Hashing Techniques

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

Vishvadeep Gothi • Lesson 64 • Feb 15, 2023

Data Structure
Hashing

By: Vishvadeep Gothi

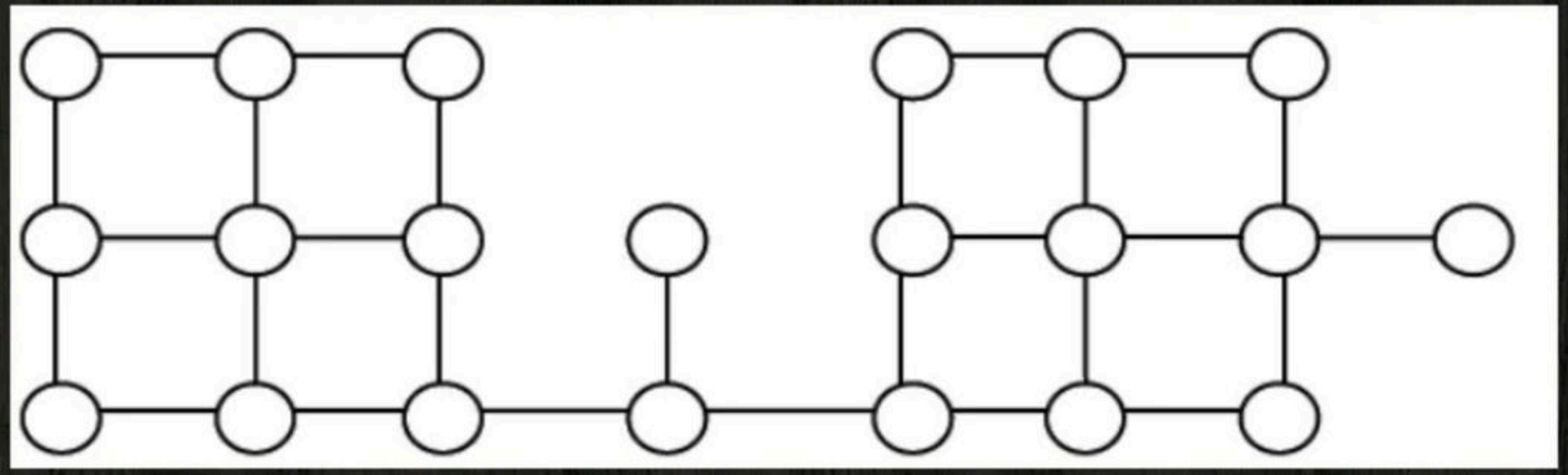
Hello!

I am Vishvadeep Gothi

I am here because I love to teach

Question 4 GATE-2014

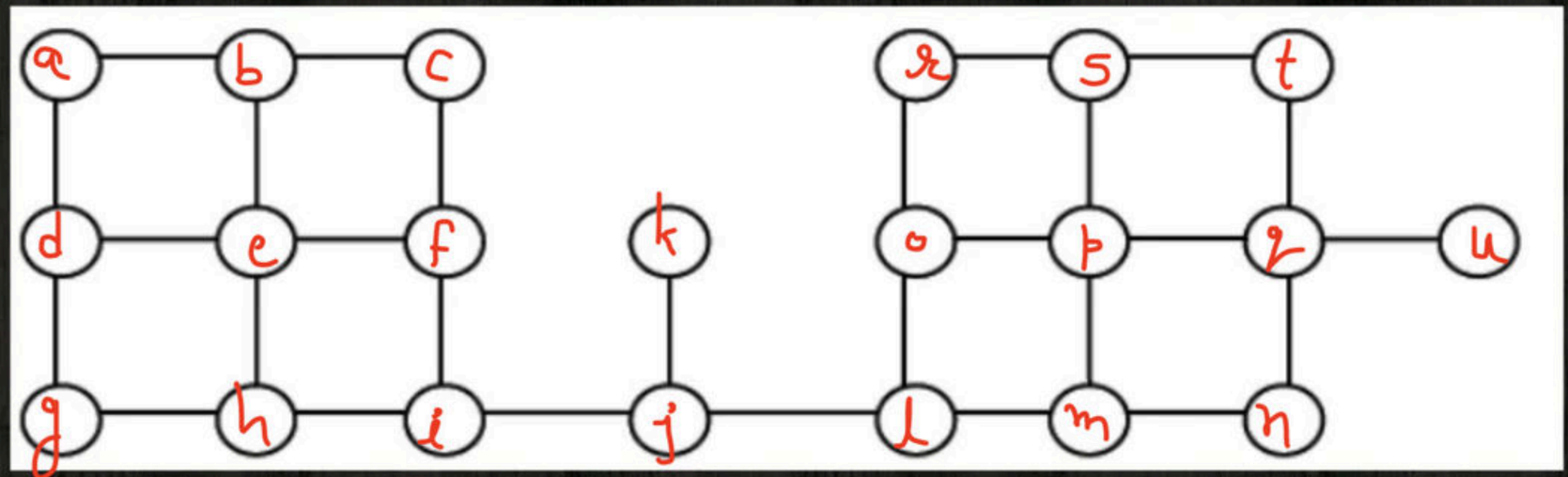
Suppose depth first search is executed on the graph below starting at some unknown vertex. Assume that a recursive call to visit a vertex is made only after first checking that the vertex has not been visited earlier. Then the maximum possible recursion depth (including the initial call) is _____.

