

Difference between Static and Dynamic Hashing

Data Structure

DBMS

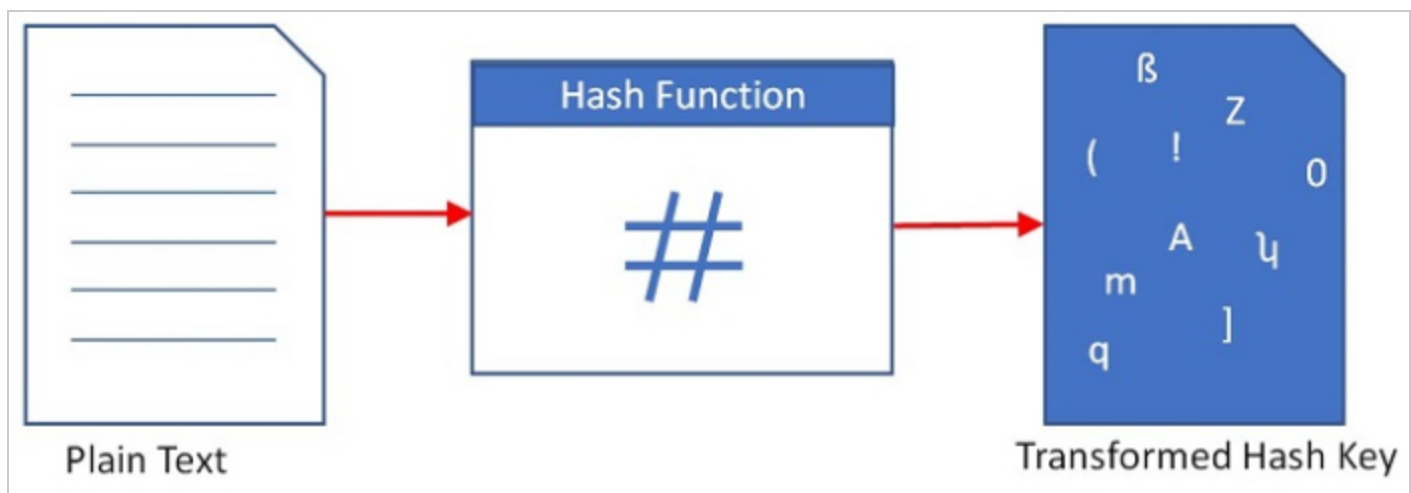
Differences

Hashing is a computation technique in which hashing functions take variable-length data as input and issue a shortened fixed-length data as output. The output data is often called a "Hash Code", "Key", or simply "Hash". The data on which hashing works is called a "Data Bucket".

Characteristics of Hashing Technique

Hashing techniques come with the following characteristics –

- The first characteristic is, hashing technique is deterministic. Means, whatever number of times you invoke the function on the same test variable, it delivers the same fixed-length result.
- The second characteristic is its unidirectional action. There is no way you can use the Key to retrieve the original data. Hashing is irreversible.



What are Hash Functions?

Hash functions are mathematical functions that are executed to generate addresses of data records. Hash functions use memory locations that store data, called 'Data Buckets'.

Hash functions are used in cryptographic signatures, securing privacy of vulnerable data, and verifying correctness of the received files and texts. In computation, hashing is used in data

processing to locate a single string of data in an array, or to calculate direct addresses of records on the disk by requesting its Hash Code or Key.

Applications of Hashing

Hashing is applicable in the following area –

- Password verification
- Associating filename with their paths in operating systems
- Data Structures, where a key-value pair is created in which the key is a unique value, whereas the value associated with the keys can be either same or different for different keys.
- Board games such as Chess, tic-tac-toe, etc.
- Graphics processing, where a large amount of data needs to be matched and fetched.

Database Management Systems where phenomenal records are required to be searched, queried, and matched for retrieval. For example, DBMS used in banking or large public transport reservation software.

Read through this article to find out more about Hashing and specifically the difference between two important hashing techniques – static hashing and dynamic hashing.

What is Static Hashing?

It is a hashing technique that enables users to lookup a definite data set. Meaning, the data in the directory is not changing, it is "Static" or fixed. In this hashing technique, the resulting number of data buckets in memory remains constant.

Operations Provided by Static Hashing

Static hashing provides the following operations –

- **Delete** – Search a record address and delete a record at the same address or delete a chunk of records from records for that address in memory.
- **Insertion** – While entering a new record using static hashing, the hash function (h) calculates bucket address "h(K)" for the search key (k), where the record is going to be stored.
- **Search** – A record can be obtained using a hash function by locating the address of the bucket where the data is stored.

- **Update** – It supports updating a record once it is traced in the data bucket.

Advantages of Static Hashing

Static hashing is advantageous in the following ways –

- Offers unparalleled performance for small-size databases.
- Allows Primary Key value to be used as a Hash Key.

Disadvantages of Static Hashing

Static hashing comes with the following disadvantages –

- It cannot work efficiently with the databases that can be scaled.
- It is not a good option for large-size databases.
- Bucket overflow issue occurs if there is more data and less memory.

What is Dynamic Hashing?

It is a hashing technique that enables users to lookup a dynamic data set. Means, the data set is modified by adding data to or removing the data from, on demand hence the name 'Dynamic' hashing. Thus, the resulting data bucket keeps increasing or decreasing depending on the number of records.

In this hashing technique, the resulting number of data buckets in memory is ever-changing.

Operations Provided by Dynamic Hashing

Dynamic hashing provides the following operations –

- **Delete** – Locate the desired location and support deleting data (or a chunk of data) at that location.
- **Insertion** – Support inserting new data into the data bucket if there is a space available in the data bucket.
- **Query** – Perform querying to compute the bucket address.
- **Update** – Perform a query to update the data.

Advantages of Dynamic Hashing

Dynamic hashing is advantageous in the following ways –

- It works well with scalable data.
- It can handle addressing large amount of memory in which data size is always changing.
- Bucket overflow issue comes rarely or very late.

Disadvantages of Dynamic Hashing

Dynamic hashing comes with the following disadvantage –

- The location of the data in memory keeps changing according to the bucket size. Hence if there is a phenomenal increase in data, then maintaining the bucket address table becomes a challenge.

Differences between Static and Dynamic Hashing

Here are some prominent differences by which Static Hashing is different than Dynamic Hashing –

Key Factor	Static Hashing	Dynamic Hashing
Form of Data	Fixed-size, non-changing data.	Variable-size, changing data.
Result	The resulting Data Bucket is of fixed-length.	The resulting Data Bucket is of variable-length.
Bucket Overflow	Challenge of Bucket overflow can arise often depending upon memory size.	Bucket overflow can occur very late or doesn't occur at all.
Complexity	Simple	Complex

Conclusion

Hashing is a computation technique that uses mathematical functions called Hash Functions to calculate the location (address) of the data in the memory. We learnt that there are two different hashing functions namely, Static hashing and Dynamic hashing.

Each hashing technique is different in terms of whether they work on fixed-length data bucket or a variable-length data bucket. Selecting a proper hashing technique is required by considering the amount of data needed to be handled, and the intended speed of the application.