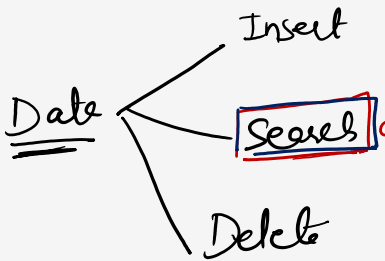
C-Programming and DS

Today Class Topics

-Hashing :- O(1)



dominating operation

⇒ unSorted Array = O(n) // linear search

⇒ Sorted Array = O(log n) // Binary search

⇒ Linked list = O(n) // Linear search

⇒ BST = O(log n) // Avg case

= O(n) // worst case

⇒ AVL-Tree = O(log n)

⇒ Hashing = O(1)

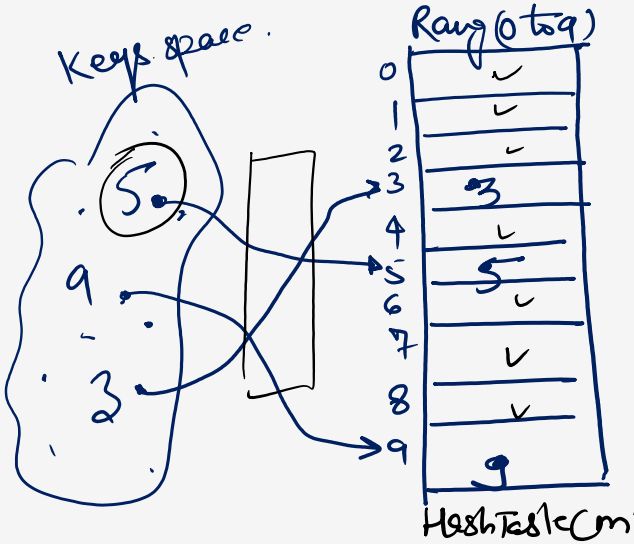
Hashing:- Searching Technique. [Good Hashing is O(1)]

⇒ Worst case search time Complexity = O(1) [99%]

⇒ The following Components are used in Hashing

1) Key space (n)
2) Hash function
3) Hash Table (m) or Hash Bucket (m)

Direct Address Table (DAT) :- Hash table size (m) = 10 (0 to 9)



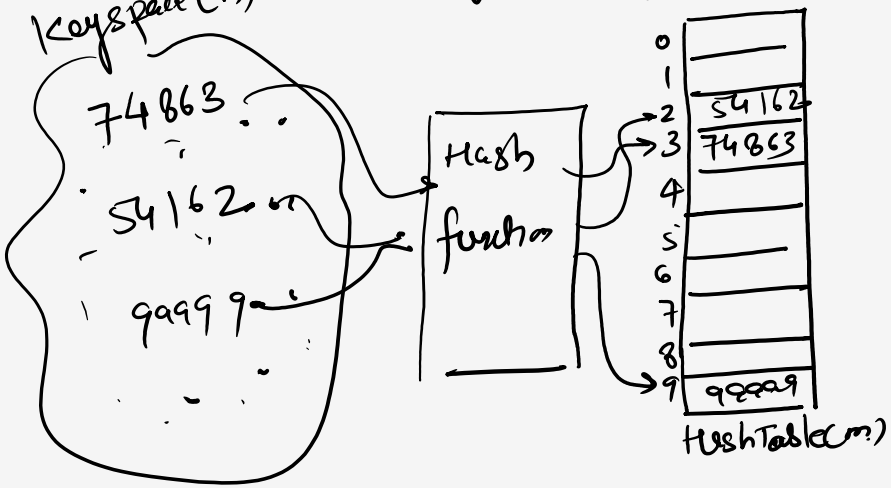
Keys = (5, 4, 6, 105)

Hash function.

