Computer Science & IT

ALGORITHMS

Algorithms

Lecture No. 09

By- Ravindra Sir

Recap of Previous Lecture

Topic

Topic

Topic

Sorting

Topics to be Covered

Topic Hashing

Topic

Topic

Inspiring Stories : Billy Long Jr.

Background: From Essex, UK. Has autism and ADHD.

Struggle: Trained in a small garage, often mocked.

Achievements: Won European boxing gold in 2024.

Impact: Proved disability can be strength.

Inspiring Stories : Shea Foster

Background: From USA. Car crash in 2021 broke his body.

Struggle: Doctors said he may not run again.

Achievements: Came back as Paralympic runner.

Impact: Will power will beats fate.

Inspiring Stories : Shachindra Bisht

Background: **From Nainital, India. Lost both legs.**

Struggle: **Had to quit mountain climbing.**

Achievements: **Became painter and filmmaker.**

Impact: **Turned pain into art.**

Hashing :-

mainly DS are used to provid i/p to algorithm

main operation

Insertion

Search → very frequent

deletion

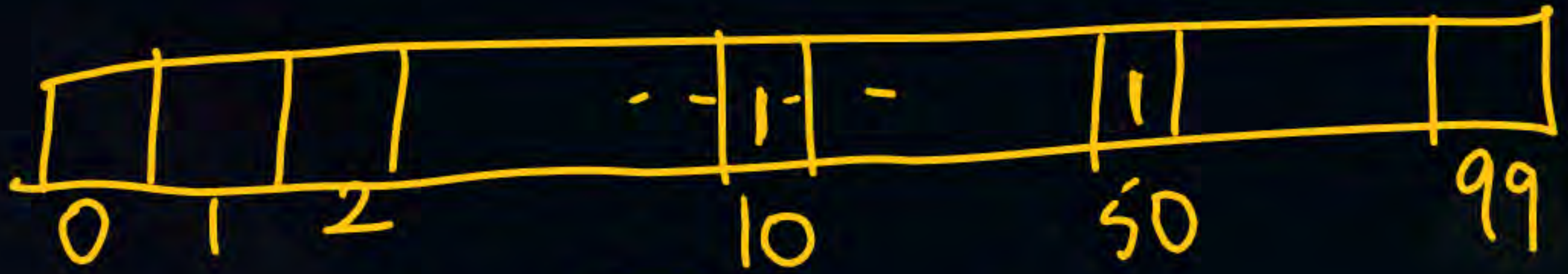
Searching :-

unsorted array - O(n)
Sorted array - O(log n)
Linked list - O(n)
BT - O(n)
Binary search tree - O(n)
Balanced BST - O(log n)
min max heap - O(n)

