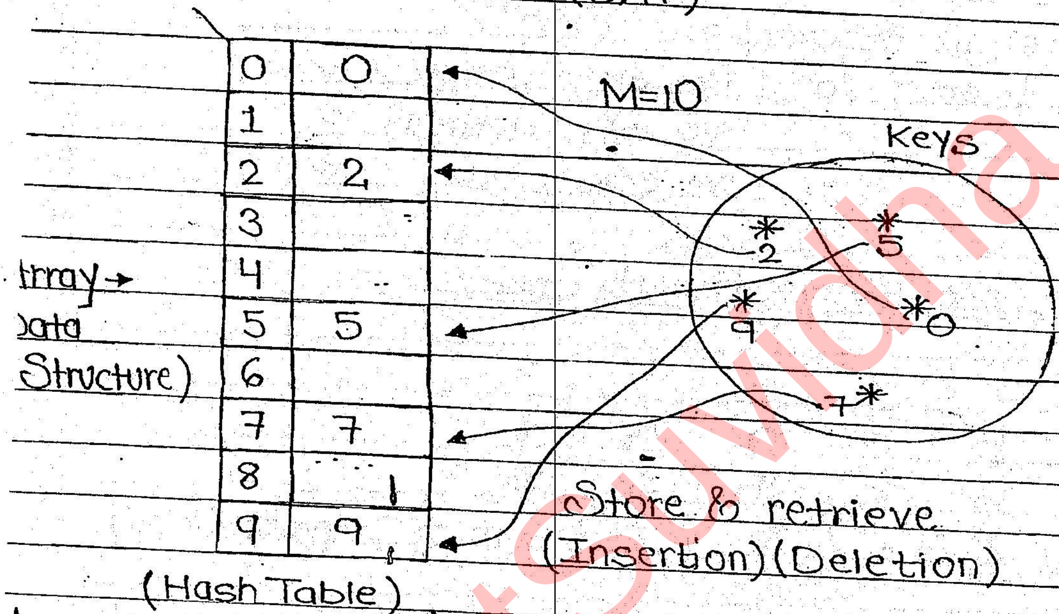
Chapter 5 : Hashing

It is one of the searching technique where worst case is $O(1)$.

: Direct Address Table (DAT) -



Accessing / Retrieving the x from array - HT[x]

∴ In DAT, key itself is the address without any calculation.

∴ Worst case searching time is $O(1)$

∴ Drawback → Huge memory required. Even though no. of keys are very less, if one of key is very large (2^{1000}), then there is need of hash table of size approx. 2^{1000}.

To eliminate above draw-backs, we are going to Hash Functions-

Page No. 7
Date : 19/03/16

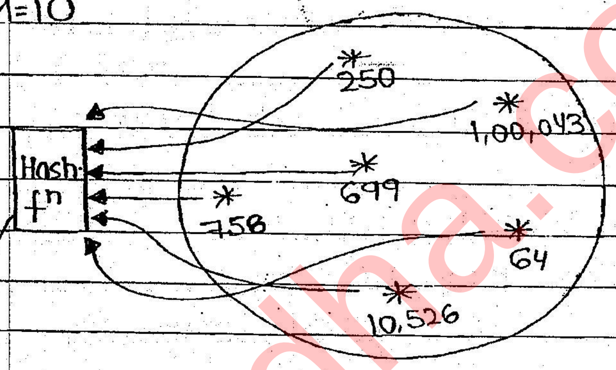
Hash Functions -

HT

0	250
1	
2	
3	1,00,043
4	64
5	
6	10,526
7	
8	758
9	699

(Hash Table)

M=10



Time - O(1)
f^n - modulo 10

Drawback - collision (if two keys contain same hash address)

Hashing functions will be best if no. of collision are as minimum as possible.