Adv: ⇒ WC search time = O(1)

Hash function:- HF(key) = key mod m

Hash Table size = m (0 — m-1)
= 10 (0 — 9)



HF(74863) = 74863 mod 10
= 3

If Key = 899

H(899) = 9 Collision

Collision Resolution Technique

Hash function:-

Ex: 15, 22, 29, 36 (÷7) MSB _____ LSB

1) Division modulo method:-

$$HF(key) = key \bmod m$$

Ex: $m = 8 = 2^3 = 2^k$

$HF(key) =$ LSB K bits

1) key = 86 = 1010110 ⇒ 86 mod 8 = 6 = 110
2) key = 102 = 1100110 ⇒ 102 mod 8 = 6 = 110
3) key = 118 = 1110110 ⇒ 118 mod 8 = 6 = 110
4) key = 126 = 1111110 ⇒ 126 mod 8 = 6 = 110

} Collision

Ex: $m = 7 \neq 2^3 \neq 2^k$

1) key = 86, 86 mod 7 = 2
2) key = 102, 102 mod 7 = 4
3) key = 118, 118 mod 7 = 6
4) key = 126, 126 mod 7 = 0

} No collision

NOTE:

1) do not choose m value as exactly power of 2
2) Choose the m value as prime number which is not nearer to power of 2 then it will give less no. of collision

2) mid square method:- $m = 1000$ (0 — 999)

Key = 786

$(Key)^2 = (786)^2 = 617796$

5 4 3 2 1 0

⇒ It is good Hash function

⇒ Counter Example is difficult to generate

Index	Key
177	786

3) Digit Extraction method:-

Key = 7 3 4 6 8 1 4 3

7 6 5 4 3 2 1 0

Index	Key
486	73468143

⇒ It is bad Hash function

⇒ Counterexample is easy to generate

4) Folding method:-

a) Fold Boundary method:-

Bad Hash function

key = 734 68 143

Key = (734) 68 (143)

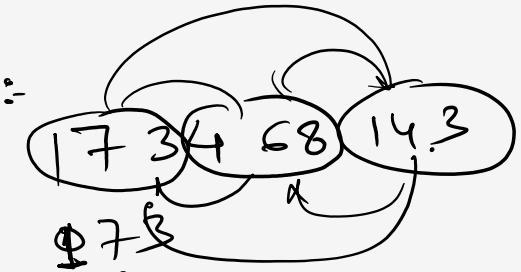
734
143

877

Index	Key
877	734 68 143

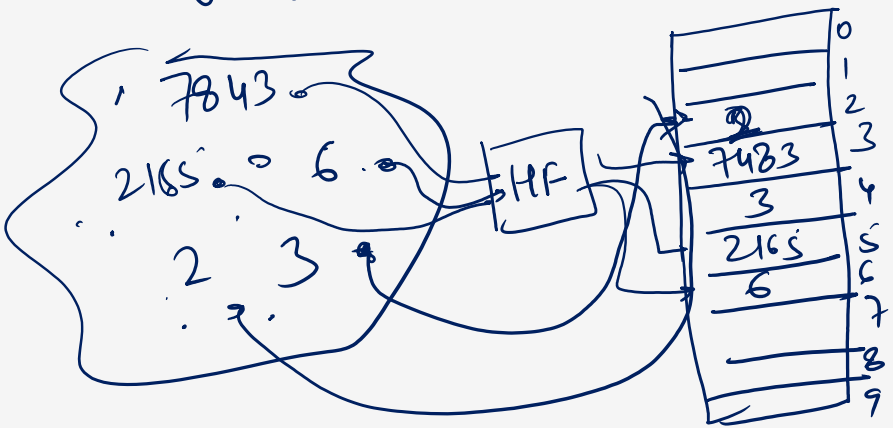
b) Fold shifting method:-

Bad Hash function



734
468
143
---
984 → Hash Index

Hashing:- mapping larger set to smaller set (many to one)



$H(key) = key \bmod m$

Let $m = 10$

Search 7843

$H(7843) = 7843 \bmod 10 = 3$

Problem: Collision

How to Remove Collision:

1st select Better Hash function ⟹ 99% No Collision ⟹ If 1% Collision Then apply Collision Resolution Technique

→ use Better Hash function But never say no Collision

→ use Collision Resolution Technique (CRT)

Collision Resolution Technique (CRT):-

- Chaining (Outside Table)
  - "Key value will be stored outside HashTable"
  - → Insertion from beginning always"
  - → No overflow
  - → Deletion is easy as compared to open Addressing.
- Open Addressing or Closed Hashing (Inside Table)
  - "Key will be stored inside HashTable"
  - → Deletion is Complex
  - → Overflow
    - → (1) Linear Probing
    - → (2) Quadratic Probing
    - → (3) Double Hashing