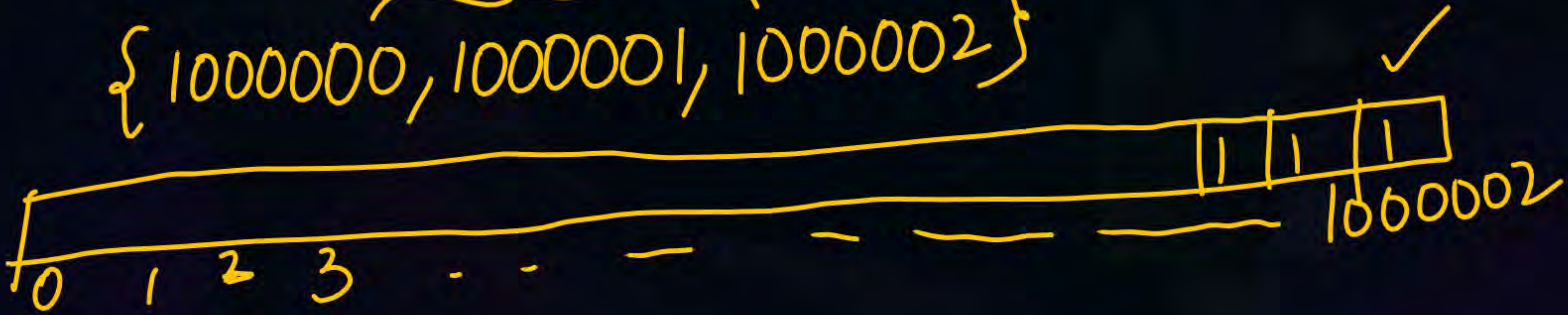
hashing - O(1) - averge

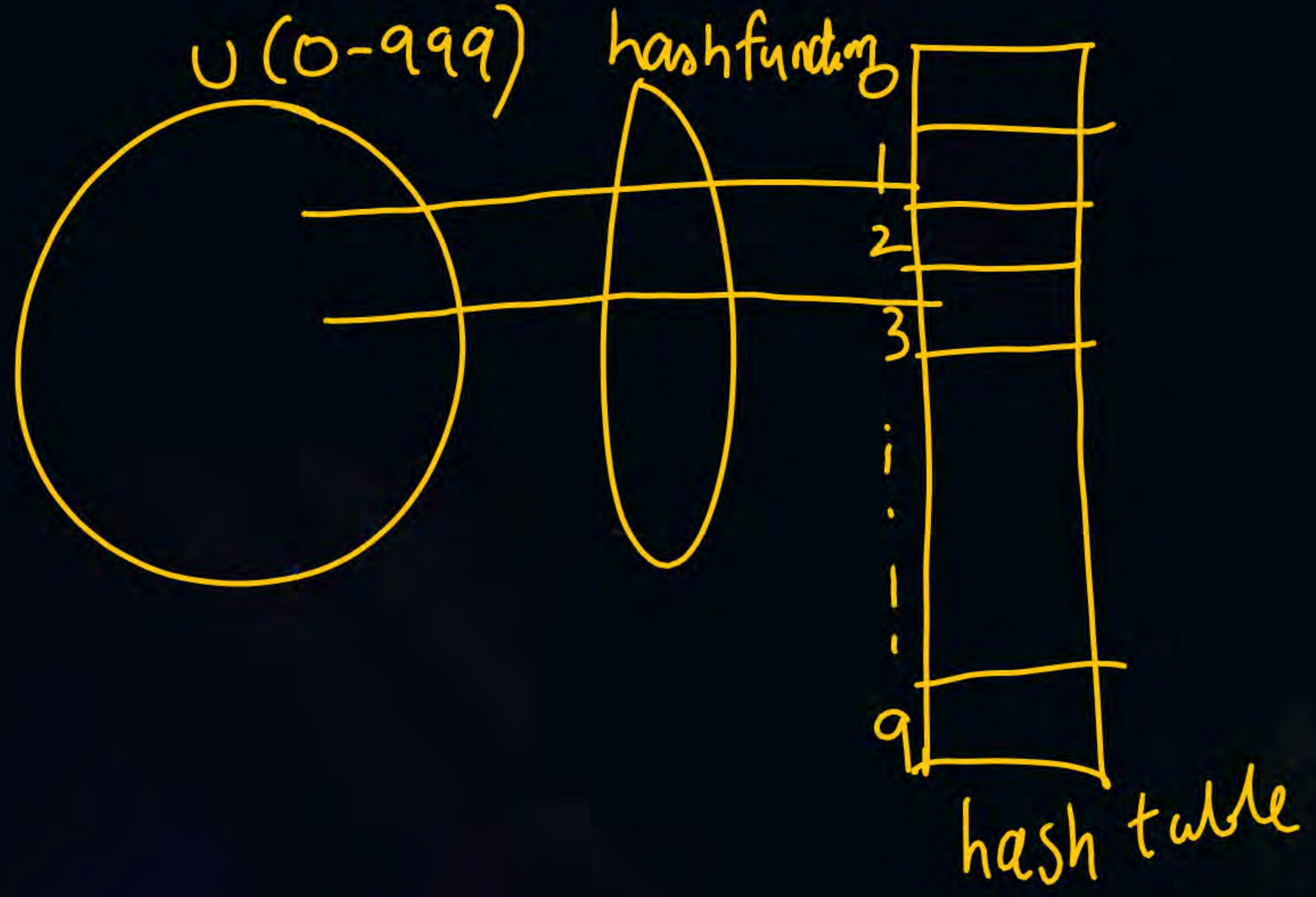
Before hash tables **Direct address table** was used

