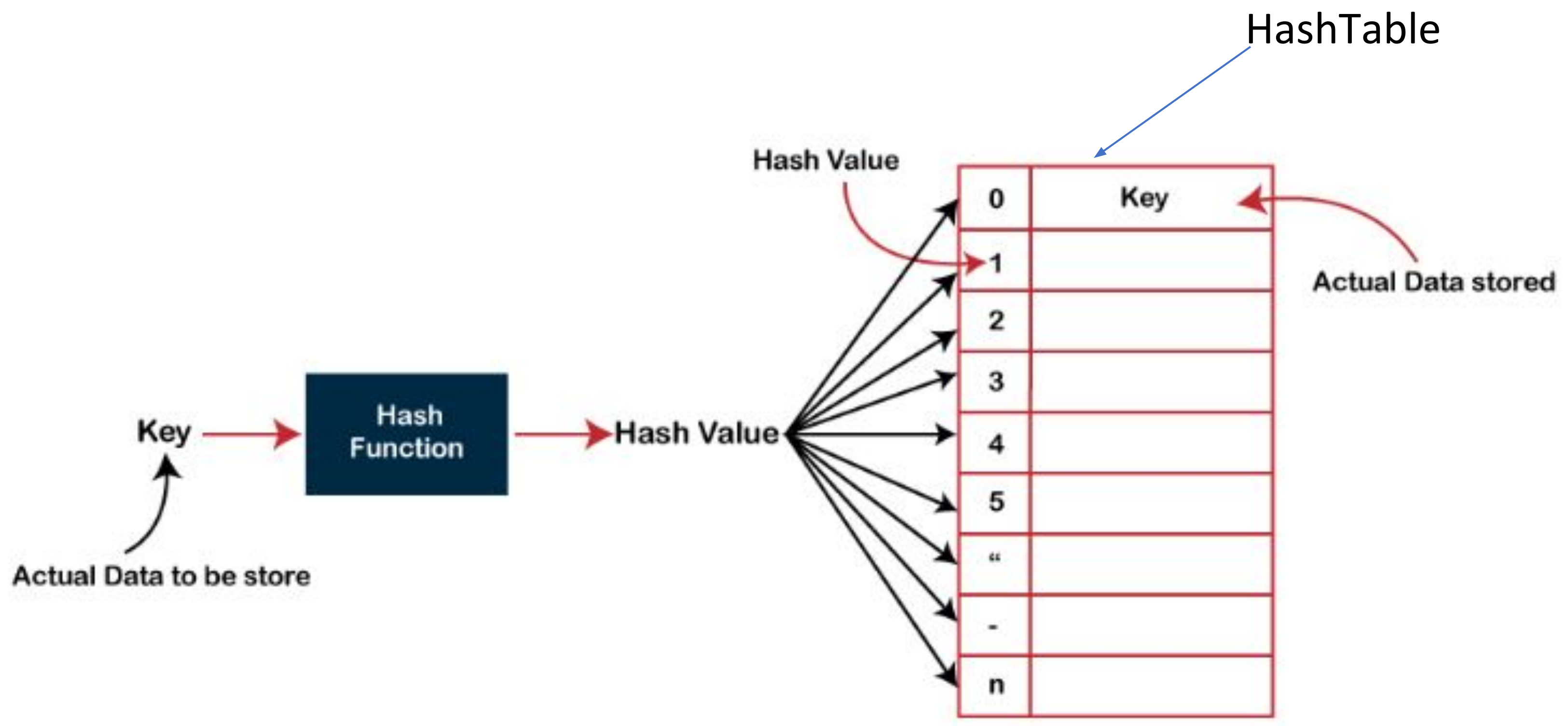
Hashing

Searching

- Two Linear Search Algorithms:
- Linear Search: O(n)
- Binary Search: O(logn)
- What if we want to perform the search operation in time O(1)?
- Answer : Hashsing

Hashing


HashTable
Hash Value
0
Key
1
2
3
4
5
"
-
n
Actual Data stored
Key
Hash Function
Hash Value
Actual Data to be store

