Applications of Hashing

In this article we will be discussing of applications of hashing.

Hashing provides *constant time search, insert and delete operations* on average. This is why hashing is one of the most used data structure, example problems are, distinct elements, counting frequencies of items, finding duplicates, etc.

There are many other applications of hashing, including modern day cryptography hash functions. Some of these applications are listed below:

- Message Digest

- Password Verification

- Data Structures(Programming Languages)

- Compiler Operation

- Rabin-Karp Algorithm

- Linking File name and path together

- Game Boards

- Graphics

Let us see them one by one in detail:

Message Digest:

This is an application of cryptographic Hash Functions. Cryptographic hash functions are the functions which produce an output from which reaching the input is close to impossible. This property of hash functions is called **irreversibility**.

Lets take an **Example**:

Suppose you have to store your files on any of the cloud services available. You have to be sure that the files that you store are not tampered by any third party. You do it by computing "hash" of that file using a Cryptographic hash algorithm. One of the common cryptographic hash algorithms is **SHA 256**. The hash thus computed has a maximum size of 32 bytes. So a computing the hash of large number of files will not be a problem. You save these hashes on your local machine.

Now, when you download the files, you compute the hash again. Then you match it with the previous hash computed. Therefore, you know whether your files were tampered or not. If anybody tamper with the file, the hash value of the file will definitely change. Tampering the file without changing the hash is nearly impossible.

Password Verification:

Cryptographic hash functions are very commonly used in password verification. Let's understand this using an **Example**:

When you use any online website which requires a user login, you enter your E-mail and password to authenticate that the account you are trying to use belongs to you. When the password is entered, a hash of the password is computed which is then sent to the server for verification of the password. The passwords stored on the server are actually computed hash values of the original passwords. This is done to ensure that when the password is sent from client to server, no sniffing is there.

Data Structures(Programming Languages):

Various programming languages have hash table based Data Structures. The basic idea is to create a key-value pair where key is supposed to be a unique value, whereas value can be same for different keys. This implementation is seen in unordered_set & unordered_map in C++, HashSet & HashMap in java, dict in python etc.

Compiler Operation:

The keywords of a programming language are processed differently than other identifiers. To differentiate between the keywords of a programming language(if, else, for, return etc.) and other identifiers and to successfully compile the program, the compiler stores all these keywords in a set which is implemented using a hash table

Rabin-Karp Algorithm:

One of the most famous applications of hashing is the Rabin-Karp algorithm. This is basically a string-searching algorithm which uses hashing to find any one set of patterns in a string. A practical application of this algorithm is detecting plagiarism. To know more about Rabin-Karp also go through Searching for Patterns | Set 3 (Rabin-Karp Algorithm).

Linking File name and path together:

When moving through files on our local system, we observe two very crucial components of a file i.e. file_name and file_path. In order to store the correspondence between file_name and file_path the system uses a map(file_name, file_path)which is implemented using a hash table.

Game Boards:

In a game like Tic-Tac-Toe or chess the position of the game may be stored using hash table

Graphics:

The central problem of storage in a graphics storage of objects. For this, data is organized by hashing. It is also used to make a grid of appropriate size. We store the grid in 1D array as we do incase of sparse matrices. All the point stored in one cell will be stored in the same place. If the three points will store in the same entry, it will contain three points. here hash function is is used to mapping the cell grid to memory location. The key advantage of this method of storage is fast execution of search operation.