$\{0, 1, 2 \cdot ---- 99\}$



$\{1000000, 1000001, 1000002\}$

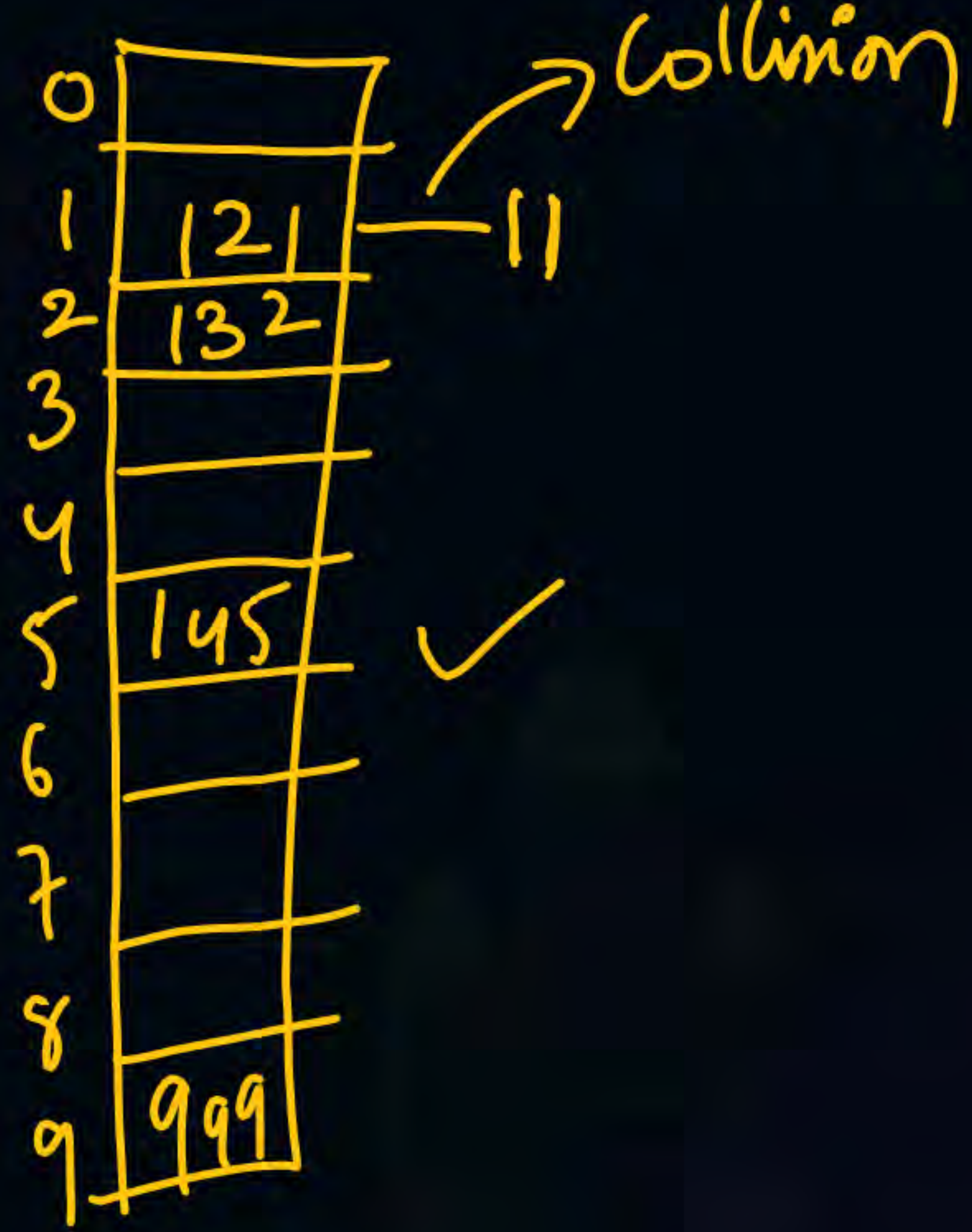



U (0-999)
hashfunction
1
2
3
9
hash table

Ex:- 121, 145, 132, 999, 11

$h(0-999) \longrightarrow (0-9)$

mod 10 - hash function



To deal with collision :-

① Better hash function — Impossible — Collision ≠ 0

Collision Resolution techniques

① chaining

any no of elements can be inserted

elements are present outside the table

2) open addressing

elements are present inside the table

The number of elements cannot be greater than size of the hash table

a) linear probing

b) quadratic probing

c) Double hashing

Chaining is better if we want to do Insert, delete and Search.