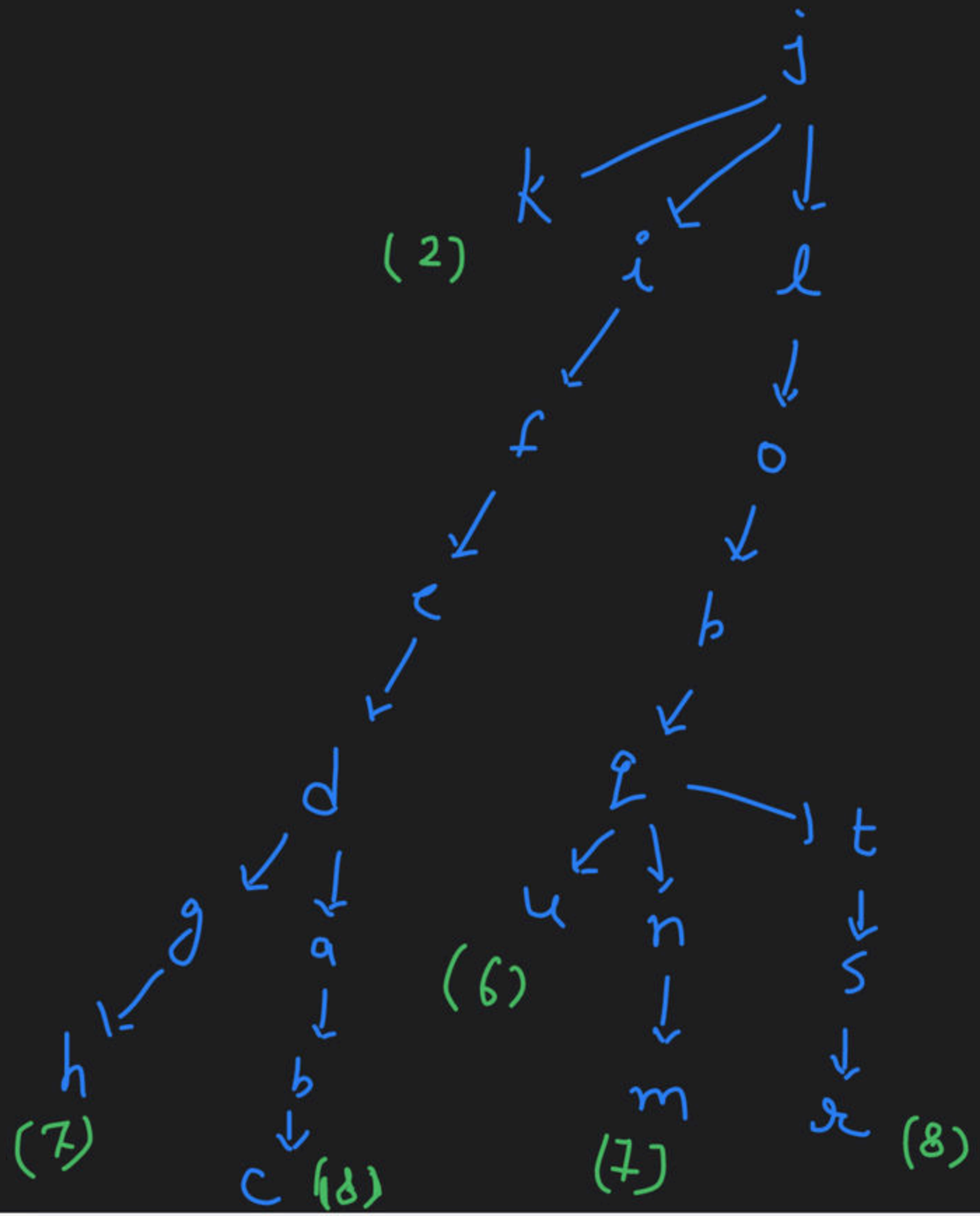


Ans = 8

Question 5

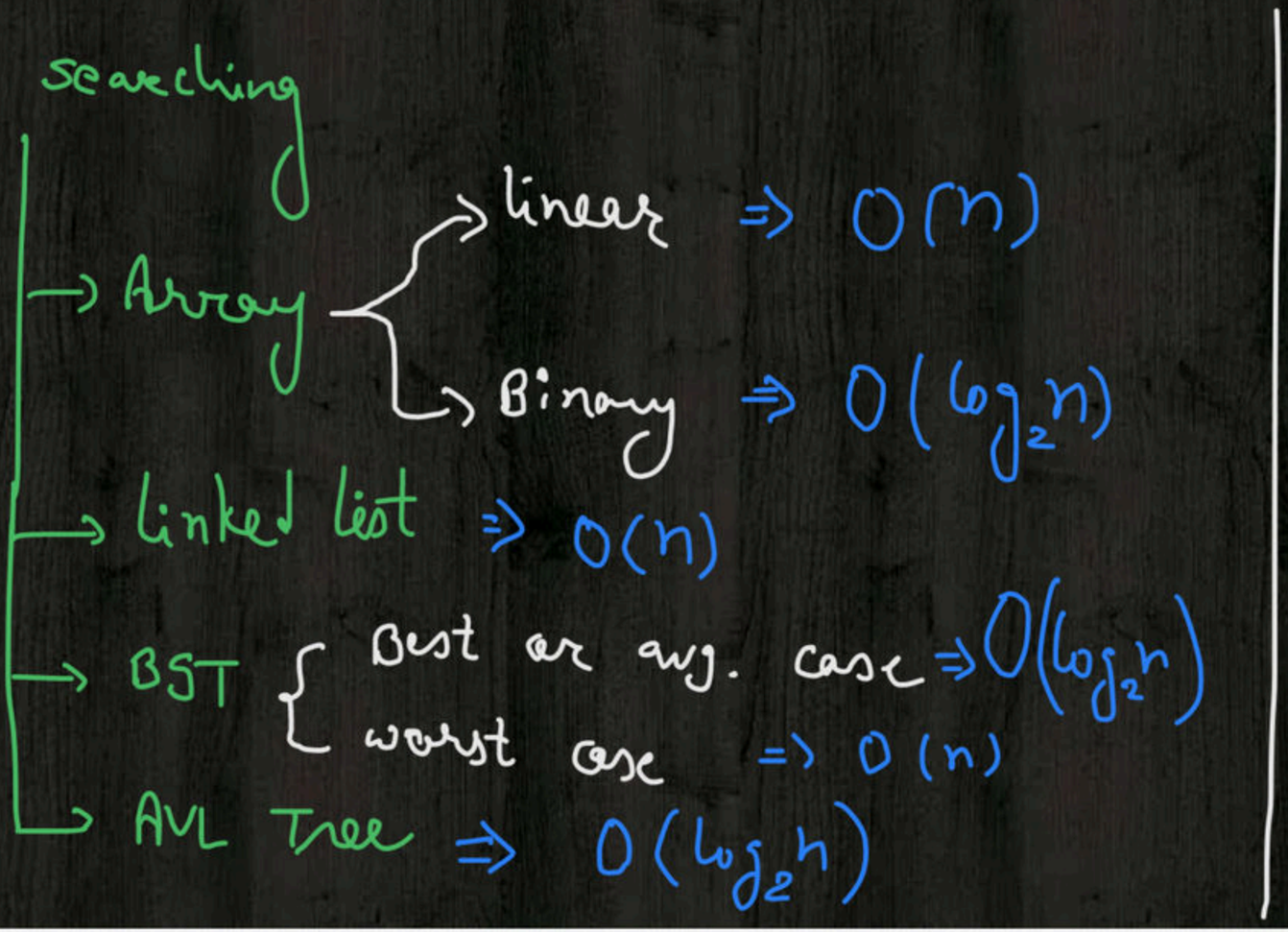
Suppose depth first search is executed on the graph below starting at some unknown vertex. Assume that a recursive call to visit a vertex is made only after first checking that the vertex has not been visited earlier. Then the maximum possible recursion depth in best case (including the initial call) is 8.