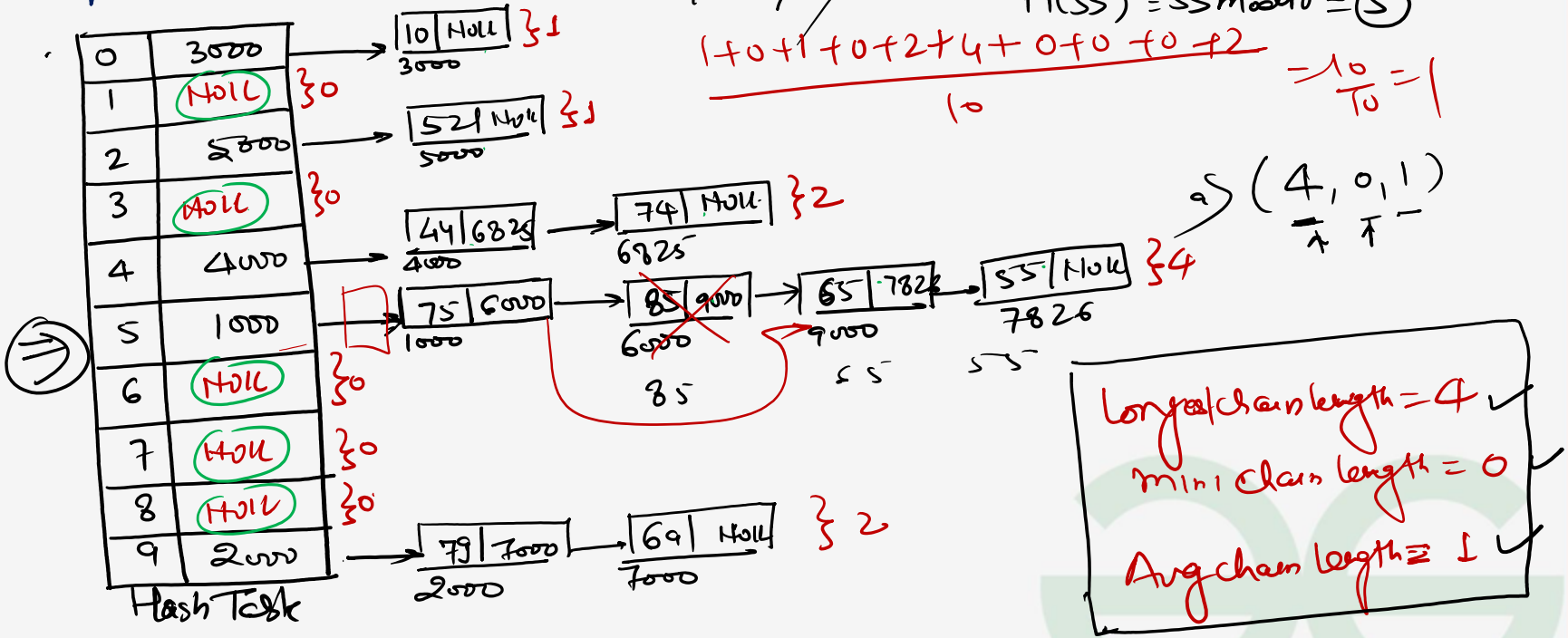
Chaining :- Let M = 10 (0–9), H(K) = K mod m

Keys = ~~10~~, ~~55~~, ~~69~~, ~~74~~, ~~52~~, ~~65~~, ~~79~~, ~~85~~, ~~44~~, ~~75~~

H(10) = 10 mod 10 = 0

H(55) = 55 mod 10 = 5

$\frac{1+0+1+0+2+4+0+0+0+2}{10} = \frac{10}{10} = 1$



a) (4, 0, 1)

Longest chain length = 4

Min chain length = 0

Avg chain length = 1

Remarks

1) It can Handle Unlimited no. of collision (No overflow) → Open Addressing

2) Inefficient Utilization of memory

3) Longest chain length with n keys is n

4) Deletion & Insertion easy as compared to open Addressing

10, 20, 30, 40, 50, 60, 70, 80, 90, 100

$n \to n$

$1 - \frac{n}{n}$

R	BC	WC
Insertion	O(1)	O(1)
Searching	O(1)	O(n)
Deletion	O(1)	O(n)

load factor:- maximum load allocated to each slot

The no. of keys getting stored in one slot is called as load factor

$$\text{load factor } (\alpha) = \frac{n}{m} \quad \left(\frac{\text{no. of keys}}{\text{size of Hash table}}\right)$$

