Data Structures and Algorithms

Lecture 39: Hashing

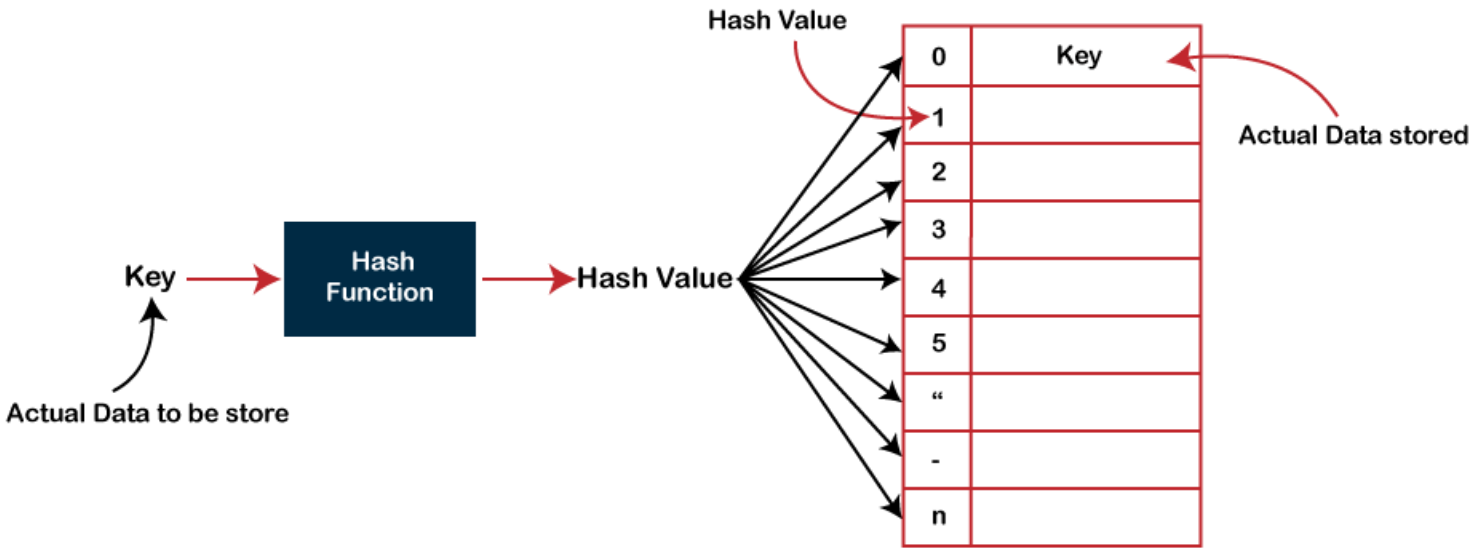
Hashing

- Hashing is the process of mapping large amount of data item to smaller table with the help of hashing function.
- Hashing is also known as Hashing Algorithm or Message Digest Function.
- It is a technique to convert a range of key values into a range of indexes of an array.
- It is used to facilitate the next level searching method when compared with the linear or binary search.
- Hashing is used with a database to enable items to be retrieved more quickly.
- It is used in the encryption and decryption of digital signatures.

Hashing

- In Hashing technique, the hash table and hash function are used. Using the hash function, we can calculate the address at which the value can be stored.
- The main idea behind the hashing is to create the (key/value) pairs. If the key is given, then the algorithm computes the index at which the value would be stored. It can be written as:

Index = hash(key)



Hashing

Examples of Hashing in Data Structure:

The following are real-life examples of hashing in the data structure –

- In schools, the teacher assigns a unique roll number to each student. Later, the teacher uses that roll number to retrieve information about that student.

- A library has an infinite number of books. The librarian assigns a unique number to each book. This unique number helps in identifying the position of the books on the bookshelf.

