17. Collision Handing Techniques:

- In Hash Table, collision Occurs when two keys are hashed to the same index in a hash table.
- It means Calculated hash value for the two keys is the same.

Collision Handing Techniques

- Seperate Chaining (Open Hashing)
- Open Addressing (Closed Hashing)
  - Linear Probing
  - Quadratic Probing
  - Double Hashing

Seperate Chaining:

To handle the collision

- This technique creates a linked list to the slot for which collision occurs.
- The new key is then inserted in the linked list.
- These linked list to the slot appear like chain that is why, this technique is called seperate chaining.

Time complexity:

Searching and Deletion: worst case $O(n)$

OPEN ADDRESSING

In open addressing

- Unlike seperate Chaining, all the keys are stored inside the hash table.
- No keys is stored outside the hash table.

Technique used for open Addressing

1). Linear Probing:

In Linear Probing, we linearly probe for next slot.

- We keep probing until an empty slot is found.

To Resolve:

$h(k,i) = (h(k)+i) \bmod m$

i = count no of probe k = Key

m = size of hash table

Advantage: easy to compute

Disadvantage: clustering, takes time to search an element or empty slot.

Time complexity: $O(m)$ m = table size

2) Quadratic Probing:

In quadratic Probing we look for i^2th slot in the ith iteration if the given hash value collides with the hash Table.

$h(k,i) = [h(k) + i^2] \bmod m$.

Advantage: more efficient for closed hash table

Disadvantage: Secondary clustering.

two keys have the same probe sequence when they hash to the same location.

Time complexity: $O(N * L)$, N = length of array

Space complexity: $O(1)$. L = size of hash table.

PAGE No.
DATE / /

3) Double Hashing
idea of applying second hash function to key when collision occurs.

~~h(k,i) = [hash 1~~ (key)

~~h(k,i) = hash 1(~~

$h(k,i) = [h1[k] + i * h2[k]] \bmod m.$

we increase i when collision occurs.

Advantages: no clusters.
best for probing coz it can find next free slot in Hash table more quickly than Linear probing.

Disadvantages:
Time consuming to compute two hash function
Poor cache performance.

Time complexity: O(m) worst case. m = size of hash table

18 Linear probing with example.

- In Linear Probe, we linearly probe for next slot.
- The simplest approach to resolve a collision is linear probing.
- In this technique, if a value is already stored at a location generated by h(k), it means collision occured then we do a sequential search to find the next empty location.
- Here the idea is to place a value in the next available position.
- Because in this approach searches are performed sequentially, so its knowns as linear probing.

- Here array or hash table is considered circular because when the last slot reached an empty location not found then search preceeds to the first location of the array.

To Resolve the collision: $h(k,i) = [h(K)+i] \bmod m$

m = size of hash table

$h(K) = (K \bmod m)$

i = probe no varies from 0 to $m-1$.

Therefor for a key k, the 1st location generated by $[h(K)+0] \bmod m$, $(i=0)$.

If the location is free, the value is stored at this location. If value successfully stored then probe count becomes 1.

If location is not Free then 2nd probe generates address of location given by $[h(k)+1] \bmod m$.

Time complexity: $O(m)$ m = table size.

Example:

Insert the following sequence of keys in the hash table
{ 9, 7, 11, 13, 12, 8 }.

$h(k,i) = [\, h(k) + i \,] \bmod m$

$h(k) = 2k+5$

$m = 10$.

Solution:

Step 1:

Empty Hash Table of size 10.

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Step 2:

Insert keys.

$h(k) = 2k+5$.

$h(9) = 2 \times 9 + 5$

$= 18 + 5$

$h(9) = 23$

$\therefore h(k,i) = [h(k)+i] \bmod m$

$h(9,0) = [h(9)+0] \bmod 10$

$= [23] \bmod 10$

$= 3$.