Hash Table

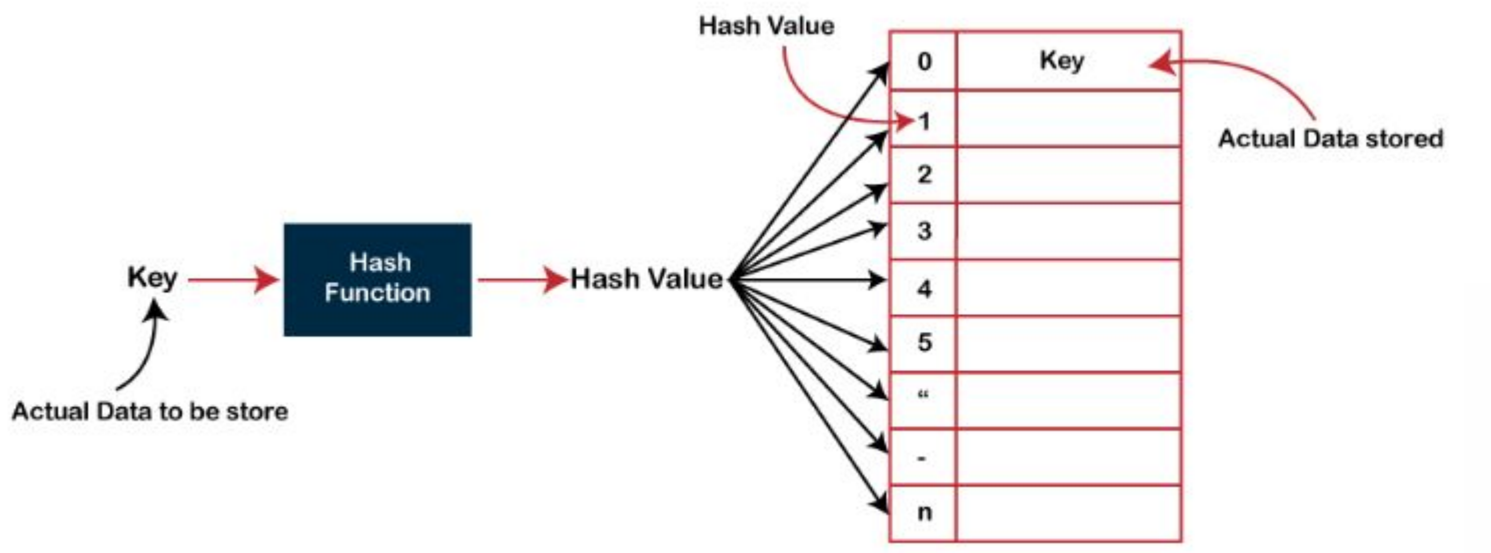
- Hash Table is a data structure which stores data in an associative manner. In a hash table, data is stored in an array format, where each data value has its own unique index value.
- Access of data becomes very fast if we know the index of the desired data.
- it is a data structure in which insertion and search operations are very fast irrespective of the size of the data.
- Hash Table uses an array as a storage medium and uses hash technique to generate an index where an element is to be inserted or is to be located from.

Hash Table

- Hash Table is a data structure in which keys are mapped to array positions by a Hash Function.
- In a hash table, an element with key k is stored at index h(k) and not k.
- Hash function is used to calculate the index at which the element with key k will be stored.
- E.g **Index = hash(key)**

Hashing

- The process of mapping the keys to appropriate locations in a hash table is called hashing.



Hash value

- **Hash value/ code**: The index in the Hash Table for storing the value obtained after computing the Hash Function on the corresponding key.

Hash Functions

- A hash function is a mathematical formula which when applied to a key produce an integer which can be used as an index for the key in the hash table.
- Properties of a Good Hash function.

1 Low cost

2 Determinism

3Uniformaity

Different Hash Functions

1. **Division method**
2. **Folding method**
3. **Mid square method**

Division method

- Simplest method
- $H(k) = k \bmod m$
- M is generally table size.
- K is key value.
- It returns index where the element can be stored.
- E.g calculate hash value of key = 1234 and m =10.
- Then h(1234)=1234 mod 10 = 4
- So key with value 1234 is placed at index 4.

Mid Square Method

- It is a good hash function which works in two steps

1 .Square the value of the key that K^2.

2 Extract the middle r digits of the result obtained in step 1.

Q. Calculate the hash value for keys 1223 using the mid square method. The hash table has 100 memory locations.

- As hash table has 100 memory locations so indices vary from 0 to 99 so two digits are need to map the key to a location in the hash table so r=2.
- K = 1234 ,k^2 = 1522756,h(1234)=27

Folding Method.

