Hash Tabel

A hash table is a data structure where data is stored in an associative manner. The data is mapped to array positions by a hash function that generates a unique value from each key.

Hash Table

Hash Table

Jorge Stolfi [CC BY-SA 3.0]

The value stored in a hash table can be searched in O(1) time, by using the same hash function which generates an address from the key. The process of mapping the keys to appropriate locations (or indices) in a hash table is called hashing.

Advantages of Hashing

The main advantage of hash tables over other data structures is speed The access time of an element is on average O(1), therefore lookup could be performed very fast. Hash tables are particularly efficient when the maximum number of entries can be predicted in advance.

Hash Functions

A hash function is a mathematical formula which, when applied to a key, produces a value which can be used as an index for the key in the hash table.

The main aim of a hash function is that elements should be uniformly distributed. It produces a unique set of integers within some suitable range in order to reduce the number of collisions.

Properties of a Good Hash Function

Uniformity

A good hash function must map the keys as evenly as possible. This means that the probability of generating every hash value in the output range should roughly be the same. This also helps in reducing collisions.

Deterministic

A hash function must always generate the same hash value for a given input value.

Low Cost

The cost of executing a hash function must be small so that using the hashing technique becomes preferable over other traditional approaches.

Applications in Programming

Identification Databases: A hash function can make a unique signature from never changing data like our Date of Birth. This can then be used in combination with other variables to uniquely identify a person.

Search Engines: As the number of pages to be crawled is huge, a hash function can be used to determine if the page is unique or it had already been crawled before, without comparing the contents of the whole webpage.

Different Hash Functions

Division Method

This is the most simple method of hashing. Any integer, for example, x is divided by a number M and the remainder obtained is used as the hash.

Generally, M is chosen to be a prime number because a prime number increases the likelihood that the keys are mapped with uniformity in the output range of values.

This function could be represented as:

h(k) = k mod M

Multiplication Method

The Multiplication method has the following steps:

A constant is chosen which is between 0 and 1, say A.

The key k is multiplied by A.

The fractional part of kA is extracted.

The result of Step 3 is multiplied by the size of the hash table (m).

This can be represented as:

h(k) = fractional_part[m(kA mod 1)]

Mid-Square Method

The Mid-Square method is as follows:

The value of the key is squared. That is, k^2 is found.

The middle r digits of the result are extracted.

The result r is the hash obtained.

The algorithm works well because most or all digits of the key-value contribute to the resulting hash.

Collisions

Collisions occur when the hash function maps two different keys to the same location. Two records cannot be stored in the same location of a hash table normally.

The method used to solve the problem of collisions is called the collision resolution technique.

There are two popular collision resolution techniques:

Open Addressing

Hash collision resolved by separate chaining

Hash collision resolved by open addressing

Jorge Stolfi [CC BY-SA 3.0]

Once a collision takes place, open addressing (also known as closed hashing)computes new positions using a probe sequence and the next record is stored in that position. There are some well-known probe sequences:

Linear Probing: The interval between the probes is fixed to 1. This means that the very next available position in the table would be tried.

Quadratic Probing: The interval between the probes increases quadratically. This means that the next available position that would be tried would increase quadratically.

Double Hashing: The interval between probes is fixed for each record but the hash is computed again by double hashing.

Chaining

Chaining is another solution to the problem of collisions.

Hash collision resolved by separate chaining

Hash collision resolved by separate chaining

Jorge Stolfi [CC BY-SA 3.0]

In chaining, each location in a hash table stores a pointer to a linked list that contains all the key values that were hashed to that location. As new collisions occur, the linked list grows to accommodate those collisions forming a chain.

This effectively means that each location in the hash table is not limited to store one value. Searching for a value in a chained hash table is as simple as scanning a linked list for an entry with the given key.

Insertion operation appends the key to the end of the linked list pointed by the hashed location.

Deleting a key requires searching the list and removing the element.

This solution, however, presents a problem if the linked list becomes large enough that it takes O(n) time to search one position. This occurs if the hash table is too small and has to accommodate many values.