"Hashing"

The process of mapping large amount of data into a smaller table is called hashing. It performs insertion, deletion and finds in constant average time. Hashing is relatively easy to program as compared to trees. However it is based on arrays, hence it has difficulty on expanding.

Sequential search, binary search and all the search trees are totally dependent on no. of element and may key comparisons are involved. Now, our need is to search the element in constant time and loss key comparisons should be involved.

Suppose all the elements are in array of size 'N'. Let us take all the keys are unique and in the rage 'o' to N-1. Now we are sorting the record in array based on key, where array index and keys are same. Then we can access the record in constant time and no key comparisons are involved.

Consider 5 records where keys are :-

9, 4, 6, 7, 2

The keys can be stored in array up

Arr [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

2 4 6 7 9

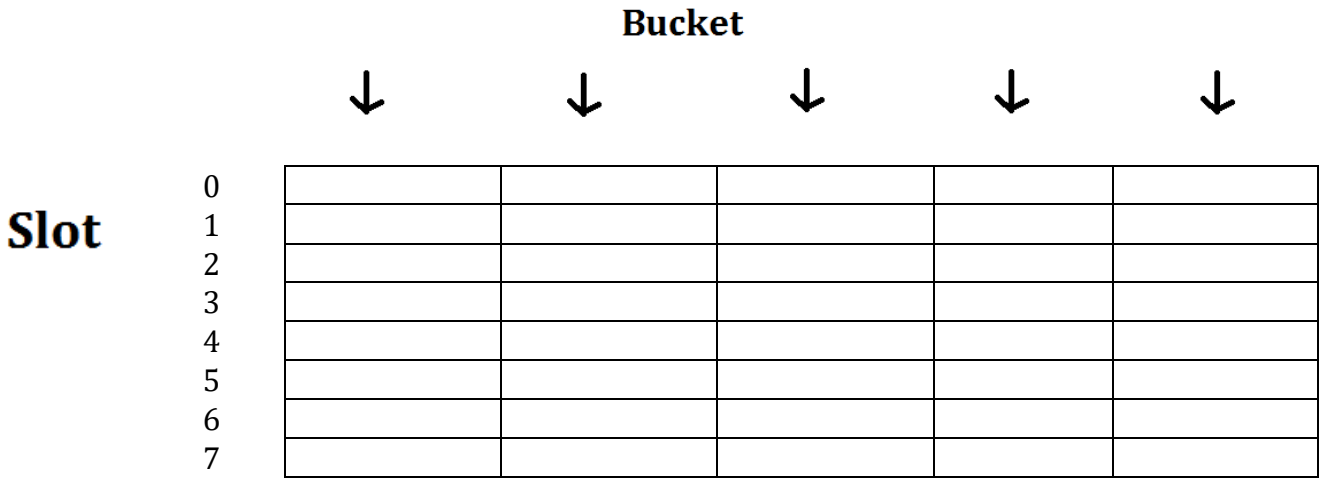
Here, we can see the record which has key value can be directly accessed through array index.

In hashing key is converted into array index and records are kept in array. In the same way for searching the record, key are converted into array index and get the records from array.

For storing records:-	For accessing record:-
Key ↓ Generate array index Key ↓ Stored the record on that array index.	Key ↓ Generate array index Key ↓ Get the records from the array index

Hash Table

A hash table is a data structure where we store a key value after applying the hash function. It is arranged in the form of an array i.e. addressed via a hash function. A hash table is divided into a number of buckets and each bucket is in turn capable of storing a numbers of records. We can say that a bucket has number of slots which is capable of holding one record.The array which supports hashing for storing records or searching records is called hash table.



Hash Function:

A function that transforms a key into a table index is called hash function. If h is a hash and k is key then h(K) is called the hash function or hash of the key 'K'. It is the index in which the key k should be placed.Each key is mapped on a particular array index through hash function. If each key is mapped on a unique hash table then this situation is ideal but there may be possibility that hash function generating some hash table address for different keys & this situation is called collision.

For missing collision we have to perform:

Choose a good hash table which perform minimum collision. Resolving the collision.

Technique for choosing hash function/Method of hashing:

Trunction method:-

In this method a part of hey is considered as address, it can be some rightmost digit or leftmost digit.

Q. apply the truncation method to get the hash index of table size of 100 for following keys.
82394561, 87139465, 83567271, 85943228
Sol^{n}:
Table size = 100
Then, Take 2 rightmost digits for getting the hash table address.

Key value	address
82394561	61
87139465	65
83567271	71
85943228	28

Example.h(key) = a
h(823994561) = 61

Modulus method/Division method:-

Modular method is best way for getting address from key .Take the key, do the modulus operation & get the remainder as address for hash table in order to minimize the collision table size should a prime number consider keys.

Choose a number 'm(nearest prime number of the table size)'. Divide key to be mapped by the number m to calculate the remainder. Generally, the prime number is choosen to reduce the number of collision.