If we dont want to do delete, only insert and search open addressing is better

<u>Chaining :-</u>

$K \rightarrow$ insert

$T[h(k)]$

In worst case, all the elements may go to same buck in the same list

Search – $O(n)$ – worst case

average – $O(1)$

m buckets

n elements

$\frac{n}{m}$ elements

$TC = O(n/m)$

$n = Km$

$n/m = K$

Seach $TC = O(K) = O(1)$

Insertion $TC = O(1)$

Deletion $TC = O(1)$

average

Disadv of chaining → pointer

Gate: An advantage of chained Hash table over open addressing

a) worst case time complexity of search is less

b) space used is less

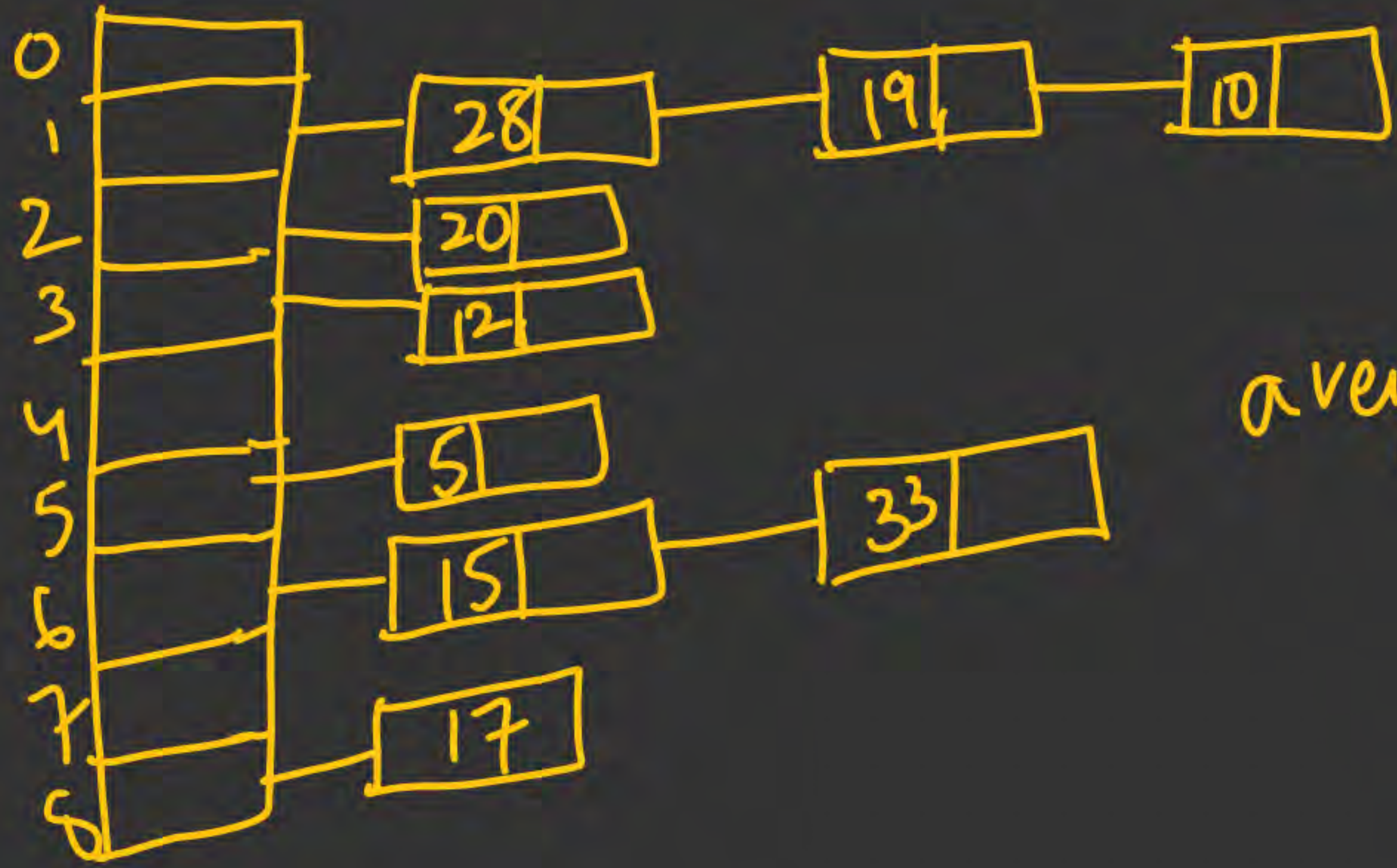
c) deletion is easier —

d) None of the above

Gate:- h(k) = k mod 9, hash table has 9 slots, chaining is used

Keys 5, 28, 19, 15, 20, 33, 12, 17 and 10

The maximum, minimum, average chain lengths



max = 3

min = 0

$$\text{average} = \frac{3+1+1+1+2+1}{9}$$

$$= \frac{9}{9} = 1$$

Gate:- Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 3 slots are unfilled after 3 insertions

$$\underset{\mathrm{I}}{\left(\frac{97}{100}\right)}\underset{\mathrm{II}}{\left(\frac{97}{100}\right)}\underset{\mathrm{III}}{\left(\frac{97}{100}\right)} \checkmark$$

