**Hashing**

Hashing is a technique to store indices of a record in a hash table for later search in an efficient manner.

Hashing is done using Hash Table and Hash Function.

**What is Hash Table and Hash Function ?**

Using Hash Function we store the key and the address (address of the key value is in the memory) in the Hash Table so that the user using the hash function can directly go to that index in the hash table and access the data in O(1) time.

Hash Table: internally is an array where the key and address is stored.

Hash Function: Applying some function on the hash code of the key to get the index of the array.

Types of function:

1. Division Method: K % m (Faster Method)

Where k is the hash code of the key

m is the size of the hash table (i.e. size of the array)

1. Multiplication Method |\_m (K\*A - |\_K\*A\_|\_|  
   where A is between 0 to 1

|\_ \_| floor

Now there is a possibility that the hash function generates same index for two or more keys and this is known as collision.

**How to handle Collisions ?**

1. Chaining Method (Open Hashing): A linked list is formed at every index of the array that is array of linked list is created. If collision will occur then the key and address will be stored on the next node in the linked list.

Load Factor = No of elements Stored / Size of the hash table

Load Factor > 1 (100 % collision will occur)

1. Open addressing method (Closed Hashing): Hash sequence is generated in this method, Hash Sequence is tells if the index generated by hash function is not empty then at what the index should move forward.

In this case load factor strictly needs to be less than 1

* 1. Linear probing: If the slot is not empty go to the next one then to the next.

Hash Sequence = (hash Function + i)%m

Where i=1,2,3….

* 1. Quadratic probing: If the slot is not empty then go to the next one exponentially

Hash Sequence = hash Function + i^2

Where i=1,2,3….

Hashing is good when

* Size of hash table is very large.
* The total entries is less.