H(k) = k mod 2

Mid square method:-

In this method we square the key, after getting number we take some middle of that number as an address, suppose keys are 4 digits and maximum address = 100.

Key value	**square**	**address**	**H(k)=L**
1456	02119936	19	h(1456) = 19
1892	03579664	79	h(1892) = 79

Here, The key is squared then the hash function is defined by H(k)=L, where L is obtained by deleting the digits from both the ends of key square.

Folding Method:

A key is partition into a number of parts k1,k2,k3…..kn where each parts are added together and last carry is ignored. The hash function is defined as:

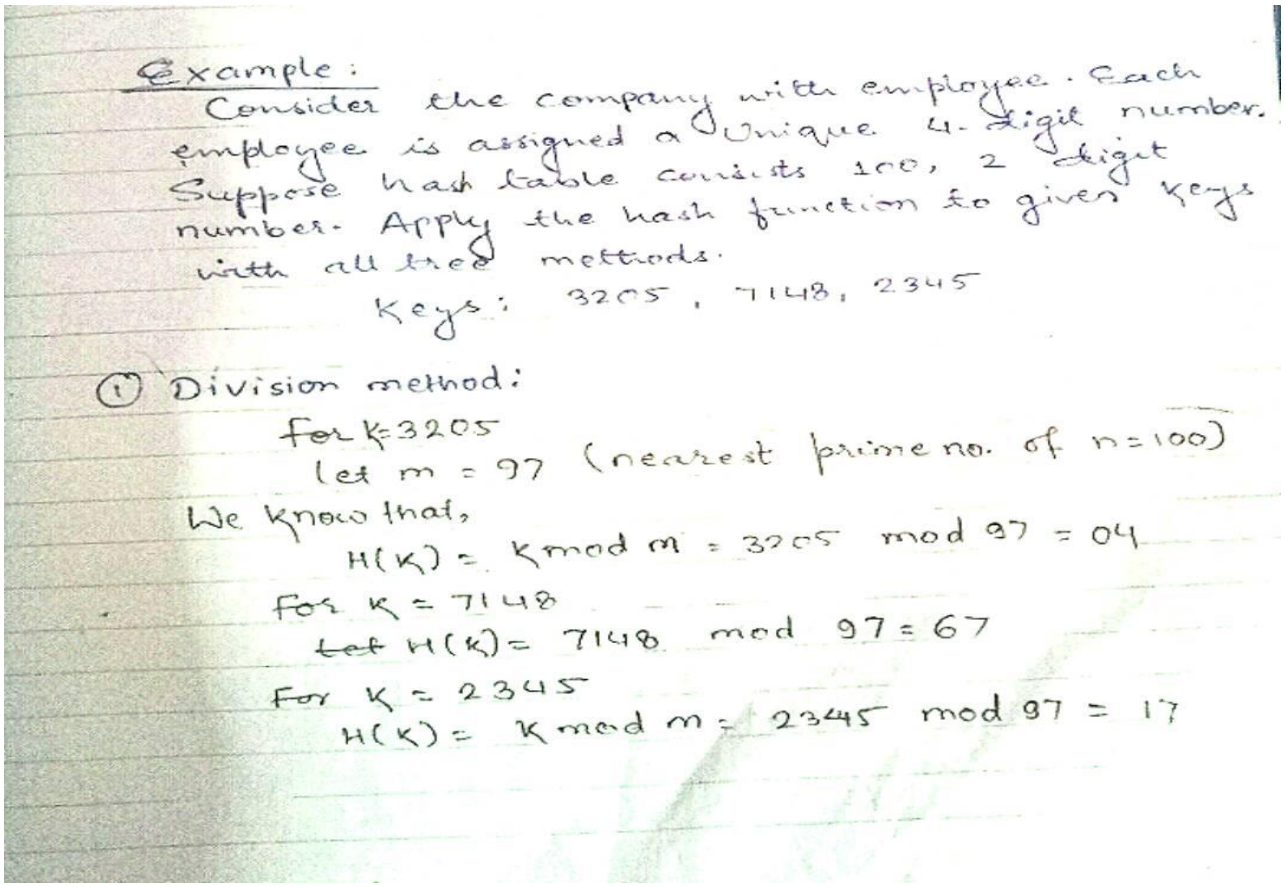
H(k) = k1+k2+k3…..+kn.

For example: Suppose we have 4 digit key 9128 and hash table consists 100, 2digit number.

