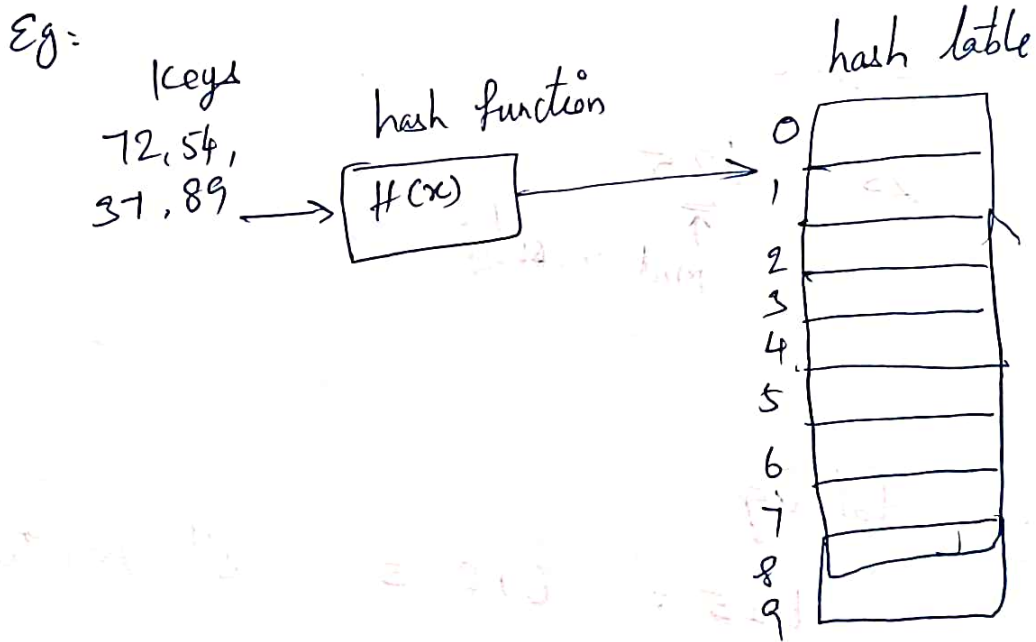


UNIT - IV

Hashing:

- It is a DS
- It is a technique of mapping large chunk of data into small tables using a hashing function.
- Also known as message digest function.



Hash function

→ Put data into table

→ Types

- ① Division method
- ② Mid square
- ③ Digit folding.

No Division Method

$$H(\text{key}) = \text{key} \% \text{Table size}$$

eg: $H(54) = 54 \% 10 = 4$

$$H(72) = 72 \% 10 = 2$$

$$H(37) = 37 \% 10 = 7$$

$$H(89) = 89 \% 10 = 9$$

② Mid square

$$25^2 = 625$$

↑
mid position

③ Digit folding

$$625 = 6 + 2 + 5 = 13^{\text{th}} \text{ position}$$

Collision

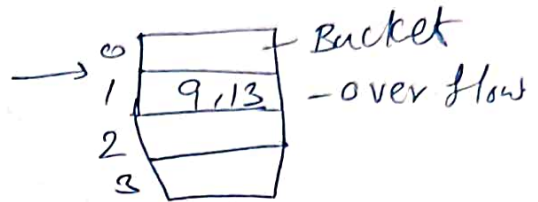
Key: 9, 13,

$H(x)$

$$9 \% 4 = 1$$

$$13 \% 4 = 1$$

'9' will be deleted.



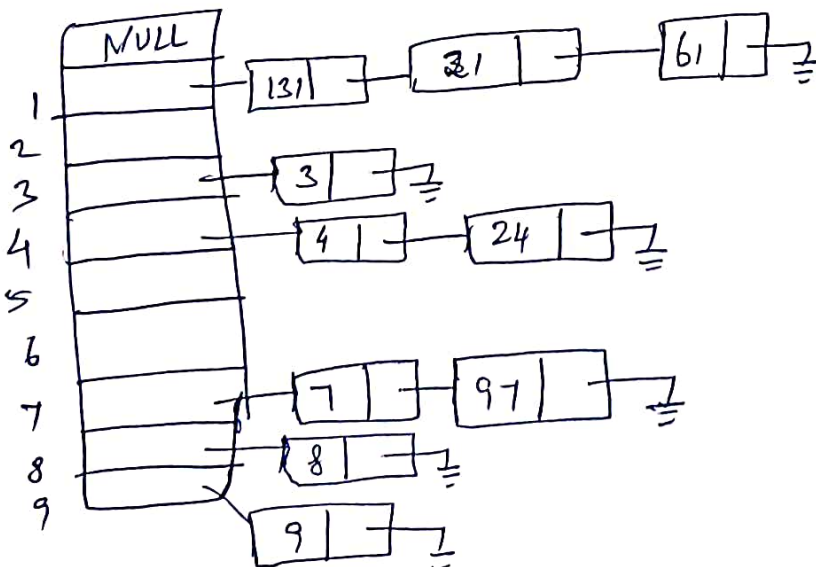
Techniques to avoid collision

- ① separate chaining / External hashing
- ② open addressing / closed hashing

Separate chaining

Eg-1

131, 3, 4, 21, 61, 24, 7, 97, 8, 9



- An additional field with data i.e. chain
- separate chain table is maintained for colliding data.

Open Addressing

Three types of collision resolution strategies.

Linear Probing

Quadratic " $[\text{hash}(x) + i^2] \% \text{HTS}$

Double hashing $\begin{matrix} 0 \\ 1 \\ 4 \\ 9 \end{matrix}$

Linear Probing:

89, 18, 49, 58, 69

$$h_i(x) = (\text{Hash}(x) + i) \% \text{Hash table size}$$

if $h_0 = (\text{hash}(x) + 0) \% \text{HTS}$ is full we try for h_1
 $h_1 = (\text{hash}(x) + 1) \% \text{HTS}$ " " " " h_2
 so on.

Eg., 89, 18, 49, 58, 69

① Insert 89

$$h_0(89) = (89 \% 10) = 9$$

② Insert 18

$$h_0(18) = (18 \% 10) = 8$$

③ Insert (49)

$$h_0(49) = (49 \% 10) = 9$$

Collision.

$$h_1(49) = (49 + 1 \% 10)$$

$$50 \% 10 = 0$$

④ Insert 58

$$h_0(58) = (58 \% 10) = 8 \quad \text{full} \quad \text{calculate } h_1$$

$$h_1(58) = \overset{(58+1)}{(59 \% 10)} = 9 \quad \text{"} \quad \text{"} \quad h_2$$

$$h_2(58) = (60 \% 10) = 0 \quad \text{"} \quad \text{"} \quad h_3$$

$$h_3(58) = (58 + 3) \% 10 = 1$$
$$61 \% 10$$

⑤ Insert 69

$$h_0(69) = 69 \% 10 = 9 \quad \text{occupied} \quad \text{calculate } h_1$$

$$h_1(69) = 69 + 1 \% 10 = 0 \quad \text{"} \quad \text{"} \quad h_2$$

0	49
1	58
2	69
3	
4	
5	
6	
7	
8	18
9	89

$$h_2(69) = 69 + 2 \% 10 = 1 \text{ occupied, calculate } h_3$$

$$h_3(69) = 69 + 3 \% 10 = 72 \% 10 = 2 \text{ place the value}$$

Double Hashing:

$$h_i(x) = (\text{hash}_1(x) + i * (\text{hash}_2(x))) \% \text{HTS}$$

Quadratic

$$h_i(x) = (\text{hash}(x) + i^2) \% \text{Hash table size}$$