About

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. Thus, it becomes 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.

Important Terms

Hashing Function

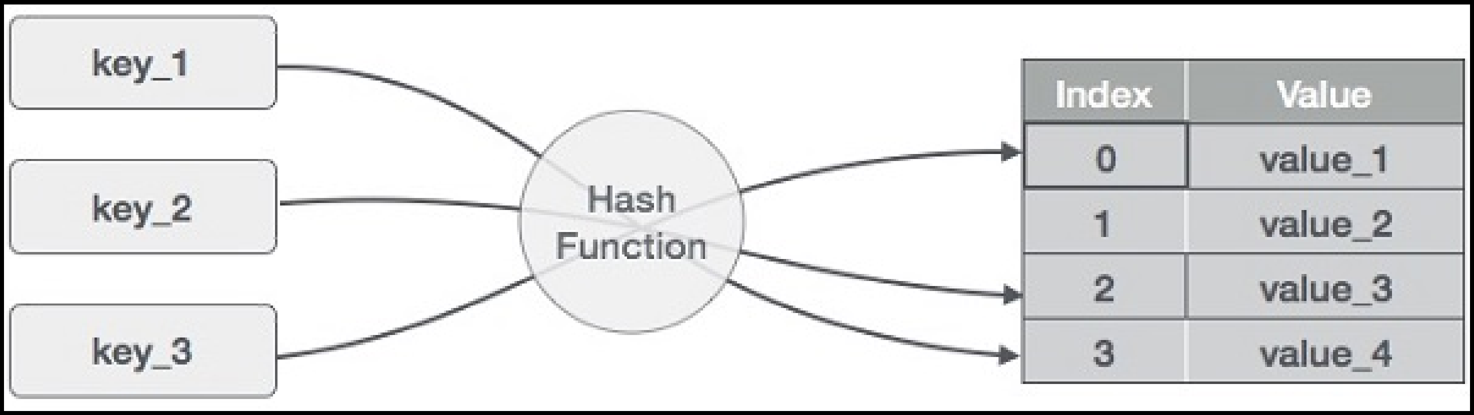
Suppose key is not a number and is a String in that case how will be the hashtable access as it's an array and it needs an index to access and index is integer so in that can Hashing Function is used to convert String key into hash value.

Hash Value

Hash value is the key that is index to search in array and will be obtain from Hashing Function

Collision

There can be many Keys (String) however hash value obtained using that Key can be finite and limited to array and probably chance that 2 different keys have the same hash key. This problem is known as Collision



Solution for Collision

Chaining

Category : Open Hashing
Links :
L-6.3: Chaining in Hashing | What is chaining in hashing with examples
https://www.tutorialspoint.com/hashing-with-chaining-in-data-structure

Linear Probing

Category : Closed Hashing / Open Addressing
Links :
L-6.4: Linear Probing in Hashing with example
https://www.tutorialspoint.com/linear-probing-in-data-structure

Quadratic Probing

Category : Closed Hashing / Open Addressing
Links :
L-6.6: Quadratic Probing in Hashing with example
https://www.tutorialspoint.com/quadratic-probing-in-data-structure

Double Hashing

Category : Closed Hashing / Open Addressing
Links :
L-6.7: Double Hashing | Collision Resolution Technique
https://www.javatpoint.com/double-hashing-in-java

Common Operations

Insert/Put

Adding at index position with specific key
Big O: O(1) constant if no collision

Read/Get

Read From index position with specific key
Big O: O(1) constant if no collision

Delete

Delete at index position with specific key
Big O: O(1) constant if no collision

Common Operations

L-6.1: What is hashing with example | Hashing in data structure
Tutorialspoint