Here, For k= 9128

We have , H(k) = k1+k2+3…+kn

H(9128)= 91+28 = 119 =19(discard last carry1)



② Mid- square method:

Here, for $k = 3205$

Squaring 3205 we get:

$k^2 = 10272025$

$\therefore H(k) = 72$

for $k = 7148$

$k^2 =$

$\therefore H(k) =$

for $k = 2345$

$k^2 =$

$\therefore H(k) =$

③ Folding method:

Here, for $k = 3205$

We have, $H(k) = k_1 + k_2 + \cdots + k_n$

$$\begin{array}{r} 32 \\ +\ 05 \\ \hline 37 \end{array}$$

for $k = 7148$

$H(k) = 71 + 48 = 119 = 19$ (discard)

for $k = 2345$

$H(k) = 23 + 45 = 68$

Imp

Collision :

(In real practice an ideal (enter) access to the table is rarely found where each key have unique address in a hash table. There may arise a situation when more than one different data item (key) may fall the same array index. In which we are trying to store a data but the place may be already occupied by the other data is called collision.

↳ So it is needed to establish some methods of handling collision. Such a way both data can be store and can be retrived. This process / method is known as collision resolution.

resolution of type of collision

① Open Addressing (Closed hashing):

In Open Addressing technique, the amount of space available for storing data is already fixed by declaring a fixed array for the hash table. There are two ways of open addressing method for collision resolution:

a) Linear probing:

This is a simplest method for collision resolution. When collision occurs while we are inserting a new item into the table, we simply probe forward in the array one step at a time until we find an empty slot and we can store a new data.

b) Quadratic probing:

In quadratic probing method, if a collision occurs at position 'i', location $(i+1^2, i+2^2, i+3^2, \ldots i+j^2)$ where $j = 1, 2, 3, \ldots n$ are tested until an empty slot is found.

2) Chaining:

Hashing with chaining is an application of linked list and gives an approach to collision resolution. In hashing with chaining, the hash table content link list of elements. The list are refer to as chains and the technique is called chaining. Each linked list contains all the elements whose keys has the same index.

3) Dehashing:

It is a technique to apply different hashing function to a key when a collision occurs. We apply the second hash function to 'x' and probe at distance $h_2(x)$, 2 hash 2(x), 3 hash 2(x), ... so on. i.e. $h_2(x)$, $2h_2(x)$, $3h_2(x)$...

Q. For the given elements:

{4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function $h(x) = x \bmod 10$ show the following

a. Separate chaining hash table
b. Open addressing hash table

i) Using linear probing
ii) Using quadratic probing
@ Open addressing hash table with second hash function : $h_2(x) = 7 - x \bmod 7$.

⟶

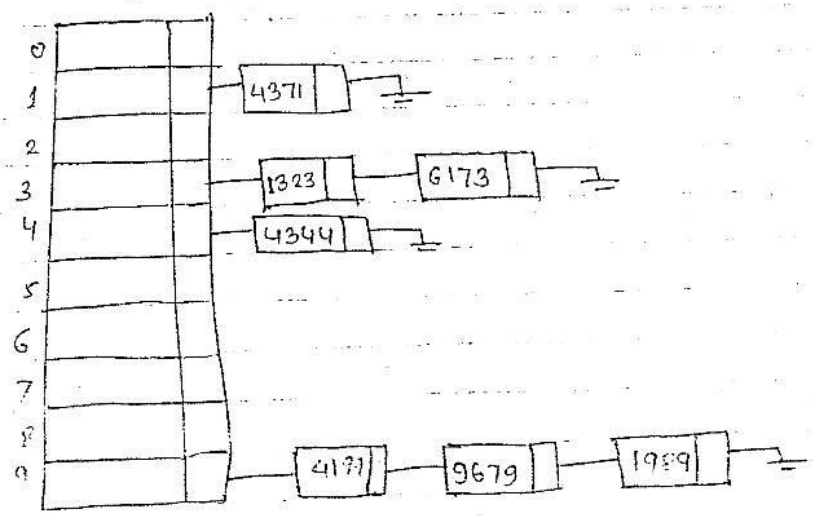
Soln:

Here,

Base hash function $h(x) = x \bmod 10$

for key 4371,
$h(x) = 4371 \bmod 10 = 1$

for key 1323, $h(x) = 3$
for key 6173, $h(x) = 3$
for key 4199, $h(x) = 9$
for key 4344, $h(x) = 4$
for key 9679, $h(x) = 9$
for key 1989, $h(x) = 9$



2. a. Open addressing hash table with linear probing:

0	9679
1	4371
2	1989
3	1323
4	6173
5	4344
6	
7	
8	
9	4199

b. Open addressing hash table with quadratic probing:

0	9679
1	4371
2	
3	1323
4	6173
5	4344
6	
7	
8	1989
9	4199

Here, Second hash-function:

for key 6173 = $h_2(6173) = 7 - 6173 \bmod 7 = 7 - 6 = 1$

for key 4344 = $7 - 4344 \bmod 7$
$= 7 - 4 = 3$

for key 9679 = $7 - 9679 \bmod 7$
$= 7 - 5 = 2$

for key 1989 = $7 - 1989 \bmod 7$
$= 7 - 1 = 6$

$h_2(x), 2h_2(x), 3h_2(x) \ldots$

0	
1	4371
2	1989
3	1323
4	6173
5	~~9679~~
6	~~4344~~
7	4344
8	
9	4199