Gate:- Consider a hash table with n buckets where chaining is used to resolve collisions. The hash function is such that the probability that a key value is hashed to a particular bucket is $1/n$. The hash table is initially empty and K distinct values are inserted in the table.

a) What is the probability that bucket '1' is empty after K insertions

$1/n$ $\qquad 1-1/n = \left(\frac{n-1}{n}\right)^K$

Gate:- Consider a hash table with n buckets where chaining is used to resolve collisions. The hash function is such that the probability that a key value is hashed to a particular bucket is $1/n$. The hash table is initially empty and K distinct values are inserted in the table.

b) what is the probability that no collisions has occurred in any of K insertion

$$\left(\frac{n}{n}\right)\left(\frac{n-1}{n}\right)\left(\frac{n-2}{n}\right)\cdots\cdots\left(\frac{n-(K-1)}{n}\right)$$

K^{th} element

Gate:- Consider a hash table with n buckets where chaining is used to resolve collisions. The hash function is such that the probability that a key value is hashed to a particular bucket is 1/n. The hash table is initially empty and K distinct values are inserted in the table.

c) What is the probability that first collision occurs after K^{th} insertion

$$\left(\frac{n}{n}\right)\left(\frac{n-1}{n}\right)\cdots\cdots\left(\frac{n-(K-2)}{n}\right)\left(\frac{K-1}{n}\right)$$