j
k
i
l
(2)
f
o
c
b
d
g
u
n
t
s
g
a
h
b
m
r
c
(6)
(7)
(8)
(7)
(8)

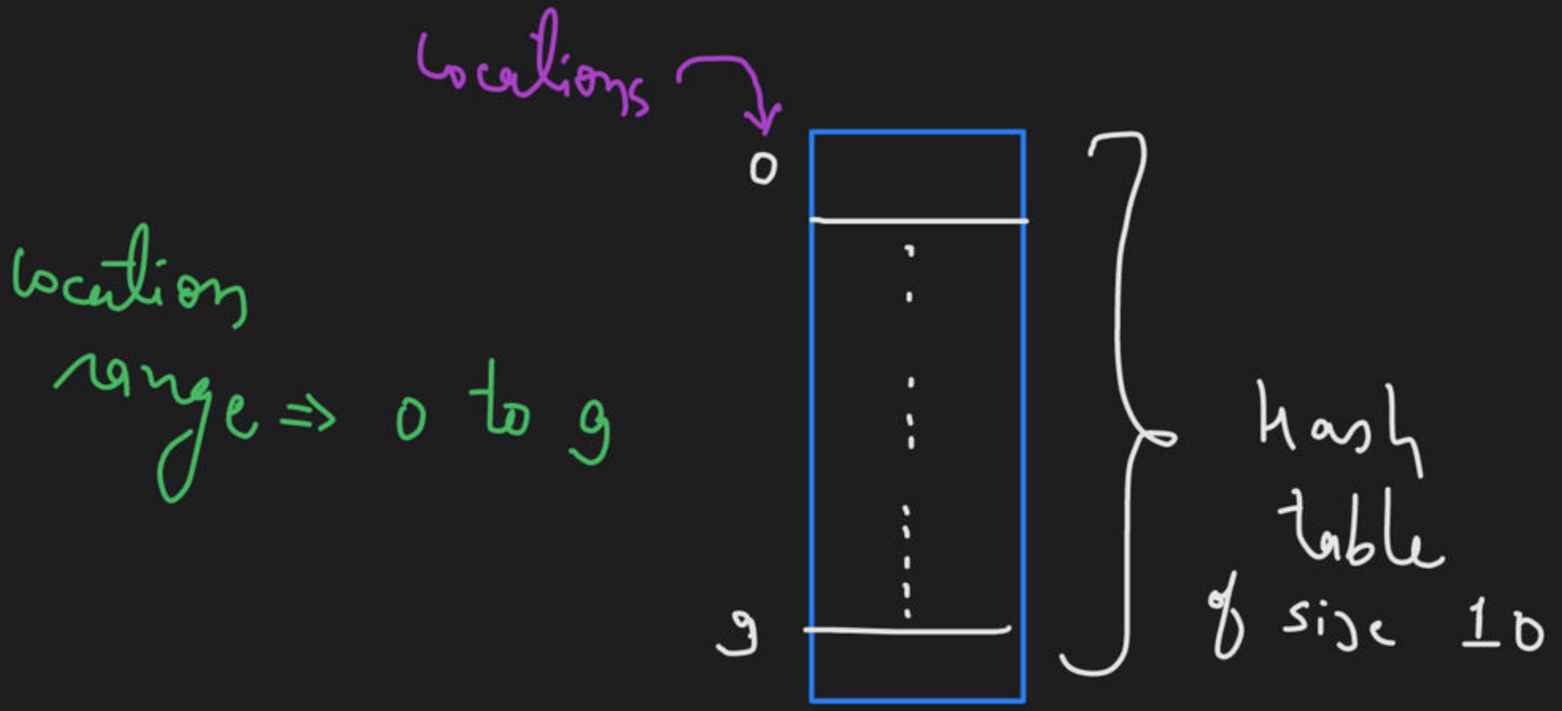
Hashing

Searching technique which can provide result in constant time



Hash function $H()$ ⇒ applied on key ⇒ result = location

$$\text{Location} = H(\text{key})$$



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

Direct Hashing

location = H(key) = key

key => 15	21	1	90
Locat^n => 15	21	1	90

range of keys limited by location range

Assume there is a hash table with location range 00 – 99.

key Range => 0 to 99

Subtraction Method

location = H(k) = k - m m is an integer

Example:-

H(k) = k - 4 keys range limited by location range

if location range 00 to 99

keys Range ⇒ 4 to 103

Division / Modulo Division Method

Location = $H(key) = key \bmod m$

m ⇒ is an integer

location range ⇒ 0 to m−1

ex:- $H(k) = k \bmod 5$

ex:- 152

$152 \bmod 5 \Rightarrow 2$

location range ⇒ 0 to 4

any key can be stored.

Question 1

Which of the following is suitable hash function to have a range of locations from 1 to 1000?

A. $H(X) = X\%1000$ → 0 to 999