Let $n = km \Rightarrow \alpha = \frac{km}{m} \Rightarrow \alpha = k$

Average:- Search time $= \Theta(1+\alpha)$ ← load factor

Compute Hash function $\Rightarrow \Theta(1+k)$ ← Constant $\Rightarrow \Theta(1)$

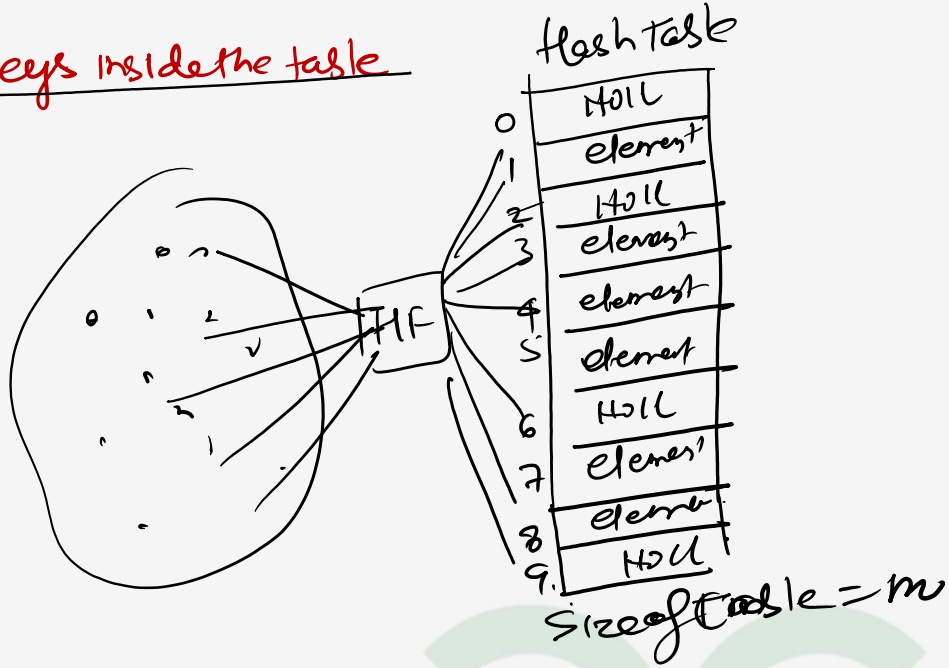
$\Rightarrow \Theta(1)$

Open Addressing :- (Closed Hashing):

We can not Insert more than m keys inside the table

$$\alpha = \frac{n}{m} \Rightarrow \boxed{0 \le \alpha \le 1}$$

$n \le m$.



1) Linear Probing :- → Attempt or checking

Let m = 10 (0–9) → HF(key) = key mod m

LP(key, i) = (HF(key) + i) mod m i = 0, 1, 2, 3 --- m-1

Ex: Keys(n) = 25, 38, 43, 68, 79, 46, 58, 65, 20

1) LP(25, 0) = 5+0 = 5 ✓

2) LP(38, 0) = 8+0 = 8 ✓

3) LP(43, 0) = 3+0 = 3 ✓

4) LP(68, 0) = 8+0 = 8 Collision

LP(68, 1) = 8+1 = 9 ✓

5) LP(79, 0) = 9+0 = 9 Collision

LP(79, 1) = 9+1 = 10%10 = 0 ✓

6) LP(46, 0) = 6 ✓

7) LP(58, 0) = 8+0 = 8 Collision

= 8+1 = 9 Collision

= 8+2 = 0 Collision

= 8+3 = 1 ✓

8) LP(65, 0) = 5 Collision

65, 1 = 6 Collision

68, 2 = 7 ✓

9) LP(20, 0) = 0 Collision

20, 1 = 1 Collision

20, 2 = 2 ✓

9 Collisions

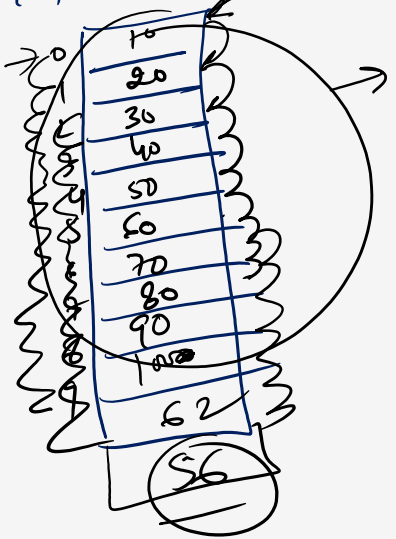
Question Keys(n) = {61, 30, 52, 43, 80, 90, 74, 82}

