

Hash Functions

The following are some of the Hash Functions –

Division Method

This is the easiest method to create a hash function. The hash function can be described as –

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

Here, $h(k)$ is the hash value obtained by dividing the key value k by size of hash table n using the remainder. It is best that n is a prime number as that makes sure the keys are distributed with more uniformity.

An example of the Division Method is as follows –

```
k=1276  
n=10  
h(1276) = 1276 mod 10  
= 6
```

The hash value obtained is 6

A disadvantage of the division method is that consecutive keys map to consecutive hash values in the hash table. This leads to a poor performance.

Multiplication Method

The hash function used for the multiplication method is –

$$h(k) = \text{floor}(n(kA \bmod 1))$$

Here, k is the key and A can be any constant value between 0 and 1. Both k and A are multiplied and their fractional part is separated. This is then multiplied with n to get the hash value.

An example of the Multiplication Method is as follows –

```
k=123  
n=100  
A=0.618033
```

```
h(123) = 100 (123 * 0.618033 mod 1)
= 100 (76.018059 mod 1)
= 100 (0.018059)
= 1
```

The hash value obtained is 1

An advantage of the multiplication method is that it can work with any value of A, although some values are believed to be better than others.

Mid Square Method

The mid square method is a very good hash function. It involves squaring the value of the key and then extracting the middle r digits as the hash value. The value of r can be decided according to the size of the hash table.

An example of the Mid Square Method is as follows –

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

```
k = 50
k*k = 2500
h(50) = 50

The hash value obtained is 50
```

Folding Method in Hashing:

Folding Method in Hashing: It breaks up a key value into precise segments that are added to form a hash value.

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, $H(x) = (a + b + c) \bmod M$, 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.

Example 1: The task is to fold the key **123456789** into a Hash Table of ten spaces (0 through 9).

- It is given that the key, say **X** is 123456789 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 = 123, b = 456, c = 789.
- Now, **$H(x) = (a + b + c) \bmod M$** i.e., $H(123456789) = (123 + 456 + 789) \bmod 10 = 1368 \bmod 10 = 8$.
- Hence, 123456789 is inserted into the table at address **8**.