Page No.
Date : / /

Hashing

- Hash Table: Hash table is a data structure used for sorting and se data quickly.

Hash table is made by using hash Function and hash key.

Every data in hash table is connected with same hash key.

- Hash Key: Hash key is the required piece of data that can be used to searching data in Hash table.

- Hash Function:

Hash Function is a Function which is used to put the data in the hash Function.

Hash Function is used to Find Hash key.

Page No.
Date : / /

* Types of Hashing Methods.

There are Five types of Hashing Methods.

1) Division Method
2) Mid Square Method
3) Multiplicative hash Function
4) Digit Folding
5) Digit Analysis

1 Division Method:

In Division method, every hash Function depends on the reminder of division.

This is the most simple and easiest method to Find hash table.

Ex. 25, 51, 52, 63 Placed in hash table and table size is 10.

For Division Method h(key) = record % table size.

From this data, table size = 10.

① 25 - h(key) = 25 %. 10 = 5

② 51 - h(key) = 51 %. 10 = 1

③ 52 - h(key) = 52 %. 10 = 2

④ 63 - h(key) = 63 %. 10 = 3

Here, we get every data's hash key.

0	
1	51
2	52
3	63
4	
5	25
6	
7	
8	
9	

Hash table.

- Advantages : This method is quite good and very fast.

- disadvantages : Sometimes extra care should be take to Find remindor.

2 Mid Square Method :

In this method, To Finding the hash key we have to follow this two step.

1) For key - Do every records square.
2) After that, according to hash table size selecte the middle part.

Ex. If table size is 10 then select 1 number to the middle.

Ex. 12, 15, 17, 10 Placed in hash table and table size is 10.

Here, table size = 10.

① H(12) = 12 x 12 = 144 = 4

② H(15) = 15 x 15 = 225 = 2

③ H(17) = 17 x 17 = 289 = 8

④ H(10) = 10 x 10 = 100 = 0

Here, We get every data's Hash key.

Page No.
Date: / /

0	10
1	7
2	15
3	
4	12
5	
6	
7	
8	17
9	

Hash Table

- Advantages: The performance of this method is good ~~bec~~ because all the records contribute to the result.

- Disadvantages: The size of the table is one of the biggest disadvantag of this method.

3 Multiplicative Hash Function:

In this method, Key value is multiplie with some contant value.

Using this method we can Find the which location, which record

is placed.

H(Key) = Floor(P * (Key Value * A))

Here, P, A = Constant value.

Ex. Key value is 25 and P = 50, A = 0.61802453 Find the record at 25th location.

H(25) = Floor(50 * (25 x 0.61802453))

= Floor(772.5306625)

= 772

772 record is placed at 25th location.

- Advantages: This method is work between any value of 0 to 1.
- Disadvantages: This method working is very slow.

4 Digit Folding:

In this method, Find the hash key we have to follow this step.

1) First Divided record into the equal size which is according to the table size.

Ex. If table size is 100 then, record must be divide into two digit.

2) After that, do addition of all the equal size of digit number.

This addition is give a location of record.

Ex. Find the 54678842 record location in hash table size 1000.

h(key) = 546 + 182

= 728.

This record is placed at location 728.

Date : / /

- Advantages: This method exeution is very simple.

- Disadvantages: The size of the table is the disadvantages of this method.

* Collision :

When we Find the hash Key and we get ~~the~~ the same hash key which is occupied by other record, this condition is called Collision in hash table.

Ex. 15, 16, 17, 25 Find the hash Key and Placed in hash table where table size is 10.

-> Here, Table size = 10.

By Division Method,

H(key) = record %. 10

① 15 -> H(K) = 15 %. 10 = 5

② 16 -> H(K) = 16 %. 10 = 6

③ 17 -> H(K) = 17 % 10 = 7

④ 25 -> H(K) = 25 % 10 = 5

0		Here, at location 5, we have to records.
1		
2		
3		
4		This condition is called Collision.
5	15 — 25	
6	16	
7	17	
8		
9		

* Collision Resolution Techniques:

To Remove the Collision, we have Two types of Collision Resolution Techniques.

1) Chaining

2) Open Addressing.

(a) Linear Probing
(b) Quadratic Probing
(c) Double hashing.

Date : / /

1 Chaining :

This method is use to remove collision in hash Function.

In this method, when collision occurs then, this records are made # chain as linked list.

Ex. 15, 16, 17, 27, 25 and table size is 10.