Let m = 10

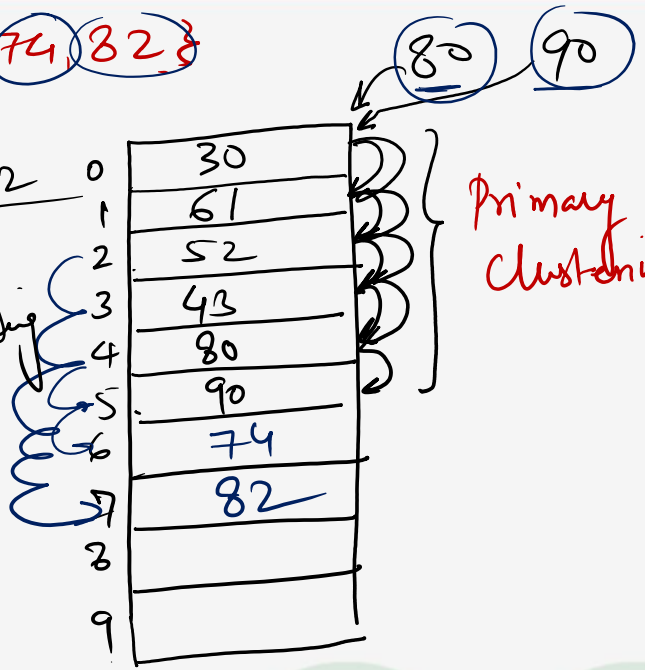
HF(k) = k mod 10

10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 62

56



primary clustering

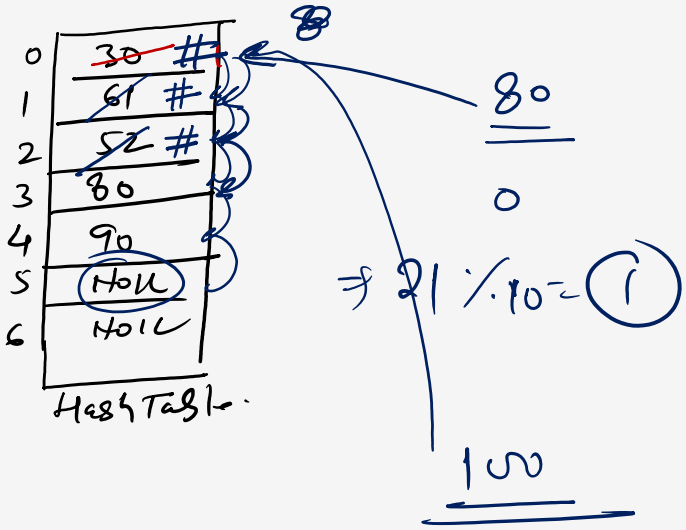


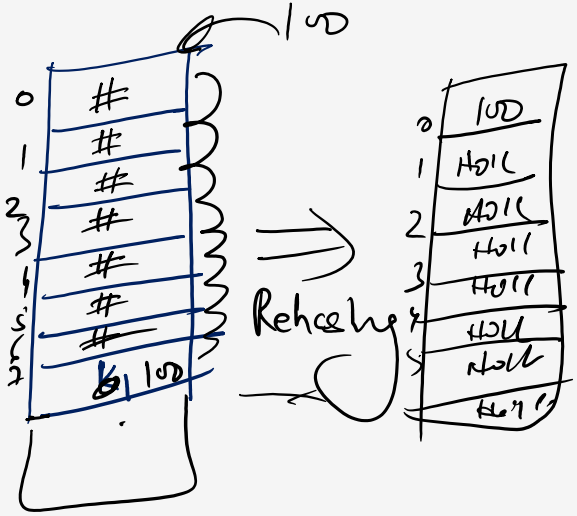
Primary clustering :-

1) If two keys are mapped on to same starting location of the Hash Table then they both follow the same path in the linear manner because of this the search time complexity will increase

2) To Avoid this Problem the Quadratic Probing will be used

Deletion:





Chaining is better for Deletion.