0	
1	
2	
3	9
4	
5	
6	
7	
8	
9	

Step3:

Insert Key

$h(k) = 2k+5$

$h(7) = 2 \times 7 + 5$

$= 19$

$h[k,i] = [h(k)+i] \bmod m$

$h(7,0) = [h(7)+0] \bmod 10$

$= [19+0] \bmod 10$

$= 19 \bmod 10$

$= 9$

0	
1	
2	
3	9
4	4
5	
6	
7	11
8	
9	7

Step4:

Insert key

$h(k) = 2k+5$

$h(11) = 2 \times 11 + 5 = 22+5 = 27$

$h(k,i) = (h(k)+i] \bmod 10$

$= (h(11)+0] \bmod 10$

$= 27 \bmod 10 = 7$

Step5:

Insert key

$h(k) = 2k+5$.

$h(14) = 2 \times 14 + 5 = 33$.

$h(k,i) = (h(k)+i] \bmod 10$

$= (33+0] \bmod 10$

$= 3$.

Collision occurs location 3 is already full ∴

h(k).

$h(14,1) = (h(14)+1] \bmod 10$

$= 33+1 \bmod 10$

$= 34 \bmod 10$

$= 4$.

This is how linear probing collision resolution technique works.

16) Linear Probing:

63, 82, 94, 77, 53, 87, 23, 55, 10, 44

$h(k,i) = [h(k)+i] \bmod 10$.

Step1:

Draw Empty Hash Table.

Given size of Table = 10 = m

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Step2:

Insert 63.

~~h'(k) =~~ k = 63 i = 0 m = 10

$h'(k,i) = (h(k)+i) \bmod m$

$h(63,0) = (h(63)+0) \bmod 10$

$= 63 \bmod 10$

$= 3.$

0	
1	
2	
3	63
⋮	
9	

Step3:

Insert 82. k = 82 i = 0 m = 10

$h(k,i) = [h(k)+i] \bmod m$

$h(82,0) = [h(82)+0] \bmod 10$

$= 82 \bmod 10 = 2.$

0	
1	
2	82
⋮	
9	

Step4:

Insert 94. k = 94 i = 0 m = 10

$h(94,0) = [h(94)+0] \bmod 10$

$= 94 \bmod 10$

$= 4.$

0	
1	
2	82
3	63
4	94

Step5:

Insert 77

$K = 77 \quad i = 0 \quad m = 10$

$h(77,0) = [h(77)+0] \bmod 10$

$= 77 \bmod 10$

$= 7.$

0	
1	
2	82
3	63
4	94
5	
6	
7	77
8	
9	

Step6:

Insert 53 $\quad i = 0$

$h(53,0) = (h(53)+0] \bmod m$

$= 53 \bmod 10$

$= 3.$ ✓

$i = 1.$

$h(53,1) = (h(53)+1) \bmod 10$

$= (53+1) \bmod 10$

$= 54 \bmod 10$

$= 4.$ ✓

$i = 2.$

$h(53,2) = (h(53)+2) \bmod 10$

$= 55 \bmod 10$

$= 5$

0	
1	
2	82
3	63
4	94
5	53
6	
7	77
8	
9	

Step7:

Insert 87

$h(87,0) = [h(87)+0] \bmod 10$

$= 87 \bmod 10$

$= 7.$

$i = 1.$

$h(87,1) = [h(87)+1] \bmod 10$

$= 88 \bmod 10$

$= 8.$

0	
1	
2	82
3	63
4	94
5	53
6	
7	77
8	87
9	

Step 8:

Insert 23

$h(23,0) = [h(23)+i] \bmod m$

$= 23 \bmod 10$

$= 3$ ✓

0	
1	
2	82
3	63
4	94
5	53
6	23
7	77
8	87
9	

$i = 1$

$h(23,1) = [h(23)+1] \bmod 10$

$= 24 \bmod 10$

$= 4.$ ✓

$i = 2$

$h(23,2) = (h(23)+2) \bmod 10$

$= 25 \bmod 10$

$= 5.$ ✓

$i = 3$

$h(23,3) = (h(23)+3) \bmod 10$

$= 26 \bmod 10$

$= 6.$

Step 9:

Insert 55.

0	
1	
2	82
3	63
4	94
5	53
6	23
7	77
8	87
9	55.

