Hashing: Part II

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Vishvadeep Gothi • Lesson 65 • Feb 16, 2023

Data Structure
Hashing 2

By: Vishvadeep Gothi

Hashing

Searching technique which can provide result in constant time

$L = H(k)$

Hashing Techniques

1. Direct Hashing
2. Subtraction Method
3. Division Method
4. Fold Shifting Method
5. Fold Boundary Method
6. Digit Extraction Method
7. Mid-square Method

Collision

Collision Resolution Techniques

open addressing
or
closed hashing

- Linear probing
- Quadratic "
- Random "
- Double Hashing

closed addressing
or
open hashing

- open chaining

Linear Probing

$L = H(k)$

There is a collision at L.

$$L_{new} = (L + i) \bmod size \qquad i = 1, 2, 3, 4, \text{-------}$$

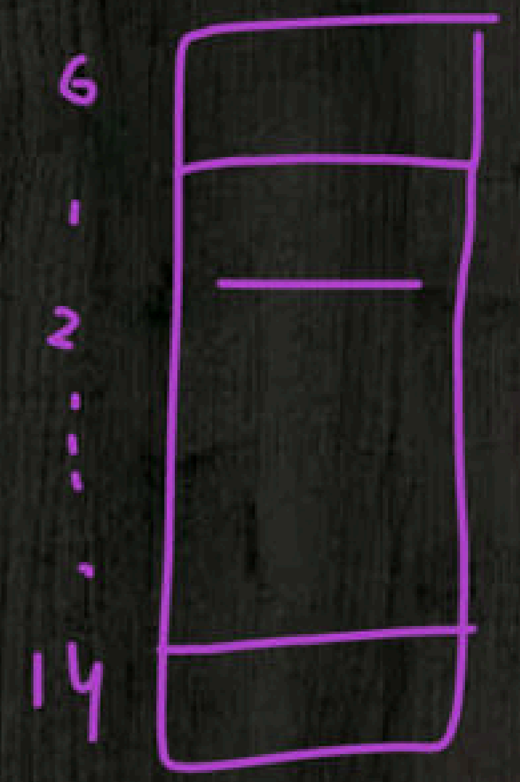
$L = 6$

$6 + 1 = 7 \Rightarrow$

$6 + 2 \Rightarrow 8 \Rightarrow$

$6 + 3 \Rightarrow 9 \Rightarrow$

$\vdots$



Hash table

size = 15

locations 0 to 14

Ques)

$H(k) = k \bmod 10$

linear probing used

keys to store :- 23, 32, 49, 43, 5, 15, 22, 92, 13

			0		1	5	6	7
23	32	49	43	5	15	22	92	13

0	1	2	3	4	5	6	7	8	9
13		32	23	43	5	15	22	92	49

Total no. of collisions = 20

Quadratic Probing

$L = H(k)$

There is a collision at L,

then

$$L_{new} = (L + i^2) \bmod size\_of\_table \qquad i = 1, 2, 3, 4,$$

ex:- assume a hash table of size 15 (Range :- 0 to 14).

$L = 3$ if collision at 3

Probing

$L_{new} = (3 + 1^2) \bmod 15 = 4$

$= (3 + 2^2) \bmod 15 \Rightarrow 7$

$= (3 + 3^2) \bmod 15 \Rightarrow 12$

$= (3 + 4^2) \bmod 15 \Rightarrow 4$

$= (3 + 5^2) \bmod 15 \Rightarrow 13$

$= (3 + 6^2) \bmod 15 \Rightarrow 9$

Random Probing

$L = H(k)$

r = random number

There is a collision at L.

$L_{new} = (L + i * r) \bmod size$

$i = 1, 2, 3, 4, 5, 6, ---$

or

$L_{new} = (L_{prev} + r) \bmod size$

ex:- Table size 15 (0 − 14)

