any point, the size of the table must be greater than or equal to the total number of keys. Procedure is based upon probing.

# Separate Chaining

1. When multiple elements are hashed into the same slot index, then these elements are inserted into a singly-linked list which is known as a chain.
2. Example: Consider a simple hash function as “key mod 7” and a sequence of keys as 50, 700, 76, 85, 92, 73, 101





# Applications

1. Memory Hashing: Hashed Page Tables are a type of data structure used by operating systems to efficiently manage memory mappings
  - between virtual (hard disk) and physical (RAM) memory addresses.
  - allows for faster lookups whenever an active process in OS requires some data to be loaded from/to virtual memory (key) to/from the physical memory for processing.
2. Compact Garbage Collection: The space that is created by garbage collection creates may be scattered among many small blocks of memory.
  - This fragmentation may prevent the space from being used effectively.
  - A compacting collector is one that tries to move the blocks of allocated memory together, compacting them so that there is no unused space between them.
  - It needs to track pointers to allocated objects
  - They use an object (hash) table containing pointers to all allocated objects (key).