// Hash Table

Is the process of taking a key (which is a piece of data)

and scrambling with an algorithm (hashing function) to produce an index which is the address of where it will be stored in a hash table.

The key - value pair is the pair of numbers which links the key to the index.

Hash table is a data structure which is indexed. It is where the keys will be stored.

Unlike an array, a hash table would allow data to be stored and accessed quickly and conveniently.

In an array, you can only locate an element by going through the array one by one.

In a hash table, you will be able to exactly where the element is in the hash table by going entering the key into the hash function.

For this to happen, we must make sure for the same input it must always generate the same output.

However, it is two or more keys to have the same value.

This is called a collision, and there are 2 ways to work around it:

Namely:

Open addressing aka probing

Closed addressing aka chaining

Open addressing is when: upon colllision, finding another index which is open.

Open addressing is good as it is very space efficient, it strives to fill up the hash table. However, the process of doing so is inefficient.

* Linear Probing
* Quadratic Probing
* Double Hash

Closed addressing is creating a bucket at each index of the hashtable to store all the keys that has the same index or value.

Closed addressing makes it very easy to add and remove data (or keys). However the extra structure (in the form of a linked list?) demands more memory and processing. This is worse when you have a bad hash function which creates a very long bucket.

An example of hash table being used is a spellchecker, when the user presses space bar, the word is passed through a hash function, if the function returns a key in the hash table, then it is spelled correctly.

<https://www.cs.usfca.edu/~galles/visualization/ClosedHash.html>

<https://www.geeksforgeeks.org/top-20-hashing-technique-based-interview-questions/>