$L = 3$

$r = 4$

$3 + 4 = 7$

$3 + 2 * 4 = 11$

$(3 + 3 * 4) \bmod 15 = 0$

$= 4$

Ex :-

$H(k) = k \bmod 11$ $\qquad x = 4$

keys $\Rightarrow$ 6, 17, 28

for 6 :-

$L = 6 \bmod 11 = 6$

for 17 :-

$L = 17 \bmod 11 = 6$ Collision

$L_{new} = (6 + 4) \bmod 11 = 10$

for 28 :-

$L = 28 \bmod 11 = 6$ Collision

$L_{new} = (6 + 4) \bmod 11 = 10$ Collision

$L_{new} = (10 + 4) \bmod 11 = 3$

0	
1	
2	
3	28
4	
5	
6	6
7	
8	
9	
10	17

Double Hashing

$L = H(k)$ $r = H_2(k)$

If there is a Collision at L.

$$L_{new} = (L + i * r) \bmod size$$ $r \neq 0$

or

$$L_{new} = (L_{prev} + r) \bmod size$$

Ex:- $H(k) = k \bmod 11$

$H_2(k) = (k \bmod 9) + 2$

keys $\Rightarrow$ 6, 17, 28

for 6:- $6 \bmod 11 = 6$

for 17:- $17 \bmod 11 = 6$ Collision

$r = (17 \bmod 9) + 2 = 10$

$L_{new} = (6 + 10) \bmod 11 = 5$

for 28:- $28 \bmod 11 = 6$ Collision

$r = (28 \bmod 9) + 2 = 3$

$L_{new} = (6 + 3) \bmod 11 = 9$

0	
1	
2	
3	
4	
5	17
6	6
7	
8	
9	28
10	

Ques) $H(k) = k \bmod 19$ $\qquad H_2(k) = (k \bmod 13) + 1$

key = 132

location is 3rd probe, if probe seq. starts from 1.

Ans :- Probe 1 :- $L = 132 \bmod 19 = 18$

$r = (132 \bmod 13) + 1 = 3$

Probe 2 :- $(18 + 3) \bmod 19 = 2$

Probe 3 :- $(2 + 3) \bmod 19 = 5$

Ans = 5

Chaining

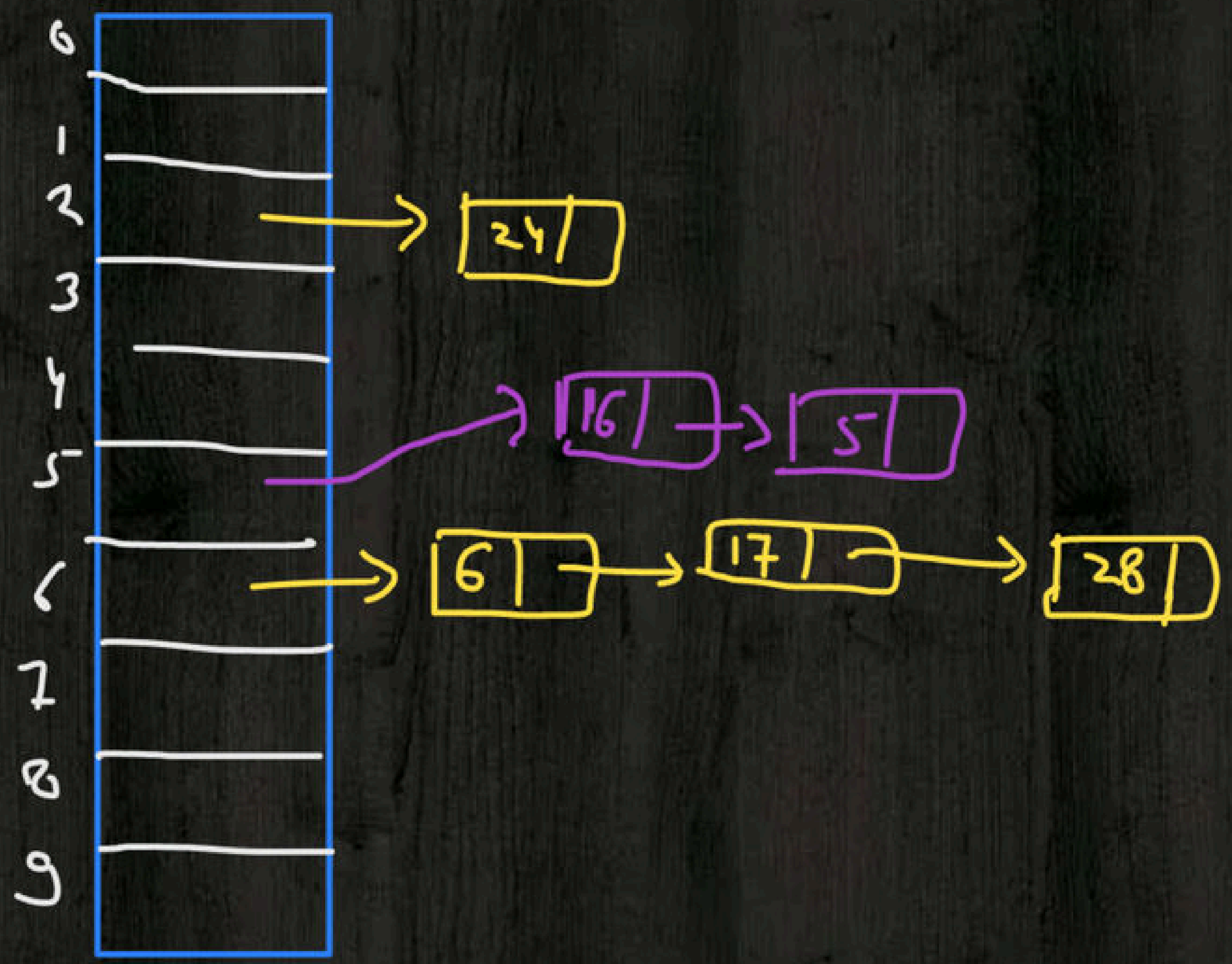
Elements are not stored in hash table slots.