- Works in two steps

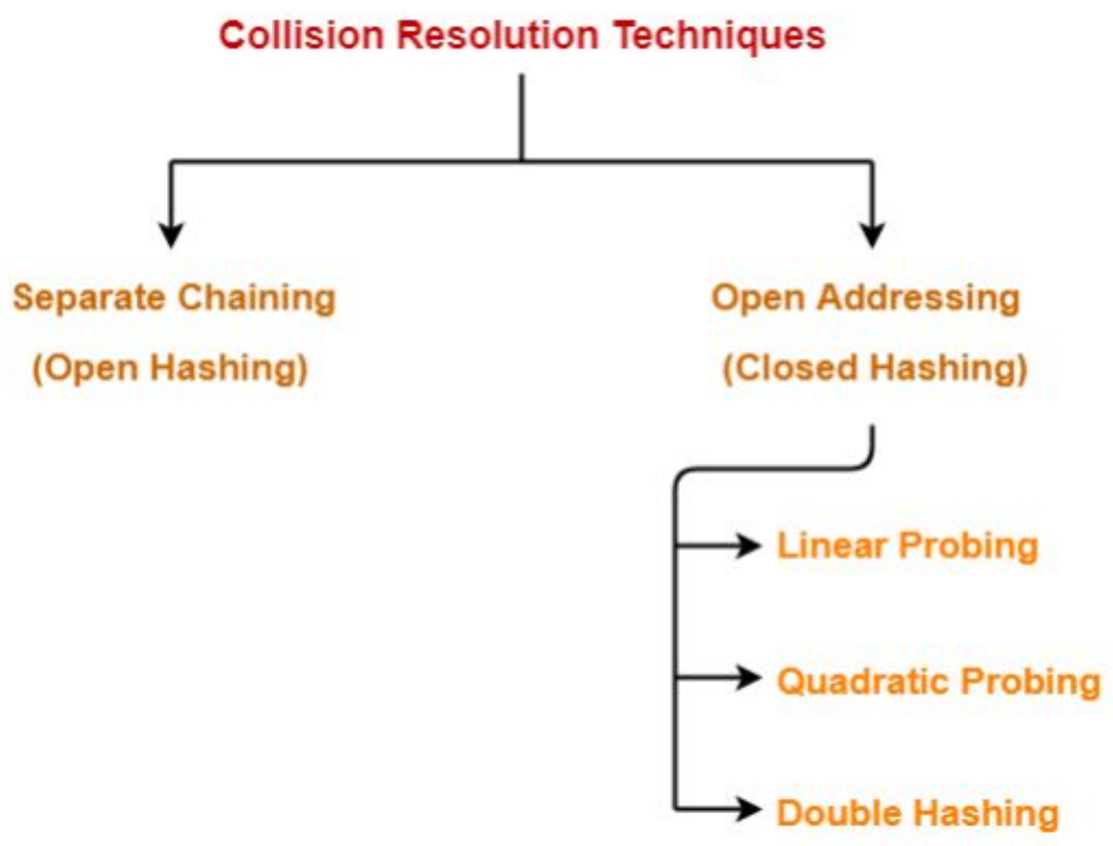
1. Divide the key value into a number of parts. That is divide k into parts k1,k2,k3….kn where each part has the same number of digits except the last part which may have lesser digits than the other parts.

2. Add the individual parts. That is obtain the sum of k1+k2+…+kn. The hash value is produced by ignoring the last carry if any.

- Given a hash table of 100 locations calculate the hash value using folding method.
- Key = 5678
- Part = 56 and 78
- Sum=134
- Hash value =34

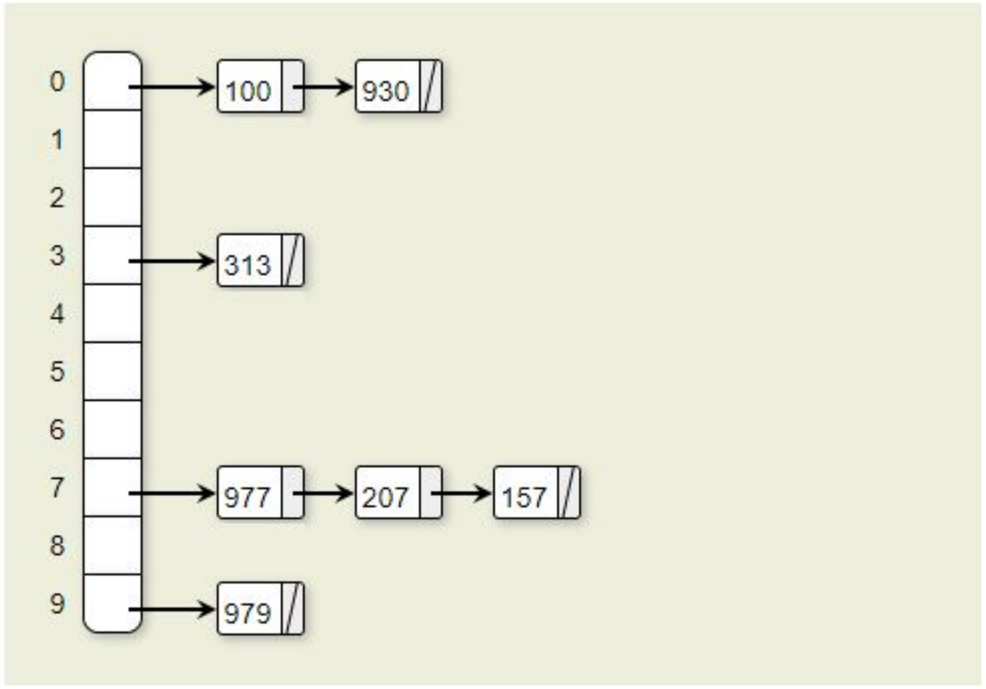
Collision resolution technique

- Collision occur when the hash function maps two different keys to the same locations.
- Collision resolution technique is a method to solve the problem of collision.