By Division method,

H(k) = record % table size

① 15 - H(K) = 15 % 10 = 5

② 16 - H(K) = 16 % 10 = 6

③ 17 - H(K) = 17 % 10 = 7

④ 27 - H(K) = 27 % 10 = 7

⑤ 25 - H(K) = 25 % 10 = 5

Here, Collision is occurs at 27, 17 and 15, 25.

- Hashing using Chaining:

0		
1		
2		
3		
4		
5	15	→ 25
6	16	
7	17	→ 27
8		
9		

2 Open Addressing:

In this method, there are three Collision Resolution Techniques.

(a) Linear Probing:

This method is use to remove collision in hash table.

In this method, When collision is occurs then sequentially a location is found when location is null then this element is placed at that location.

Page No.
Date : / /

Ex 22, 42, 44, 54, 77, 88 records are placed in hash table and hash table size is 10.

-> Here, Hash table size = 10

By Divison Method,

H(Key) = Record % Table size

- 22 - H(K) = 22 % 10 = 2
- 42 - H(K) = 42 % 10 = 2
- 44 - H(K) = 44 % 10 = 4
- 54 - H(K) = 54 % 10 = 4
- 77 - H(K) = 77 % 10 = 7
- 88 - H(K) = 88 % 10 = 8

Here, For 22 and 42, Collision is occurs.

22 is placed at location 2, For 42 we can use Linear Probing.

For 42, location 3rd is null after the location 2nd.

So, We placed 42 at location 3rd.

Similarly, 44 and 54 has collision
54 is placed at location 5th

0	
1	
2	22
3	42
4	44
5	54
6	
7	77
8	88
9	

(b) Quadratic Probing:

This method is use to remove collision in hash table.

Using Quadratic Probing

$$H_i(Key) = (H(K) + i^2) \% m$$

where m = table size
i = 0 to n

Ex. 22, 42, 44, 54, 77, 88 records are placed in hash table and hash table size is 10.

-> Here, Hash table size = 10

By Division Method,

HCKey) = Records %. Table size

- 22 - HCK) = 22 %. 10 = 2
- 42 - HCK) = 42 %. 10 = 2
- 44 - HCK) = 44 %. 10 = 4
- 54 - HCK) = 54 %. 10 = 4
- 77 - HCK) = 77 %. 10 = 7
- 88 - HCK) = 88 %. 10 = 8

Here, at 2nd location, we have two records 22 and 42.

22 is Placed at 2nd location.
For 42 Using Quadratic Probing

$H_0(42) = (42 + 0^2) \% 10$
$= 4$

Here,

→ $H_0(42) = (42 + 0^2) \% 10$
$= 2$

location 2nd is already use.

$H_1(42) = (42 + 1^2) \% 10$
$= 3$

location 3rd is null. So, 42 is Placed at 3rd location.

→ Similarly, For 54

$H_0(54) = (54 + 0^2) \% 10$
$= 4$

location 4th is already use.

$H_1(54) = (54 + 1^2) \% 10$
$= 5$

location 5th is null. So, 54 is Placed at 5th location.

0	
1	
2	22
3	42
4	44
5	54
6	
7	77
8	88
9	

(c) Double Hashing:

This method is use to remove collision in hash table.

In Double Hashing, there are two keys are use.

First Key, For without collision H(Key) = Record %. size

Second Key, For Collision

$H_1(Key) = M - \cancel{H(Key)}$ (Record %. 7)

where M = Prime number which is smaller than ~~to~~ table size.

Page No.
Date : / /

Ex. 37, 90, 45, 22, 17, 49, 55 Placed in Hash table and table size is 10.

-> Here table size = 10

-> First Key

37 -> 37 % 10 = 7
90 -> 90 % 10 = 0
45 -> 45 % 10 = 25
17 -> 17 % 10 = 7
22 -> 22 % 10 = 2
49 -> 49 % 10 = 9
55 -> 55 % 10 = 5

Here, Collision occurs for record 37, 17 and 45, 55.

-> For 17 second key.

M = 7 (Smaller Prime number)

H1(key) = 7 - (17 % 7)
= 7 - 3
= 4

Here, H1(key) = 4.

So, We have to take 4 null Jump.

→ For 55 second key,

$M = 7$ (Smaller Prime number)

$H_1(Key) = 7 - (55 \% 7)$
$= 7 - 6$
$= 1.$

Here $H_1(Key) = 1$

So, We have to take 1 null Jump in table.

0	90
1	17
2	22
3	
4	
5	45
6	55
7	37
8	
9	49

4 Jump

1 Jump