**Hash Functions and list/types of Hash functions**

Hashing is the process of generating a value from a text or a list of numbers using a mathematical function known as a hash function.

A **Hash Function**is a function that converts a given numeric or alphanumeric key to a small practical integer value. The mapped integer value is used as an index in the hash table. In simple terms, a hash function **maps** a significant number or string to a small integer that can be used as the **index** in the hash table.

The pair is of the form **(key, value)**, where for a given key, one can find a value using some kind of a “function” that maps keys to values. The key for a given object can be calculated using a function called a hash function. For example, given an array A, if i is the key, then we can find the value by simply looking up A[i].

**Types of Hash functions**

There are many hash functions that use numeric or alphanumeric keys. This article focuses on discussing different hash functions:

1. **Division Method.**
2. **Mid Square Method.**
3. **Folding Method.**
4. **Multiplication Method.**

Let’s begin discussing these methods in detail.

**1. Division Method:**

This is the most simple and easiest method to generate a hash value. The hash function divides the value k by M and then uses the remainder obtained.

**Formula:**

|  |
| --- |
| ***h(K) = k mod M***  *Here,* ***k****is the key value, and* ***M****is the size of the hash table.* |

It is best suited that **M**is a prime number as that can make sure the keys are more uniformly distributed. The hash function is dependent upon the remainder of a division.

**Example:**

|  |
| --- |
| *k = 12345 M = 95 h(12345) = 12345 mod 95                 = 90*  *k = 1276 M = 11 h(1276) = 1276 mod 11* |

**Pros:**

1. This method is quite good for any value of M.
2. The division method is very fast since it requires only a single division operation.

**Cons:**

1. This method leads to poor performance since consecutive keys map to consecutive hash values in the hash table.
2. Sometimes extra care should be taken to chose value of M.

**2.Mid Square Method:**

The mid square method is a very good hashing method. It involves two steps to compute the hash value-

1. Square the value of the key k i.e. k2
2. Extract the middle **r** digits as the hash value.

**Formula:**

|  |
| --- |
| ***h(K) = h(k x k)***  *Here,* ***k****is the key value.* |

The value of **r**can be decided based on the size of the table.

**Example:**

Suppose the hash table has 100 memory locations. So r = 2 because two digits are required to map the key to the memory location.

|  |
| --- |
| *k = 60 k x k = 60 x 60         = 3600 h(60) = 60*  *The hash value obtained is 60* |

**Pros:**

1. The performance of this method is good as most or all digits of the key value contribute to the result. This is because all digits in the key contribute to generating the middle digits of the squared result.
2. The result is not dominated by the distribution of the top digit or bottom digit of the original key value.

**Cons:**

1. The size of the key is one of the limitations of this method, as the key is of big size then its square will double the number of digits.
2. Another disadvantage is that there will be collisions but we can try to reduce collisions.
3. **Digit Folding Method:**

This method involves two steps:

1. Divide the key-value **k**into a number of parts i.e. **k1, k2, k3,….,kn**, where each part has the same number of digits except for the last part that can have lesser digits than the other parts.
2. Add the individual parts. The hash value is obtained by ignoring the last carry if any.

**Formula:**

|  |
| --- |
| ***k = k1, k2, k3, k4, ….., kn******s = k1+ k2 + k3 + k4 +….+ kn******h(K)= s***  *Here,* ***s****is obtained by adding the parts of the key****k*** |

**Example:**

|  |
| --- |
| *k = 12345 k1 = 12, k2 = 34, k3 = 5 s = k1 + k2 + k3   = 12 + 34 + 5   = 51  h(K) = 51* |

**Note:**  
The number of digits in each part varies depending upon the size of the hash table. Suppose for example the size of the hash table is 100, then each part must have two digits except for the last part that can have a lesser number of digits.

1. **Multiplication Method**

This method involves the following steps:

1. Choose a constant value A such that 0 < A < 1.
2. Multiply the key value with A.
3. Extract the fractional part of kA.
4. Multiply the result of the above step by the size of the hash table i.e. M.
5. The resulting hash value is obtained by taking the floor of the result obtained in step 4.

**Formula:**

|  |
| --- |
| ***h(K) = floor (M (kA mod 1))***  *Here,* ***M****is the size of the hash table.* ***k****is the key value.* ***A****is a constant value.* |

**Example:**

|  |
| --- |
| *k = 12345 A = 0.357840 M = 100*  *h(12345) = floor[ 100 (12345\*0.357840 mod 1)]                = floor[ 100 (4417.5348 mod 1) ]                = floor[ 100 (0.5348) ]                = floor[ 53.48 ]                = 53* |

**Pros:**

The advantage of the multiplication method is that it can work with any value of between 0 and 1, although there are some values that tend to give better results than the rest.

**Cons:**

The multiplication method is generally suitable when the table size is the power of two, then the whole process of computing the index by the key using multiplication hashing is very fast.