Question:- M = 10, H(K) = K mod m, CRT = LP, After inserting 6 values into an empty HashTable shown below

A) In which sequence Key value could have been inserted to get the above HashTable

a) 46, 42, 34, 52, 23, 33

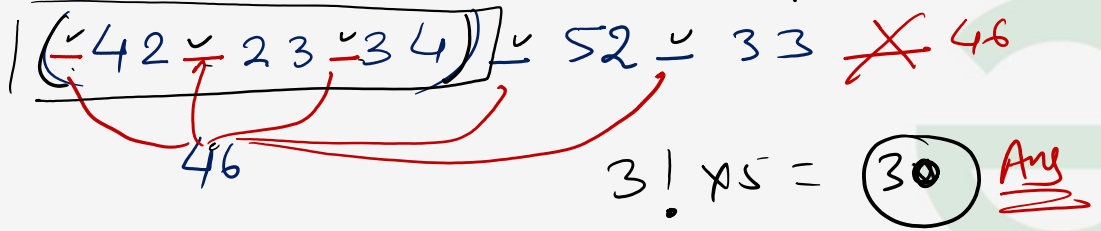
b) 34, 42, 23, 52, 33, 46

⇒ c) 46, 34, 42, 23, 52, 33

d) 42, 46, 33, 23, 34, 52

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

B) How many different insertion sequence Possible to get above HashTable 30 ?



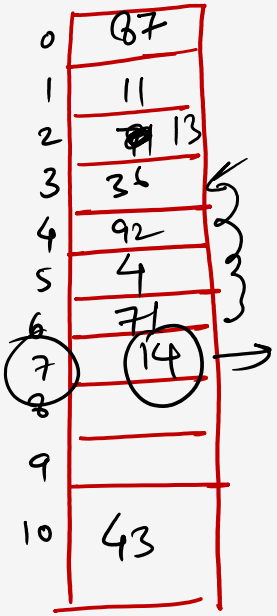
3! x 5 = 30 Any

Question: HTS = 11 (0–10), LP

h(K) = K mod 11

Keys : 43, 36, 92, 87, 11, 4, 71, 13, 14

What is the index into the last record is inserted = ?



7 Ans

2:40 pm

10 minute Break

2) Quadratic Probing:- $HF(key) = key \bmod m$, $m = 10$ (0 — 9)

$$Qp(key, i) = (HF(key) + c_1 * i + c_2 * i^2) \bmod m$$

where $c_1 = 1$, $c_2 = 1$, $i = 0, 1, 2, \ldots - 9\ (m-1)$

Ex: Keys (n) = 25, 98, 57, 75, 97, 18, 78, 46

1) $Qp(25,0) = 5+0+0 = 5$
2) $Qp(98,0) = 8+0+0 = 8$
3) $Qp(57,0) = 7+0+0 = 7$
4) $Qp(75,0) = 5+0+0 = 5$ Collision
   $Qp(75,1) = 5+1+1 = 7$ Collision
   $Qp(75,2) = 5+2+4 = 1$
5) $Qp(97,0) = 7$ collision
   $Qp(97,1) = 9$
6) $Qp(18,0) = 8$ Collision
   $= 0$
7) $Qp(78,0)$
   $= 8$ Coll
   $= 0$ Coll
   $= 4$
8) $Qp(46,0) = 6$

0	18
1	75
2	
3	
4	78
5	25
6	46
7	57
8	98
9	97

Resultant Hash Table (m)

Question: Keys = 41, 60, 82, 93, 70, 80 , $c_1 = c_2 = 1$, Op = ?