Hash function

Example: Need to store some items (arranged in a key-value pair) inside a hash table with 30 cells. The values are: (3,21) (1,72) (40,36) (5,30) (11,44) (15,33) (18,12) (16,80) (38,99). The hash table will look like the following:

Serial Number	Key	Hash	Array Index
1	3	3%30 = 3	3
2	1	1%30 = 1	1
3	40	40%30 = 10	10
4	5	5%30 = 5	5
5	11	11%30 = 11	11
6	15	15%30 = 15	15
7	18	18%30 = 18	18
8	16	16%30 = 16	16
9	38	38%30 = 8	8

Hash Function

Hash Function is used to index the original value or key and then used later each time the data associated with the value or key is to be retrieved. Thus, hashing is always a one-way operation. There is no need to "reverse engineer" the hash function by analyzing the hashed values.

Characteristics of Good Hash Function:

- The hash value is fully determined by the data being hashed.
- The hash Function uses all the input data.
- The hash function "uniformly" distributes the data across the entire set of possible hash values.
- The hash function generates complicated hash values for similar strings.

Hash Function

There are three ways of calculating the hash function:

- **Division method**
- **Folding method**
- **Mid square method**

In the division method, the hash function can be defined as:

$$h(k_i) = k_i \% m;$$

where **m** is the size of the hash table.

For example, if the key value is 6 and the size of the hash table is 10. When we apply the hash function to key 6 then the index would be:

h(6) = 6%10 = 6, the index is 6 at which the value is stored.

Hash Function

1. Division Method:

Choose a number m smaller than the number of n of keys in k (The number m is usually chosen to be a prime number or a number without small divisors, since this frequently a minimum number of collisions).

The hash function is:

$$h(k) = k \bmod m$$

$$h(k) = k \bmod m + 1$$

For Example: if the hash table has size m = 12 and the key is k = 100, then h (k) = 4.

Since it requires only a single division operation, hashing by division is quite fast.

Hash Function

2. Multiplication Method:

The multiplication method for creating hash functions operates in two steps. First, we multiply the key k by a constant A in the range $0 < A < 1$ and extract the fractional part of kA. Then, we increase this value by m and take the floor of the result.
The hash function is:

$$h(k) = \lfloor m(k A \bmod 1) \rfloor$$

where "k A mod 1" means the fractional part of k A, that is, k A -⌊k A⌋.

Hash Function

3. Mid Square Method:

The key k is squared. Then function H is defined by

$$H(k) = L$$

where L is obtained by deleting digits from both ends of k2. We emphasize that the same position of k2 must be used for all of the keys.

Hash Function

4. Folding Method:

It breaks up a key value into precise segments that are added to form a hash value, and look at another technique is to apply a multiplicative hash function to each segment individually before adding. Some folding methods go one step further and reverse every other piece before the addition. This folding method is independent of distribution.

Hash Function

4. Folding Method:

Algorithm:

- The folding method is used for creating hash functions starts with the item being divided into equal-sized pieces i.e., the last piece may not be of equal size.
- The outcome of adding these bits together is the hash value, H(x) = (a + b + c) mod M, where a, b, and c represent the preconditioned key broken down into three parts and M is the table size, and mod stands for modulo.
- In other words, the sum of three parts of the preconditioned key is divided by the table size. The remainder is the hash key.

Hash Function

Folding Method - Example 1:

The task is to fold the key 452378912 into a Hash Table of ten spaces (0 through 9).

- It is given that the key, say X is 452378912 and the table size (i.e., M = 10).
- Since it can break X into three parts in any order. Let's divide it evenly.
- Therefore, a = 452, b = 378, c = 912.
- Now, H(x) = (a + b + c) mod M i.e., H(452378912) =(452 + 378 + 912) mod 10 = 1742 mod 10 = 2.
- Hence, 452378912 is inserted into the table at address 2.

Hashing

Drawback of Hash function:

A Hash function assigns each value with a unique key. Sometimes hash table uses an imperfect hash function that causes a collision because the hash function generates the same key of two different values.