$h(55,0) = (h(55)+0] \bmod 10$

$= 55 \bmod 10$

$= 5$ ✓

$i=1$ $h(55,1) = [h(55)+1] \bmod 10$

$= 56 \bmod 10$

$= 6$ ✓

$i=2$ $h(55,2) = [h(55)+2] \bmod 10$

$= 57 \bmod 10$

$= 7$ ✓

$i=3$ $h(55,3) = [h(55)+3] \bmod 10$

$= 58 \bmod 10.$ 8 ✓

$i=4$ $h(55,4) = (h(55)+4) \bmod 10$

$= 59 \bmod 10$

$= 9$

Step 10:

Insert 10.

$h(10, 0) = (h(10) + 0) \bmod 10$
$= 10 \bmod 10$
$= 0.$

0	10
1	
2	82
3	63
.	
:	
9	55

Step 11:

Insert 44.

$h(44, 0) = (h(44) + 0) \bmod 10$
$= 44 \bmod 10$
$= 4.$

$i = 1$ $h(44, 1) = (44 + 1) \bmod 10$
$= 45 \bmod 10$
$= 5$ ✓

$i = 2$ $h(44, 2) = 44 + 2 \bmod 10$
$= 46 \bmod 10$
$= 6$ ✓

$i = 3 \rightarrow 7$
$i = 4 \rightarrow 8$
$i = 5 \rightarrow 9$

$i = 6$ $h(44, 6) = (44 + 6) \bmod 10$
$= 50 \bmod 10$
$= 0$

$i = 7$ $h(44, 7) = (44 + 7) \bmod 10$
$= 51 \bmod 10$
$= 1.$

0	10
1	44
2	82
3	63
4	94
5	53
6	23
7	77
8	87
9	55.

PAGE NO

DATE / /

What is Hashing:

- Hashing is a technique of mapping a large chunk of data into small table using a hashing function.
- It is process of mapping keys and values into the hash table by using a hash function.
- It is done for faster access to elements.

Quadratic Probing:

63, 82, 94, 77, 53, 87, 23, 55, 10, 44

$h(k,i) = [h(k) + i^2] \bmod m$

Step1: Draw empty table of size 10.

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Step2: Insert 63.

$h(k,i) = (h(k) + i^2) \bmod m$

$= (63 + 0) \bmod 10$

$= 63 \bmod 10$

$= 3.$

0	
1	
2	
3	63
4	
5	
⋮	
9	

Step3: Insert 82.

$h(82,0) = (h(82) + 0^2) \bmod 10$

$= 82 \bmod 10$

$= 2.$

0	
1	
2	82
3	63

Step4: Insert 94

$h(94,0) = (h(94) + 0^2) \bmod 10$

$= 94 \bmod 10$

$= 4$

0	
1	
2	82
3	63
4	94

Step5: Insert 77.

$h(77,0) = (77+0^2) \bmod 10$

$= 77 \bmod 10$

$= 7.$

0	
1	
2	82
3	63
4	94
5	
6	
7	77
8	
9	

Step6: Insert 53.

$h(53,0) = (53+0^2) \bmod 10$

$= 53 \bmod 10$

$= 3$ ✓

i=1 $h(53,0) = (53+1^2) \bmod 10$

$= 54 \bmod 10$

$= 4$ ✓

i=2 $h(53,2) = (53+2^2) \bmod 10$

$= 53+4 \bmod 10$

$= 57 \bmod 10$

$= 7$ ✓

i=3 $h(53,3) = h(53)+3^2) \bmod 10$

$= (53+9) \bmod 10$

$= 62 \bmod 10$

$= 2$ ✓

i=4 $h(53,4) = (h(53)+4^2) \bmod 10$

$= (53+16) \bmod 10$

$= 69 \bmod 10$

$= 9.$

0	
1	
2	82
3	63
4	94
5	
6	
7	77
8	
9	53

0	
1	
2	82
3	63
4	94
5	
6	
7	77
8	87
9	53

Step7: Insert
87.

$h(87,0) = (87+0^2) \bmod 10$
$= 87 \bmod 10$
$= 7$ ✓

$i=1$ $h(87,1) = (h(87)+1) \bmod 10$
$= 88 \bmod 10$
$= 8.$

Step8: Insert 23.