Types of Hash Functions -

I. Division Modulo method -

(i) M = 1000 [0 - 999)

Key = 123456789

hf(Key) = Key modulo m

= 123456789 mod 1000

= 789

0	
789	123456789

Page No. 08
Date : 19/03/16

ii) $M = 8\ (2^3)$

Key = 101001010101100101

$hf(Key)$ = Key mod M

101001010101100101 mod 2^3 = 101

↳ LSB

Irrespective of key, it cares only for LSB's.

iii) $M = 2^K$

Key = 101011001011

$hf(Key)$ = Key mod M

$101011\underbrace{001011}_{K\ LSB}$ mod 2^K = 001011

- Don't pick m value exactly powers of 2 because if $m = 2^K$, then hash function of key = LSB-K bits always. (Chances of collision)
2. Pick the m value which is prime no. but it can not be close to powers of 2.
3. Which one will you take for m in this method?
   a) 61 c) 523
   b) 729 d) 1024

II Digit-Extraction Method-

(i) $M = 1000\ (0-999)$

Key = 789123456

$hf(Key)$ = 715

0	
!	
715	789123456
!	
999	

Upto the choice which digit to extract. 3 digit extracted means 0-999 addresses.
Extraction similar way to storing.
This method is easy to form counters (No condition)
Worst method.

III Mid-square method -

M = 1000

Key = 84925

$hf(Key) \Rightarrow (84925)^2 \Rightarrow 7212255625 = 225$ or 25.

Making counter difficult if squaring first and finding mid. Taking three as majority numbers are three digited.

Best method among above three.

IV Folding Method -

M = 1000 (0-999)

Key = 123456789

Fold boundry

123456789

123
789
912

If it is 1912

191
+ 2
193

Fold shifting

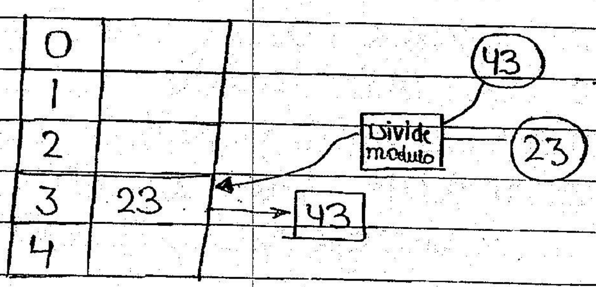
123456789

123
456
+ 789
1368

136
+ 8
144

(Better (all digits are involved))

Collision Resolution Technique -



1. Chaining (Outside)
2. Open Addressing (Inside) → Linear probing, Quadratic probing, Double Hashed

1. Chaining -

M = 10 (0-9)

Keys = 55, 98, 63, 75, 99, 73, 60, 95, 44, 29, 39, 73, 25, 108

hf(Key) = Key modulo m

collision res. tech. = chaining

0	7000	→ [60 | Null] (7000)
1	Null	
2	Null	
3	3000	→ [6000 | 63] (3000) → [Null | 73] (6000)
4	44	
5	1000	→ [4000 | 55] (1000) → [8000 | 75] (4000) → [5500 | 95] (8000) → [Null | 25] (5500)
6	NULL	
7	NULL	
8	2000	→ [4500 | 98] (2000) → [Null | 108] (4500)
9	5000	→ [1500 | 99] (5000) → [2500 | 29] (1500) → [Null | 39] (2500)

1. In chaining, space wasted in form of linked list even though space available inside.
2. The length of longest chain possible is n. So worst case searching time is $O(n)$.
3. Best case $O(1)$ Average case $\frac{n}{2} = O(n)$

3. Greatest advantage with the chaining is infinite no. of keys can be stored because of linked list.
4. Insertion time O(1) [Every case]
5. Deletion time : Best case O(1)
Worst case O(n) Average case O(1)
(If the node address is given, directly go to tha and delete the node) O(1) [doubly linked list]
If singly linked list, then worst case O(n)

Q- What is the length of longest chain?
Maximum, minimum & average length

II Open Addressing -
(i) Linear Probing - (Closed Hashing)
M = 10 (0-9)
Keys = 55, 99, 60, 75, 89, 42, 58, 69, 25
hf(Key) = Key mod m.
collision res. tech. = Linear probing
LP(Key, i) = (hf(Key)+i) mod m
$\forall i \in \{0,1,2 ... m-1\}$

0	60
1	89
2	42
3	69
4	
5	55
6	75
7	25
8	58
9	99

LP(55,0) = 5 0-C
LP(99,0) = 9 0-C
LP(60,0) = 0 0-C
LP(75,0) = 5 1-C
LP(75,1) = 5+1 = 6
LP(89,0) = 9 2-C
LP(89,1) = 0
LP(89,2) = 1

