**Hashing:** Hashing is a process of accessing a record by mapping a key value into a position in the Hash Table.

A Hash table is a Data Structure in which the location of a data item is determined directly as a function of the data item itself. Time required to locate a data item in a hash table is O(1).

A Hash function is a mathematical formula that manipulates the key in some form to compute the index for this key in the hash table.

**Different Hash functions:**

**1. Division Method:** Here the Hash function is:

H(k) = k mod m

Here H(k) is the Hash function, k is the key item that we want to insert in the Hash table and m is the size of the Hash table.

Example: Insert the following items into a hash table of size 11

5, 28, 19, 15, 20, 33, 12, 17

Hash Table of size 11

|  |
| --- |
| 33 |
| 12 |
|  |
|  |
| 15 |
| 5 |
|  |
|  |
| 19 |
| 20 |
|  |

Key: 5 H(5)=5 mod 11= 5 0

Key: 28 H(28)=28 mod 11=6 1

Key: 19 H(19)= 19 mod 11=8

Key: 15 H(15)= 15 mod 11= 4

Key: 20 H(20)= 20 mod 11= 9

17

28

Key: 33 H(33)=33 mod 11=0

Key: 12 H(12)= 12 mod 11= 1

Key: 17 H(17)= 17 mod 11= 6

**Collision:** A Collision is a phenomenon that occurs when more than one key wants to map to same slot in the Hash table.

**Collision Removal Techniques:**

**1. Open Addressing:** This technique can be classified into two different types:

**I.** **Linear Probing:** This technique searches for a free space in the Hash table until and unless the key item is placed in a vacant position in the Hash table. Linear Probing uses the following hash function:

H(k,i) = [H1(k) + i] mod m for i=0,1,2,…

Here m is the size of the hash table and H1(k) = k mod m is the basic hash function and i is the probe number.

Example: Insert the following items into a Hash table:

76, 26, 37, 59, 21, 65, 88, 19

Let us consider the size of the Hash table m as: 11

|  |
| --- |
|  |
|  |
|  |
|  |
| 26 |
| 37 |
| 59 |
|  |
|  |
|  |
| 76 |

Key: 76

H(76,0) = (76 mod 11 + 0) mod 11 = 10

Key: 26

H(26,0) = (26 mod 11 + 0) mod 11 = 4

Key: 37

H(37,0) = (37 mod 11 + 0) mod 11 = 4

Collision occurred.

H(37,1)= (37 mod 11 + 1) mod 11 = 5

Key: 59

H(59, 0) = (59 mod 11 + 0) mod 11 = 4

Collision Occurred.

H(59, 1) = (59 mod 11 + 1) mod 11 = 5

Again Collision Occurred

H(59, 2)= (59 mod 11 + 2) mod 11 = 6

**II. Double Hashing:** Here the Hash function is:

H(k,i) = [H1(k) + iH2(k)] mod m for i=0,1,2,,….

Where m is the size of the hash table; H1(k) = k mod m and H2(k) = k mod (m-1)

Example: Insert the following items in a hash table:

76, 26, 37, 59, 21, 65, 88, 19

Let us consider the size of the hash table as 11

Key= 76

H(76,0) = [76 mod 11 + 0\*76 mod 10] mod 11

= 10

Key =26

H(26, 0) = [26 mod 11 + 0\*26 mod 10] mod 11

= 4

|  |
| --- |
| 37 |
|  |
|  |
|  |
| 26 |
|  |
|  |
|  |
|  |
|  |
| 76 |

Key = 37

H(37, 0) = [37 mod 11 + 0\*37 mod 10] mod 11

= 4 (Collision Occurred)

H(37, 1) = [37 mod 11 + 1\*37 mod 10] mod 11

= 0

…………………………………….

…………………………………….

**2. Chaining:** If collision occurred then in this technique we need to create a linked list from that index. The drawback of this technique is that if the size of the hash table is very small then maximum no of items will try to fit at the same index. Then in the created linked list, searching time will increase to again O(n).

**Another Hash Function:**

**Mid Square Method:**

K=18 K2 = 324 = Here the index is at 2

K=79 K2 = 6241= Here the index may be 2 or 4