Elements are stored in a linked-list, starting from slot.

ex:-

$$H(k) = k \bmod 11$$

keys $\Rightarrow$ 6, 17, 28, 24, 16, 5



Clustering

Primary clustering:
The tend is for long sequence of preoccupied positions still become longer, primarily at one place

Secondary clustering:
The tend is for long sequence of preoccupied positions still become longer, primarily at different places

Open Addressing vs Closed Addressing

if collision at one address then other addresses are open to be generated

Closed Hashing :- no. of keys are limited by table size.

no other add. is open to be generated

Open Hashing :- no. of keys stored is not limited by table size.

Open Addressing vs Cloased Addressing

Disadvantages of open addressing:

1. Collided records require more probes
2. Deletion not possible
3. Overflow problem

Advantages of chaining:

1. Collided records require less probes
2. Deletion possible
3. No overflow problem

Chaining

Advantages:

1) Simple to implement.
2) Hash table never fills up, we can always add more elements to the chain.
3) Less sensitive to the hash function or load factors.
4) It is mostly used when it is unknown how many and how frequently keys may be inserted or deleted.

Chaining

Disadvantages:

1) Cache performance of chaining is not good as keys are stored using a linked list. Open addressing provides better cache performance as everything is stored in the same table.
2) Wastage of Space (Some Parts of hash table are never used)
3) If the chain becomes long, then search time can become O(n) in the worst case.
4) Uses extra space for links.

Perfect Hashing

no. of slots => n

no. of keys => n

distribute n keys. over n slots w/o collision.

<u>ex:-</u> student to exam sitting arrangement

Minimal Perfect Hashing

no. of keys $\Rightarrow n$

no. of slots $\Rightarrow m$ $\qquad m < n$

distribute first m keys on m slots w/o collision, then for remaining all $n-m$ keys there is collision.

Universal Hashing / Universal Hash function

It refers to selecting a hash function randomly from all available hash functions based on some mathematic properties, to guarantee min. expected collisions.

Load Factor (α)

average no. of keys per slot

$$\alpha = \frac{\text{no. of keys}}{\text{no. of slots}}$$

open addressing :-

$$0 \leq \alpha \leq 1$$

chaining :-

$$0 \leq \alpha$$

Space Utilization

fraction of occupied slots as compared to total no. of slots

$$\text{Space utilization} = \frac{\text{no. of occupied slots}}{\text{total no. of slots}}$$

for open addressing $\Rightarrow$ load factor = space utilization

for all

$$0 \leq \text{Space utilization} \leq 1$$

Ques) Consider a hash table with 50 slots. The table is used to store 3000 keys occupying 30 slots of table.

load factor = ?

space utilization = ?

Ans:-

$$\text{load factor} = \frac{3000}{50} = 60$$

$$\text{space utilization} = \frac{30}{50} = 0.6$$

Happy Learning