B. $H(X) = X\%999 + 1$ → (0 to 998) + 1 => 1 to 999

C. $H(X) = (X+1)\%\ 1000$ → 0 to 999

D. $H(X) = (X\%1000)+1$ → (0 to 999) + 1 => 1 to 1000

E. $H(X) = X\%(1000+1)$

↓

x % 1001 ⇒ 0 to 1000

fold shifting

if location range => 000 - 999

key :- 123|123|12

```
  123
+ 312
  312
-----
  547
```

<= location

or

```
  123
+ 312
  211
-----
```

<- reverse of centre fold

if sum > 999

sum % 1000 => location

Fold boundary

if location range => 000 - 999

key :- 321 112531

```
  312
+ 531
-----
  843
```

⇒ location

if sum > 999

(sum % 1000) ⇒ location

key :- 12345

12 345 or 12345

```
  345
+  12
-----
```

```
  123
   45
-----
```

Digit Extraction

if location range ⇒ 000 to 999

$$H(k) = (key)_{1,4,5}$$

ex:- key = 49245 3

$H(k)$ = 923 or 329

445

Mid square:-

if location range ⇒ 000 – 999

$$H(k) = (k^2)_{\text{middle 3 digits}}$$

ex:- $k = 136$

$$H(k) = (136^2)_{\text{middle}} = (18496)_{\text{middle}}$$

$$= \underline{\underline{849}}$$

if key = 58

$58^2 = 3364$

↓

middle ⇒ 36

⇒ 364

⇒ 336

Collision

if H(k1) => location L

Collision at L, if location L is preoccupied.