Total = 9-

Page No. 12
Date : 19/03/16

i) Quadratic Probing →

M = 1000

Keys = 25, 39, 46, 55, 89, 23, 68, 70, 94, 56

hf(Key) = key mod m

Collision res. tech. = Quadratic probing

$QP(\text{Key}, i) = (hf(\text{Key}) + c_1 i + c_2 \cdot i^2)$

$\forall i \in \{0, 1, 2 \ldots m-1\}$ $c_1 = c_2 = 1$

for 25 $QP(25,0) = 5 + 1\cdot0 + 1\cdot0^2 = 5$

$QP(39,0) = 9 + 1\cdot0 + 1\cdot0^2 = 9$

$QP(46,0) = 6 + 1\cdot0 + 1\cdot0^2 = 6$

$QP(55,0) = 5 + 1\cdot0 + 1\cdot0^2 = 5$ - 1C

$QP(55,1) = 5 + 1\cdot1 + 1\cdot1^2 = 7$

$QP(89,0) = 9 + 1\cdot0 + 1\cdot0^2 = 9$ - 1C

$QP(89,1) = 9 + 1\cdot1 + 1\cdot1^2 = 1$

$QP(23,0) = 3 + 1\cdot0 + 1\cdot0^2 = 3$ - 0C

$QP(68,0) = 8 + 1\cdot0 + 1\cdot0^2 = 8$ - 0C

$QP(70,0) = 0 + 1\cdot0 + 1\cdot0^2 = 0$ - 0C

$QP(94,0) = 4 + 1\cdot0 + 1\cdot0^2 = 4$ - 0C

0	70
1	89
2	
3	23
4	94
5	25
6	46
7	55
8	68
9	39

$QP(56,0) = 6 + 1\cdot0 + 1\cdot0^2 = 6$

$QP(56,1) = 6 + 1\cdot1 + 1\cdot1^2 = 8$ 2-C

$QP(56,2) = 6 + 1\cdot2 + 1\cdot2^2 = 12 \bmod 10 = 2$

$QP(78,0) = 8 + 1\cdot0 + 1\cdot0^2 = 8$

$QP(78,1)$ $8 + 1\cdot1 + 1\cdot1^2 = 0$

$QP(78,2)$ $8 + 1\cdot2 + 1\cdot2^2 = 4$

$QP(78,3)$ $8 + 1\cdot3 + 1\cdot3^2 = 0$

$QP(78,4)$ $8 + 1\cdot4 + 1\cdot4^2 = 8$ 10-C

$QP(78,5)$ $8 + 1\cdot5 + 1\cdot5^2 = 8$

$QP(78,6)$ $8 + 1\cdot6 + 1\cdot6^2 = 0$

$QP(78,7)$ $8 + 1\cdot7 + 1\cdot7^2 = 4$

$QP(78,8)$ $8 + 1\cdot8 + 1\cdot8^2 = 0$ Total 12c

$QP(78,9)$ $8 + 1\cdot9 + 1\cdot9^2 = 8$

Page No. 13
Date : 19/03/16

(III) Double Hashed →

$DH(Key, i) = (hf_1(Key) + i \times hf_2(Key))$
$hf_1(Key) = Key \bmod m$
$hf_2(Key) = 1 + (Key \bmod (m-2))$
Keys = 25, 39, 46, 55, 89, 23, 68, 70, 94, 78

