

Unit 3

Q7

Hashing

Hashing is a common method of accessing data records using the hash table. Hashing can be used to build, search, or delete from a table.

Hash Table :-

A hash table is a data structure that stores records in an array, called a hash table.

Hash table can be used for quick insertion and searching.

Hash function :-

Hash function is a function which is applied on a key by which it produces an integer, which can be used as an address of hash table. Hence one can use the same hash function for accessing the data from the hash table. In this the integer returned by the hash function is called Hash Key.

Types of hash function

There are various types of hash function which are used to place the data in a

hash table,

1) Division method

In this, the hash function is dependent upon the remainder of a division. For example:- if the record 52, 68, 99, 84 is to be placed in the hash table and let us take the table size as 10.

Then:

$$\begin{aligned}
 h(\text{Key}) &= \text{record} \% \text{table size} \\
 &= 52 \% 10 = 2 \\
 &= 68 \% 10 = 8 \\
 &= 99 \% 10 = 9 \\
 &= 84 \% 10 = 4
 \end{aligned}$$

0	
1	
2	52
3	
4	84
5	
6	
7	
8	68
9	99

2) Mid square method

In this method firstly key is squared and then mid part of the result is taken as the index. For example: consider that if we want to place a record of 3101 and the size of table is 1000. So

$$3101 \times 3101 = 9616201$$

i.e

$h(3101) = 162$ (middle 3 digit)

3) Folding Method

There are 2 types of folding methods used, fold shift and fold boundary

Fold Shift

You divide the key in parts whose size matches the size of required address. The parts are simply added to get the required address.

Key: 123456789 and size of required address is 3 digits.

$$\begin{array}{r} 123 \\ + 456 \\ + 789 \\ \hline 1368 \end{array}$$

To reduce the size to 3 either 1 or 8 is removed and accordingly the key would be 368 or 136 respectively.

Fold Boundary
you again divide the key in parts whose size matches the size of required address. But now you also apply folding, exp- except for the middle part, if its there

Key: 123456789 and size of required address is 3 digits.

$$321 \text{ (folding applied)} + 456 + 987 \text{ (folding applied)} = 1764$$

(discard 1 or 4)

4) Digit-extraction Method

Using digit extraction selected digits are extracted from the key and used as the address.

Example:

- Using six-digit employee number to hash to a three digit address (000-999), we could select the first, third and fourth digits (from the left) and use them as the

address.

The keys are:

379452 \rightarrow 394

121267 \rightarrow 112

378845 \rightarrow 388

5) Rotation Method

Rotation method is generally not used by itself but rather is incorporated in combination with other hashing methods.

It is most useful when keys are assigned serially.

Example

600101 \rightarrow 160010

600102 \rightarrow 260010

600103 \rightarrow 360010

Now apply fold shift

16 + 00 + 10 \Rightarrow 26

26 + 00 + 10 \Rightarrow 36

36 + 00 + 10 \Rightarrow 46

6) Direct Method

In direct hashing the key is the address without any algorithmic manipulation

Example: Imagine that a small organization is assigned an employee number between 1 and 100. In this case, if we create an array of 101 employee records, the employee number can be directly used as the address of any individual record.

7) Pseudo Random Method

Key used as seed in a pseudo random number generator
Random number scatter scaled into possible address range.

$$y = ax + c$$

$$x = \text{Key}$$

$$a, c = \text{coeff}$$

Ex:

$$y = (17 * (121267) + 7) \text{ modulo } 307 + 1$$

Problems with hashing

- Collision: No matter what the hash function, there is the possibility that two different keys could resolve to the same hash address.

This situation is known as

a collision.

Handling the Collisions: The following techniques can be used to handle the collisions

- Chaining
- Double hashing (Re-hashing)
- Open Addressing (Linear probing, Quadratic probing)
- ~~link list addressing~~

1) Chaining

It is a method in which additional field with data i.e. chain is introduced. A chain is maintained at the home bucket. In this when a collision occur then a linked list is maintained for colliding data.

Example

Let us consider a hash table of size 10 and we apply a hash function of $H(\text{Key}) = \text{Key} \% \text{Size of table}$. Let us take the keys to be inserted are 31, 33, 77, 61. In the above diagram we can see at same bucket 1 there are

two records which are maintained by linked list or we can say by chaining method.

0	NULL	
1		→ 31 → 61 N
2	NULL	
3		→ 33 N
4	NULL	
5	NULL	
6	NULL	
7		→ 77 N
8	NULL	
9	NULL	

2) Linear probing

It is very easy and simple method to solve or to handle the collision. In this collision can be solved by placing the second record linearly down, whenever the empty place is found. In this method there is a problem of clustering which means at some place block of a data is formed in a hash table.

Example:

let us consider a hash table

of size 10 and hash function is defined as $H(\text{key}) = \text{key} \% \text{table size}$. Consider that following keys are to be inserted that are 56, 64, 36, 71

0	NULL
1	71
2	NULL
3	NULL
4	64
5	NULL
6	56
7	36
8	NULL
9	NULL

In this diagram we can see that 56 and 36 need to be placed at some bucket but by linear probing technique the records linearly placed downward if place is empty i.e. it can be seen 36 is placed at index 7.

3) Quadratic probing

This is a method in which solving of clustering problem is done. In this method

the hash function is defined by the $H(\text{key}) = (H(\text{key}) + x \times x) \% \text{table size}$.

let us consider we have to insert following elements that are: 67, 90, 55, 17, 49

0	90
1	
2	
3	
4	
5	55
6	
7	67
8	17
9	49

In this we can see if we insert 67, 90 and 55 it can be inserted easily but at case of 17 hash function is used in such a manner that:-

$(17 + 0 \times 0) \% 10 = 7$ (when $x=0$ it provide the index value 7 only) by making the increment in value of x . let $x=1$ so $(17 + 1 \times 1) \% 10 = 8$. In this case bucket 8 is empty hence we

will place 17 at index 8.

4) Double hashing

It is a technique in which two hash functions are used when there is an occurrence of collision. In this method 1st hash function is simple as same as division method. But for the second hash function there are two important rules which are

1) It must never evaluate to zero.

2) Must be sure about the buckets, that they are probed.

The hash functions for this technique are:-

$$H_1(\text{key}) = \text{key} \% \text{table size}$$

$$H_2(\text{key}) = p - (\text{key} \bmod p)$$

where p is the prime number which should be taken smaller than the size of a hash table.

Example :- let us consider we have to insert 67, 90, 55, 17, 49.

0	90	
1	17	
2		
3		
4		
5	55	
6		
7	67	
8		
9	49	

In this we can see 67, 90 and 55 can be inserted in a hash table by using first hash function but in case of 17 again the bucket is full and in this case we have to use the second hash function which is $H_2(\text{Key}) = P - (\text{Key} \bmod P)$ here P is a prime number which should be taken smaller than the hash table so value of P will be the 7.

i.e. $H_2(17) = 7 - (17 \div 7) = 7 - 3 = 4$ that means we have to take 4 jumps for placing the 17. Therefore, 17 will be placed at index 1.