Open Hashing

- Open hashing or more widely known as chaining is one of the simplest approaches to avoid collision in hash tables.
- In open hashing ,each location in a hash table stores a pointer to a linked list that contains all the key values that were hashed to that location. That is location l in the hash table points to the head of the linked list of all the key values that shed to l.

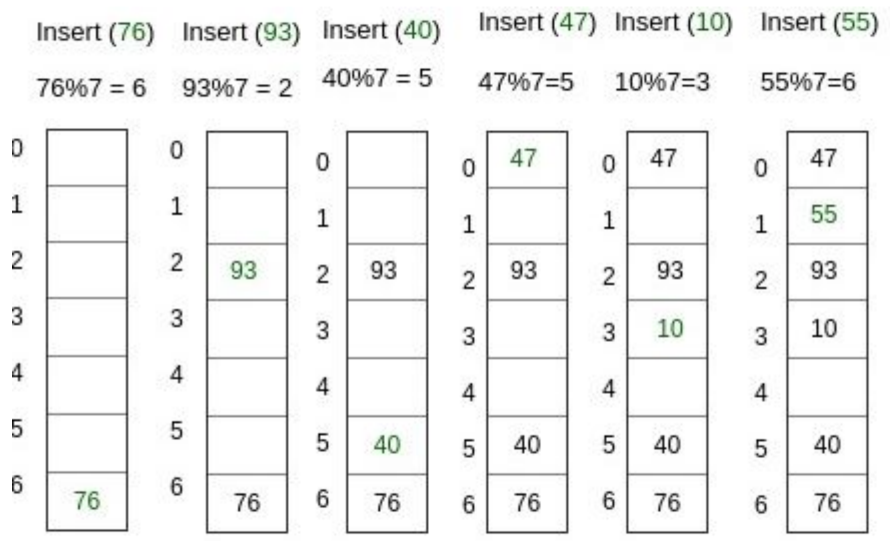


Open Addressing-Closed Hashing

Linear Probing:

- Linear probing is a form of open addressing.
- When the hash function causes a collision by mapping a new key to a cell of the hash table that is already occupied by another key, linear probing searches the table for the closest following free location and inserts the new key there.
- Lookups are performed in the same way, by searching the table sequentially starting at the position given by the hash function, until finding a cell with a matching key or an empty cell.

Linear Probing:



	Insert (76)	Insert (93)	Insert (40)	Insert (47)	Insert (10)	Insert (55)
0				47	47	47
1						55
2		93	93	93	93	93
3					10	10
4						
5			40	40	40	40
6	76	76	76	76	76	76

Quadratic Probing

- In this technique if a value is already stored at a location generated at h(k) ,then the following hash function is used to resolve the collision.
- H(k,i) =[h`(k) + C_1 i + C_2 i^2] mod m
- Where m is the size of hash table.
- H’(k) = k mod m
- C_1 and C_2 are constants.

Double Hashing

- To start with double hashing uses one hash value and then repeatedly steps forward an interval until an empty location is search. The interval is decided using a second independent hash function ,hence the name double hashing.
- We use two hash functions rather than a single function.
- H(k,i)= [$h_1(k)$ + i h_2(k)] mod m

Example

- Suppose you wish to store a set of numbers = {0,1,2,4,5,7} into a hash table of size 5.
- Now, assume that we have a hash function H, such that H(x) = x%5
- So, if we were to map the given data with the given hash function we'll get the corresponding values.

Advantages of Hashing:

- Fast Access: Hashing provides constant time access to data, making it faster than other data structures like linked lists and arrays.
- Efficient Search: Hashing allows for quick search operations, making it an ideal data structure for applications that require frequent search operations.
- Space-Efficient: Hashing can be more space-efficient than other data structures, as it only requires a fixed amount of memory to store the hash table.

Applications of Hashing

- Databases: Hashing is used to index and search large databases efficiently.
- Cryptography: Hash functions are used to generate message digests, which are used to verify the integrity of data and protect against tampering.
- Caching: Hash tables are used in caching systems to store frequently accessed data and improve performance.
- Spell checking: Hashing is used in spell checkers to quickly search for words in a dictionary.
- Network routing: Hashing is used in load balancing and routing algorithms to distribute network traffic across multiple servers.