$DH(25,0) = 5 + 0 \cdot hf_2(25) = 5$ – 0C
$DH(39,0) = 9 + 0 \cdot hf_2(39) = 9$ – 0C
$DH(46,0) = 6 + 0 \cdot hf_2(46) = 6$ – 0C
$DH(55,0) = 5 + 0 \cdot hf_2(55) = 5$ – 1C
$DH(55,1) = 5 + 1 \cdot 8 = 13 \Rightarrow 3$
$DH(89,0) = 9 + 0 \cdot hf_2(89) = 9$ – 1C
$DH(89,1) = 9 + 1 \cdot 2 = 11 \Rightarrow 1$
$DH(23,0) = 3 + 0 \cdot hf_2(23) = 3$
$DH(23,1) = 3 + 1 \cdot 8 = 11 \Rightarrow 1$
$DH(23,2) = 3 + 2 \cdot 8 = 19 \Rightarrow 9$
$DH(23,3) = 3 + 3 \cdot 8 = 7$ – 3C
$DH(68,0) = 8 + 0 \cdot hf_2(68) = 8$ – 0C
$DH(70,0) = 0 + 0 \cdot hf_2(70) = 0$ – 0C
$DH(94,0) = 4 + 0 \cdot hf_2(94) = 4$ – 0C
$DH(78,0) = 8 + 0 \cdot hf_2(78) = 8$
$DH(78,1) = 8 + 1 \cdot hf_2(78) = 8 + 7 = 15 \Rightarrow 5$ – 2C
$DH(78,2) = 8 + 2 \cdot 7 = 2$

0	70
1	89
2	78
3	55
4	94
5	25
6	46
7	23
8	68
9	39

Example to try – M = 10
Keys = 25, 15, 90, 75, 69, 59, 44, 58, 85
hf(Key) = K mod m
Try for linear, quadratic probing, double hashing

Page No. 14
Date : 19/03/16

LINEAR PROBING -
1. We are not wasting space outside in form of linked list.
2. Searching time O(1) - Best case
O(m) - Worst case
3. Insertion time same as searching time.
4. Deletion time O(1) - Best case
O(m) - Worst case Same.
5. Deletions in linear probing will trouble the other elements, it can be managed with special symbol \$. If more no. of \$ symbols, then go for re-hashing.

QUADRATIC PROBING -
1. Not wasting space
2. Searching, Insertion, Deletion
O(1) Best case O(m) - Worst case.
3. Deletion will trouble, managed by \$. If more deletions, then go for re-hashing.

DOUBLE HASHING -
1. -
2. -
3. -

Problem - M=10
Keys = 53, 60, 82, 91, 80, 90
hf(Key) = K mod m
Collision Res. Tech = Linear Probing

0	1	2	3	4	5	6	7	8	9
60	91	82	53	80	90				

Linear Probing.

Primary Clustering - If two keys contain same starting hash address (index) they both will follow same path unnecessarily in linear manner, then average searching time is increased, the problem is primary clustering.
Linear probing suffers from primary clustering.

Average case $= (1+2+3+ \dots m)/m$
$= \frac{m+1}{2} = O(m)$

0	1	2	3	4	5	6	7	8	9
60	91	82	53			80			

Quadratic Probing

Secondary Clustering - If a two keys contain same starting hash address they both will follow same path unnecessarily in quadratic manner, then average searching time is increased (less than linear probing searching) the problem is secondary clustering.
Quadratic probing suffers from this problem.
Average case $= O(m)$ [asymptotically same] mathematically less.

0	1	2	3	4	5	6	7	8	9
60	91	82	53	80		90			

Double Hashed

Average case $= (3+5+7+ \dots 2_1)/m$
slots
$m/m = 1 \Rightarrow O(1)$

In double hashing, even though two Keys contain same starting hash addresses, they both will follow different path, because the second hash f^n will give differunt values. Because of this reason, 99% redundancy is eliminated. Secondary clustering is there little bit.

Average case = $O(1)$

Using perfect hashing, worst case searching time is $O(1)$ (Hashing within hashing)

0	h_0	c, d, e
1	h_1	
2	h_2	
3	h_3	a, b
4	h_4	
5	h_5	

$h(c) = 0$
$h(d) = 0$
$h(e) = 0$

$h(a) = 3 \quad h_3(a) = \cancel{2}1$
$h(b) = 3 \quad h_3(b) = 3$

m - slots
n - Keys

In m-slots $\rightarrow$ n-Keys
1-slot $\rightarrow$ n/m Keys / slot
Load factor (α).

The expected no. of probes (attempts) in an unsuccessful search of an open addressiong technique = $1/1-\alpha$

The expected no. of proble in successful search of an ~~open~~ $= \frac{1}{\alpha} \log_e \frac{1}{1-\alpha}$