If for a key, hash function generates a preoccupied location, then there is a collision at that location.

ex:- $H(k) = k \bmod 10$ ⇒ location range ⇒ 0 to 9

keys to store :-

32, 64, 55, 62

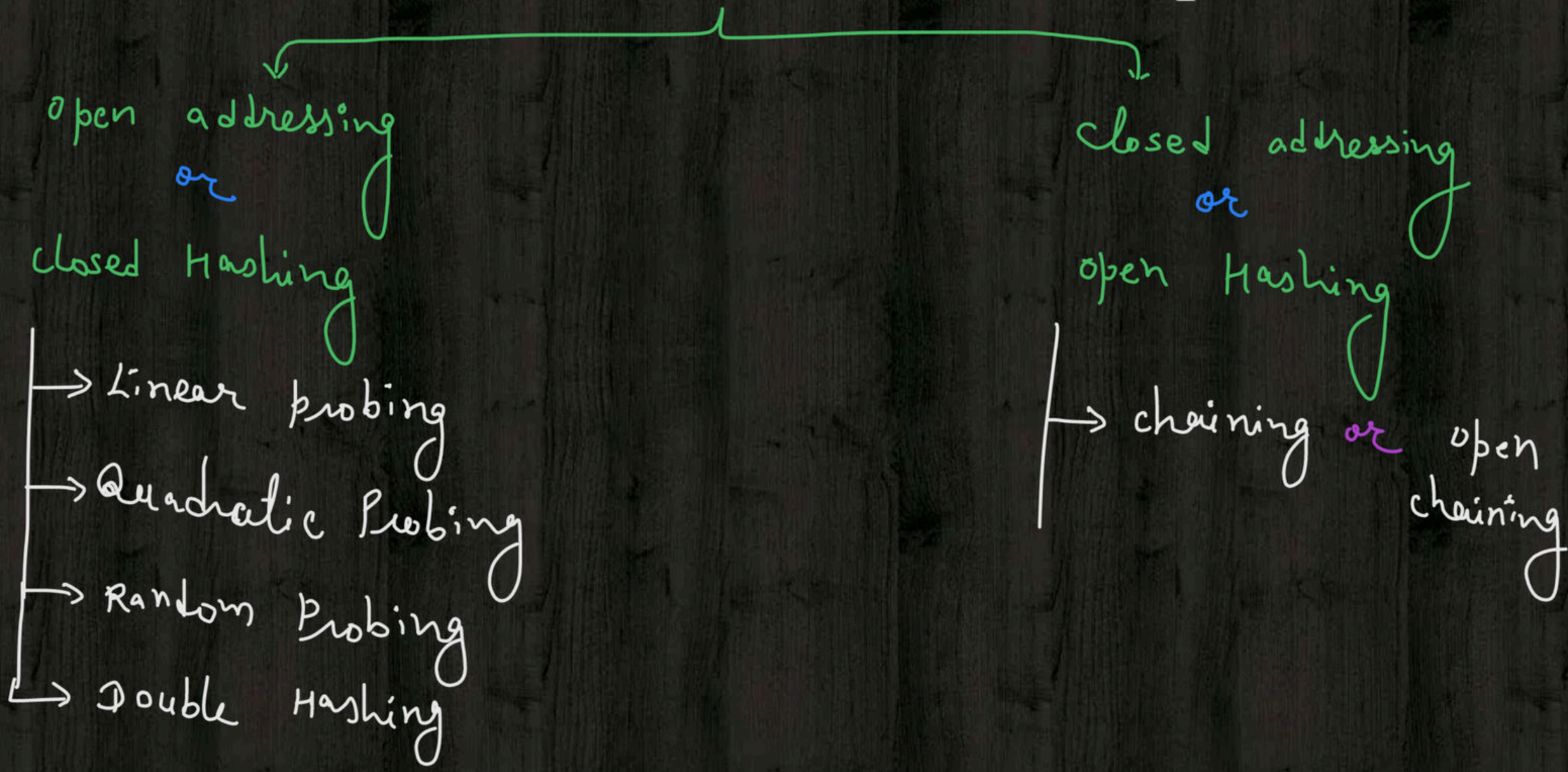
32 % 10 ⇒ 2

64 % 10 ⇒ 4

55 % 10 ⇒ 5

62 % 10 ⇒ 2 ⇒ Collision

Collision Resolution Techniques



Linear Probing

Quadratic Probing

Random Probing

Double Hashing

Chaining

Happy Learning