$H(K) = K \bmod m$, $m = 10 \ (0-9)$

$HT[\ ?\ ] = 80$

$Op(80,0) = 0$ collision

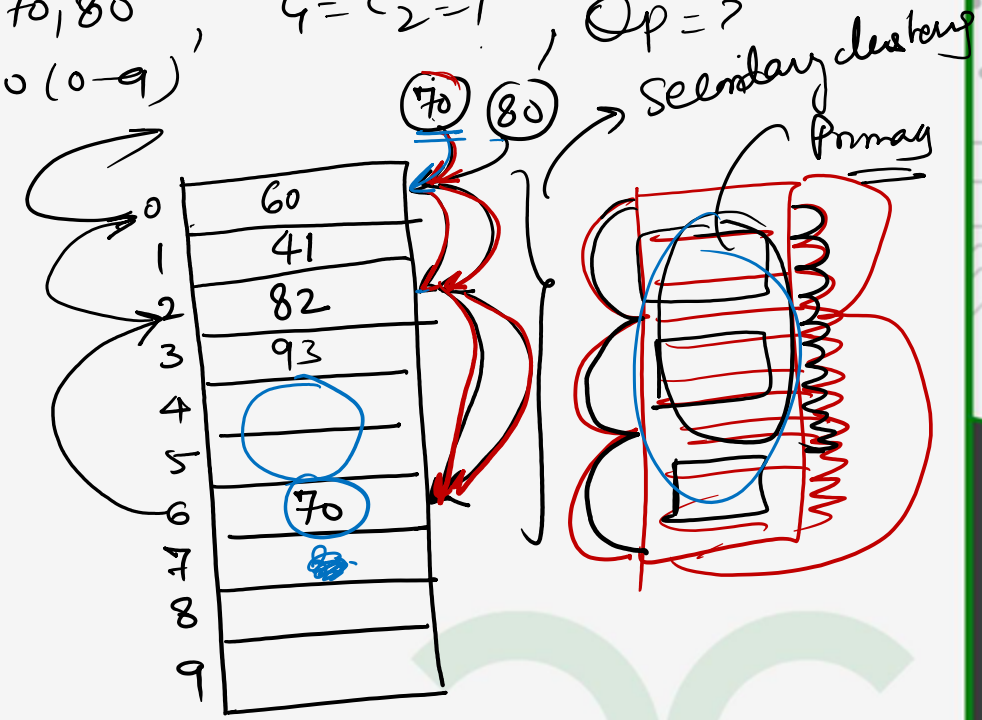
$Op(80,1) = 2$ collision

$Op(80,2) = 6$ collision

$Op(80,3) = 2$ collision

$Op(80,4) = 0$ collision

$Op(80,5) =$



Secondary clustering :-

If two keys are mapped onto same starting location in the Hash Table then they both follow the same path in the Quadratic manner because of the search time complexity will increase

Note :

TC	BC	WC
Insertion	O(1)	O(n)
Searching	O(1)	O(n)
Deletion	O(1)	O(n)

Note : To avoid this problem Double Hashing will be used

3) Double Hashing :-

$HF_1(key) = key \bmod m$

$HF_2(key) = 1 + (key \bmod (m-2))$

$m = 10$ (0 — 9)

$i = 0, 1, 2, 3, \ldots, m-1$

$$DH(key, i) = (HF_1(key) + i * HF_2(key)) \bmod m$$

Ex:- keys(n) = 25, 98, 57, 75, 97, 18, 78, 65, 46

1) $DH(25, 0) = 5$

2) $DH(98, 0) = 8$

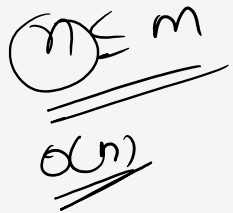
$DH(75, 0) = 5$ Collision

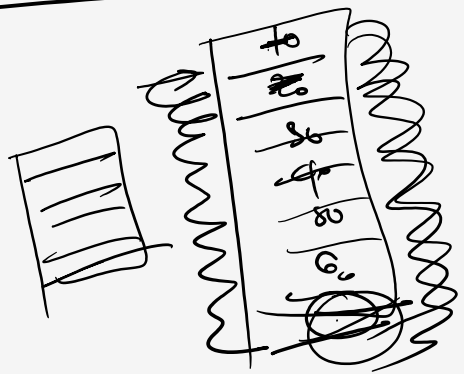
$DH(75, 1) = (5 + 1 + 3) \bmod 10 = 9$

0	
1	97
2	78
3	65
4	18
5	25
6	46
7	57
8	98
9	75

TC	BC	WC
Insertion	O(1)	O(m)
Searching	O(1)	O(m)
Deletion	O(1)	O(m)

WC = O(1) [99%]