I II (K-1) K

open addressing

We are not going to store elements outside the table

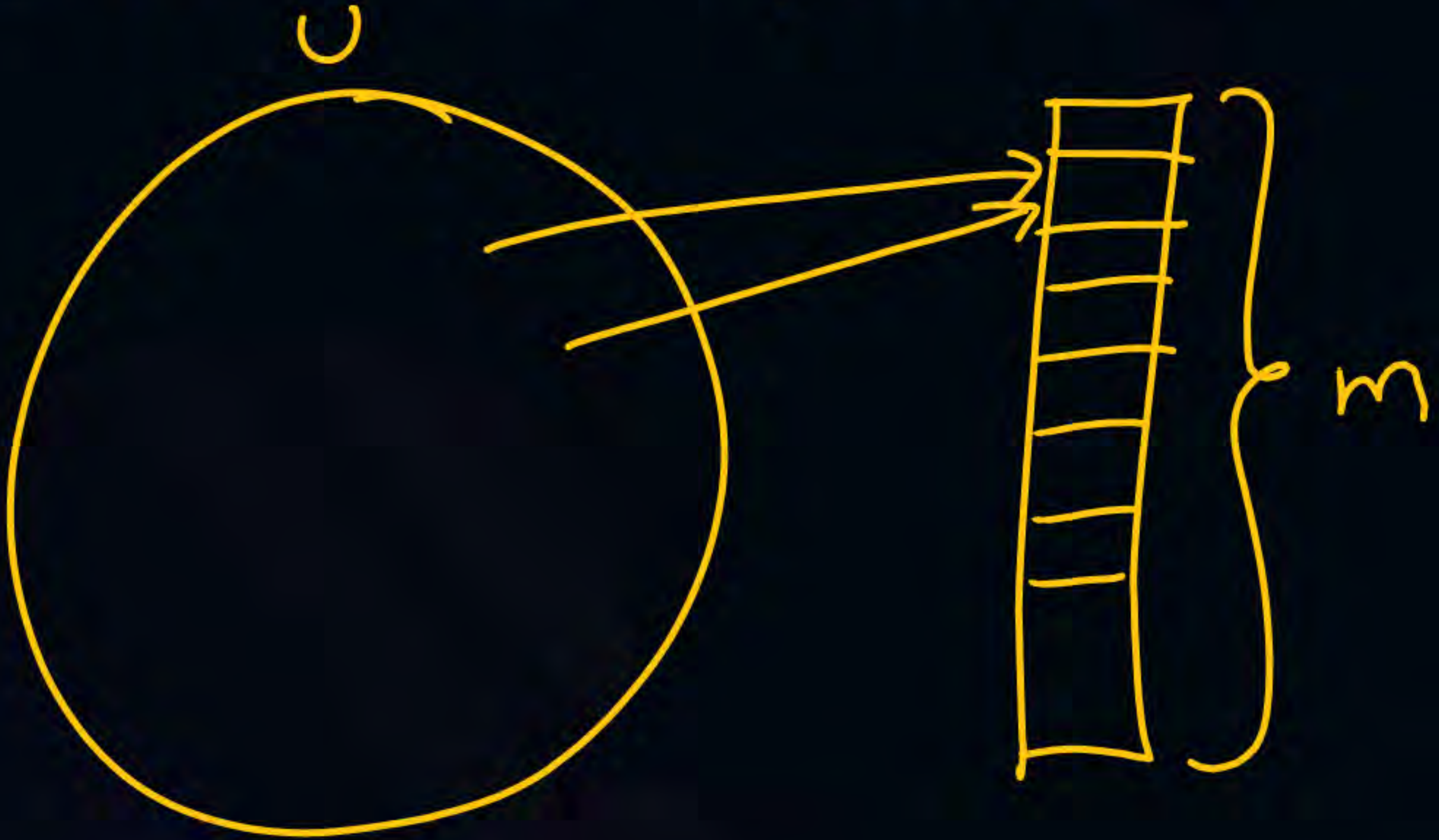
we can insert only a maximum of m elements

let no of element = n

$$n \le m \Rightarrow \frac{n}{m} \le 1$$

$$0 \le n/m \le 1$$

Collisions will happen, how to resolve the collisions



$$h\left(U \times \{0, 1, 2, 3 \ldots\ldots m-1\}\right) \rightarrow \left(\underline{\underline{0 - m}}\right)$$

k_1

$$h(k_1, 0) = 2$$
$$h(k_1, 1) = 6$$
$$h(k_1, 2) = 3$$

0	N
1	N
2	X
3	$\cancel{N}$ k_1
4	X
5	N
6	X
7	N

k_1

In worst case Seach = $O(n)$

$h(k_1, 0) = 2$ ✓
$h(k_1, 1) = 6$ ✓
$h(k_1, 2) = 3$ ✓
$h(k_1, 3) = 4$ ✓
$h(k_1, 4) = 5$ ✓
$h(k_1, 5) = 1$ ✓
$h(k_1, 6) = 7$ ✓
$h(k_1, 7) = 0$ ✓