$h(23,0) = (23+0) \bmod 10$
$= 3$ ✓

$h(23,1) = (23+1) \bmod 10$
$= 24 \bmod 10$
$= 4$ ✓

$h(23,2) = (23+4) \bmod 10$
$= 27 \bmod 10$
$= 7$

$h(23,3) = (23+9) \bmod 10$
$= 32 \bmod 10$
$= 2$ ✓

$h(23,4) = (23+16) \bmod 10$
$= 39 \bmod 10$
$= 9.$ ✓

$h(23,5) = (23+25) \bmod 10$
$= 48 \bmod 10 = 8$ ✓

$h(23,6) = (23+36) \bmod 10$
$= 59 \bmod 10 = 9$ ✓

$h(23,7) = (23+49) \bmod 10$
$= 72 \bmod 10 = 2$ ✓

NO slot available.

Step 9: Insert 55

$h(55,0) = 55 \bmod 10$

$= 5$ ✓

0	
1	
2	82
3	63
4	94
5	55
6	
⋮	

Step 10: Insert 10.

$h(10,0) = (10+0) \bmod 10$

$= 0.$

0	10
1	
2	82
3	63

Step 11: Insert 44

$h(44,0) = 44 \bmod 10$

$= 4.$ ✓

$h(44,1) = (44+1) \bmod 10$

$= 45 \bmod 10.$

$= 5$ ✓

$h(44,2) = (44+4) \bmod 10$

$= 48 \bmod 10$

$= 8$ ✓

$h(44,3) = (44+9) \bmod 10$

$= 53 \bmod 10 = 3.$

No spot Available.

0	10
1	
2	82
3	63
4	94
5	55
6	.
7	77
8	87
9	53.

15. Given : Quadratic probing
Table size $m = 10$.
Keys: 27, 72, 63, 42, 36, 18, 29, 101
$c1 = 1$ $c2 = 3$.

$h(k,i) = (h(k) + c_1 \cdot i + c_2 i) \bmod m$

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Steps:
Draw an empty table of size 10.

Step2:
Insert 27.
$\therefore h(27,0) = (h(27) + 1 \cdot 0 + 3 \cdot 0) \bmod 10$.
$= 27 \bmod 10$
$= 7$.

	0	
	1	
S2 →	2	72
S3 →	3	63.
	4	
	5	
	6	
S1 →	7	27.
	8	
	9	

Step3 Insert 72
$h(72,0) = (72 + 1 \cdot 0 + 3 \cdot 0) \bmod 10$
$= 72 \bmod 10$
$= 2$.

Step4 Insert 63 $= (63 + 1 \cdot 0 + 3 \cdot 0) \bmod 10$
$= 63 \bmod 10$
$= 3$

Step5 Insert 42 $= (42 + 1 \cdot 0 + 3 \cdot 0) \bmod 10$
$= 42 \bmod 10 = 2$.

$i=1$. Insert 42.

$h(42,1) = (42+1\cdot1+3\cdot1) \bmod 10$
$= (42+1+3) \bmod 10$
$= 46 \bmod 10$
$= 6.$

0	
1	36.
2	72
3	63
4	
5	
6	42
7	27
8	18
9	29.

Step 6: Insert 36

$h(36,0) = (h(36)+1\cdot0+3\cdot0) \bmod 10$
$= 36 \bmod 10$
$= 6$ ✓

$i=1$ $h(36,1) = (h(36)+1\cdot1+3\cdot1) \bmod 10$
$= (36+1+4) \bmod 10$
$= (36+5) \bmod 10$
$= 41 \bmod 10$:
$= 1.$

Step 7: Insert 18.

$h(18,0) = (18+1\cdot0+3\cdot0) \bmod 10$
$= 18 \bmod 10$
$= 8.$

0	
1	36
2	72
3	63
4	
5	101.
6	42
7	27
8	18
9	19.

Step 8: Insert 29.