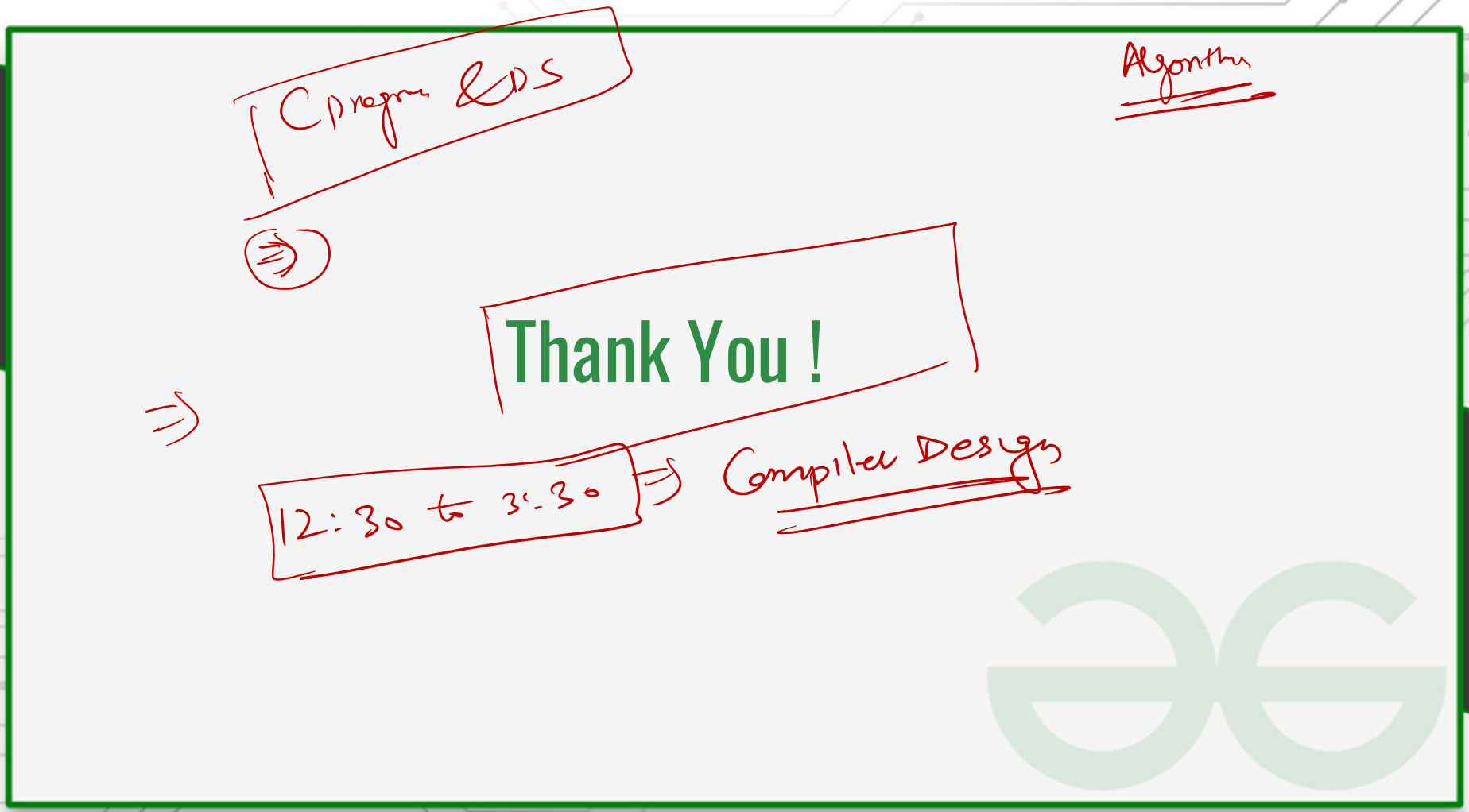
Universal Hashing ⇒ O(1)

NOTE:-

1) The expected no. of Probes in an unsuccessful search of open Addressing Technique is $= \boxed{\frac{1}{1-\alpha}}$, where α is load factor $= n/m$

2) The expected no. of probes in Successful search of open Addressing Technique is $= \boxed{\frac{1}{\alpha} \log \frac{1}{1-\alpha}}$, where α is load factor

$$\alpha = \frac{n}{m}$$


C Program & DS
Algorithm
Thank You !
12:30 to 3:30
Compiler Design