0	N	
1	N	
2	~~a~~	D.
3	~~c~~ ~~N~~	✓ D ✓
4	~~d~~	✓ D ✓
5	~~N~~ k_1	✓
6	~~b~~	✓ D.
7	K	

Search for k_1

adv of open addressing
— No chains

linear probing:-

$$h : U \to \{0 \ldots\ldots m-1\}$$

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

$$h'(k, 0) = h(k) \bmod m = a$$

$$h'(k, 1) = (h(k) + 1) \bmod m$$

$$= a + 1$$

$$h'(k, 2) = (\underset{a}{h(k)} + 2) \bmod m$$

Disadvantage of linear probing

- Secondary clustering ✓

$(0, 1, 2, 3, \ldots \ldots m-1)$

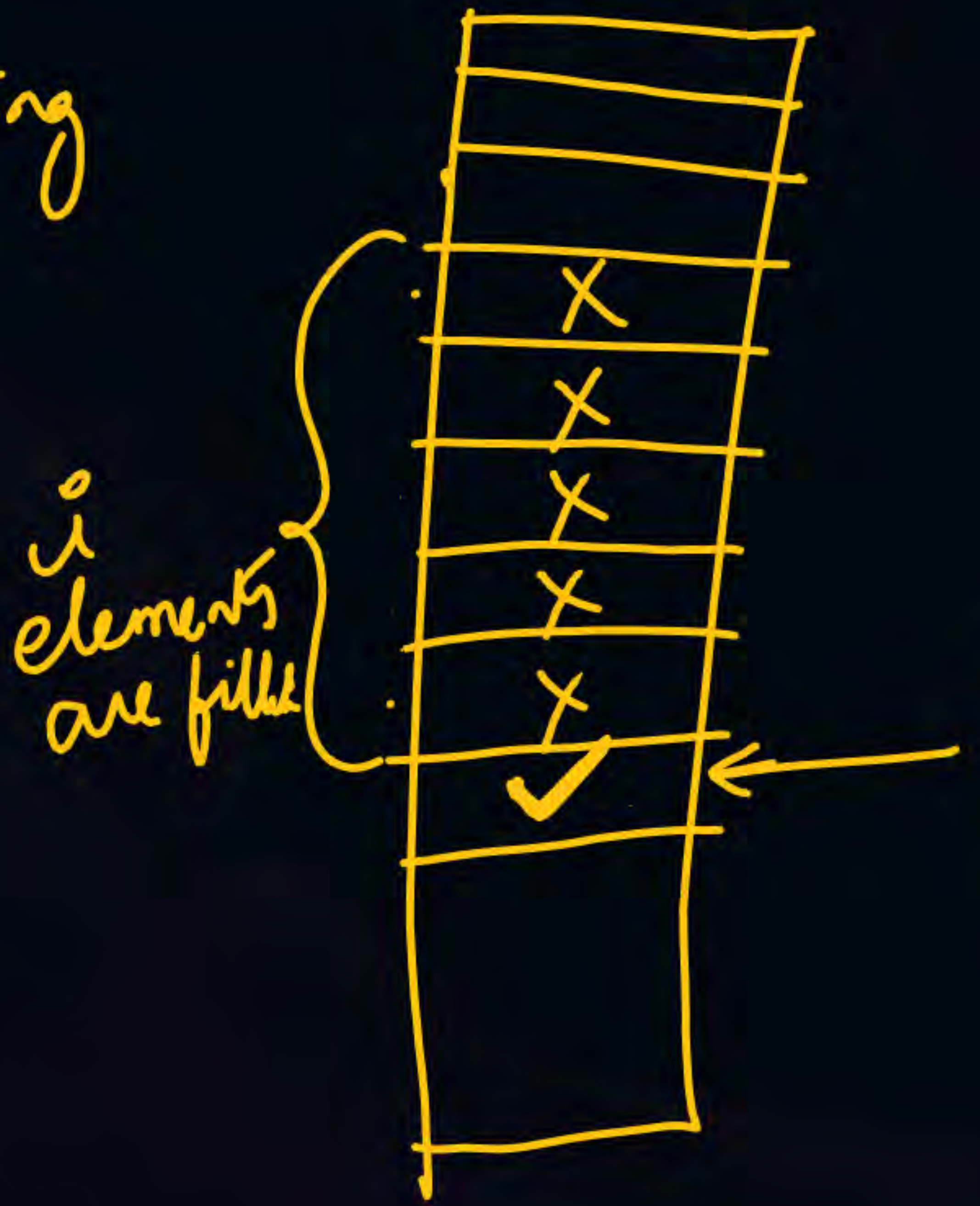
$(1, 2, 3, \ldots \ldots m-1, 0)$

$(2, 3, 4, 5 \ldots \ldots m-1, 0, 1)$

no of probe sequences $= m$

Disadvantage of linear probing:-

Primary clustering



$h(k)$

$$\left(\frac{i+1}{m}\right) \checkmark$$

$K \rightarrow$ Search

$h(K) = a$

a

Worst case TC = $\underline{O(m)}$

average case TC = $\underline{O(1)}$

Gate:- Keys = 12, 18, 13, 2, 3, 23, 5 and 15

$h(k) = k \bmod 10$ and linear probing is used

What is the resultant hash table if size of HT is 10

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

$$\frac{8+1}{10} = \frac{9}{10}$$

$$\frac{1}{10}$$

→ **Gate:-** Hash table size = 11 (0-10)

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

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

what is the index into which last record is inserted using linear probing.

0	87
1	11
2	13
3	36
4	92
5	4
6	71
7	14
8	
9	
10	43

7