$h(29,0) = (29+1\cdot0+3\cdot0) \bmod 10$
$= 29 \bmod 10$
$= 9.$ ✓

Step 9: Insert 101

$h(101,0) = (h(101)+1\cdot0+3\cdot0) \bmod 10$
$= 101 \bmod 10$
$= 1.$ ✓

$i=1$ $h(101,1) = (h(101)+1\cdot1+3\cdot1) \bmod 10$
$= (101+1+3) \bmod 10$
$= 105 \bmod 10$
$= 5.$

14. Linear Probing:

45, 8, 33, 85, 61, 10, 48, 76, 89.

$h(k,i) = (h(k)+i) \bmod m.$

Step1: Draw an empty table of size 10.

0	
1	
2	
3	
⋮	
10	

Step2:

Insert 45:

$h(45,0) = (h(45)+0) \bmod 10$

$= 45 \bmod 10$

$= 5.$

0	
1	
2	
3	
4	
5	45
9	

Step3: Insert 8.

$h(8,0) = (h(8)+0) \bmod m$

$= 8 \bmod 10 = 8.$

Step3: Insert 33

$h(33,0) = (h(33)+0) \bmod 10$

$= 33 \bmod 10 = 3.$

Step4: Insert 85

$h(85,0) = (h(85)+0) \bmod 10$

$= 85 \bmod 10 = 5$

$h(85,1) = (h(85)+1) \bmod 10$

$= 86 \bmod 10$

$= 6.$

Step5: Insert 61.

$h(61,0) = (h(61)+0) \bmod 10$

$= 61 \bmod 10$

$= 1.$

0	
1	61
2	
3	33
4	
5	45
6	85
7	
8	8
9	
10	
11	

PAGE No.
DATE / /

Step6: Insert 10.

$h(10,0) = (h(10)+0) \bmod 10$

$= 10 \bmod 10 = 0.$

0	10
1	61
2	
3	33
4	
5	45
6	85
7	76.
8	8
9	48
10	
~~11~~	

Step7: Insert 48.

$h(48,0) = (h(48)+0) \bmod 10$

$= 48 \bmod 10$

$= 8.$

$h(48,1) = (h(48)+1) \bmod 10$

$= 49 \bmod 10$

$= 9.$

Step8: Insert 76.

$h(76,0) = (h(76)+0) \bmod 10$

$= 76 \bmod 10$

$= 6.$

$h(76,1) = (h(76)+1) \bmod 10$

$= 77 \bmod 10$

$= 7.$

Step9: Insert 89.

$h(89,0) = (h(89)+0) \bmod 10$

$= 89 \bmod 10$

$= 9.$ ✓

$h(89,1) = (h(89)+1) \bmod 10$

$= 90 \bmod 10$

$= 0$ ✓

$h(89,2) = (89+2) \bmod 10$

$= 91 \bmod 10 = 1$ ✓

$h(89,3) = (89+3) \bmod 10$

$= 92 \bmod 10 = 2.$

0	10
1	61
2	89
3	33
4	
5	45
c	85
7	76
8	8
9	48
10	
~~11~~	

Division modulo: h(k) = k % m

45, 8, 33, 85, 61, 10, 48, 76, 89

Step1: Draw empty table of size 11

0	10
1	61
2	
3	33
4	
5	45
6	85
7	76
8	8
9	48
10	76
11	

Step2: Insert 45

∴ h(45) = 45 % 10
= 5.

Step3: Insert 8

h(8) = 8 % 10
= 8.

prob → (1)

Step4: Insert 33

h(33) = 33 % 10
= 3.

Step 5: Insert 85

h(85) = 85 % 10
= 5.

Since 5 is already occupied, 85 get stored in next available location i.e 6. (probe = 1)

Step6: Insert 61.

h(61) = 61 % 10 = 1.

Step7: Insert 10

h(10) = 10 % 10 = 0

Step8: Insert 48

48 % 10 = 8.

Since 8 is already occupied, 48 stored in 9 (next).

Step9: Insert 76 h(76) = 76 % 10 = 6. ∴ (7)

Step10: Insert 89 h(89) = 89 % 10 = 9. ∴ (10)