$\frac{8+1}{11}$ $\frac{9}{11}$ ✓

$1/11$

Gate:- Consider a hash function that distributes Keys uniformly. The hash table size is 20. After hashing how many elements will the probability that any new key hashed collides with an existing one **exceeds 0.5**

a) 5 b) 6 c) 7 d) 10 ✓

$$\frac{i}{m} > 0.5$$

$$\frac{i}{20} > 0.5 \Rightarrow i > \underline{10} \checkmark$$

Gate:- $h(k) = k \bmod 10$, linear probing

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

Quadratic probing :-

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

$h'(k, i) = (h(k) + \underline{c_1 i + c_2 i^2}) \bmod m \rightarrow$ quadratic pm

No primary clustering Secondary clustering

$(1, 4, 6, 8, 10 \ldots -)$

$$h'(k,i) = (h(k) + \underline{\underline{c_1}}\, i + \underline{\underline{c_2}}\, i^2) \bmod \underline{\underline{m}}$$ ✓

we cannot take random values for c_1, c_2, m

$m = 10$

$(0, 1, 2 \ldots \ldots 9)$

$c_1 = 1 \quad c_2 = 1$

$h(k_1) = 1$

$h(k,1) = 1+1+1 = 3$

$h(k,2) = 1+2+4 = 7$

$h(k,3) = (1+3+9) \bmod 10 = 3$

<u>Disad</u>:- Secondary clustering

c_1, c_2, m

$(m) \to$ no of probe sequence ✓

$(0, 1, 4, 6, 7, 8 \ldots \ldots)$ ✓

Double hashing :-

$$h'(k, i) = (h_1(k) + i\, h_2(k)) \bmod m$$

m^2 probe sequences

Primary clustering ×

Secondary clustering ×

(0, 2, 4, 7, 8, ... — ...)

(0, 1, 3, 8, ... —)

Seach

worst case $O(n)$

average case $O(1)$

THANK - YOU