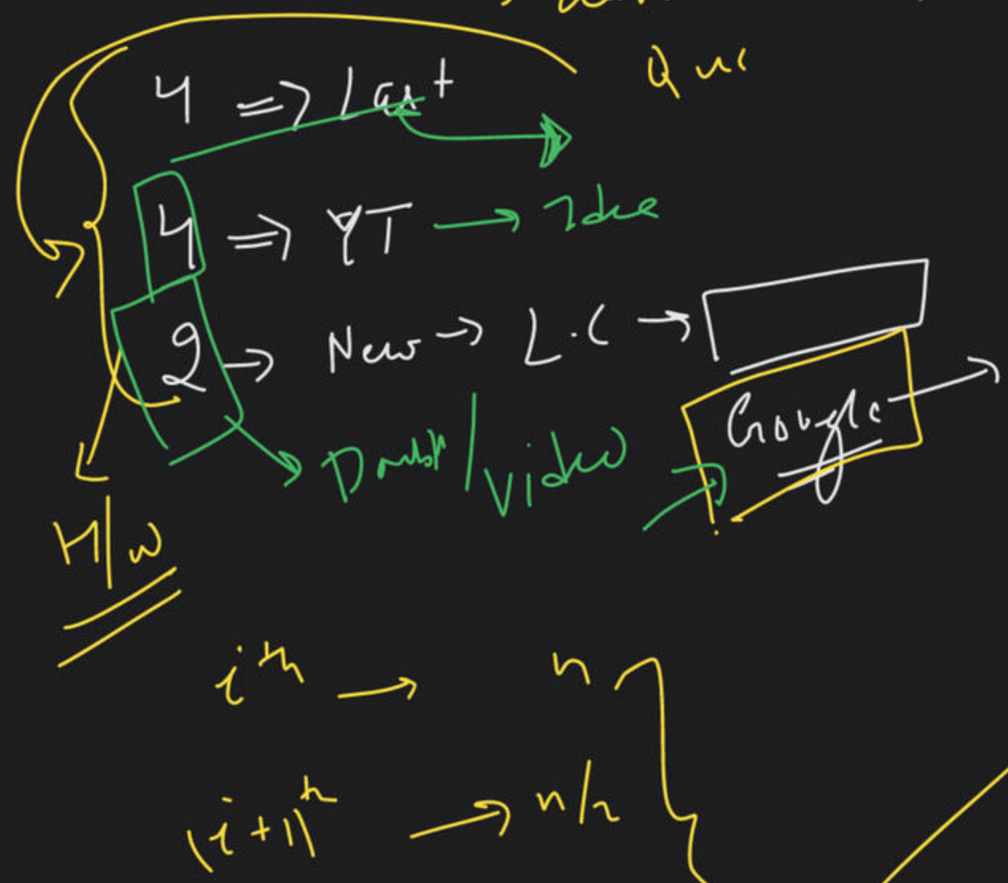
Binary Search Problems - I

Foundation Course on Data Structures & Algorithm - Part I

Love Babbar • Lesson 11 • May 5, 2022

→ Binary Search → LO Q (H/W) → Yes or No

→ classical ID



find an element in a sorted array of unknown size

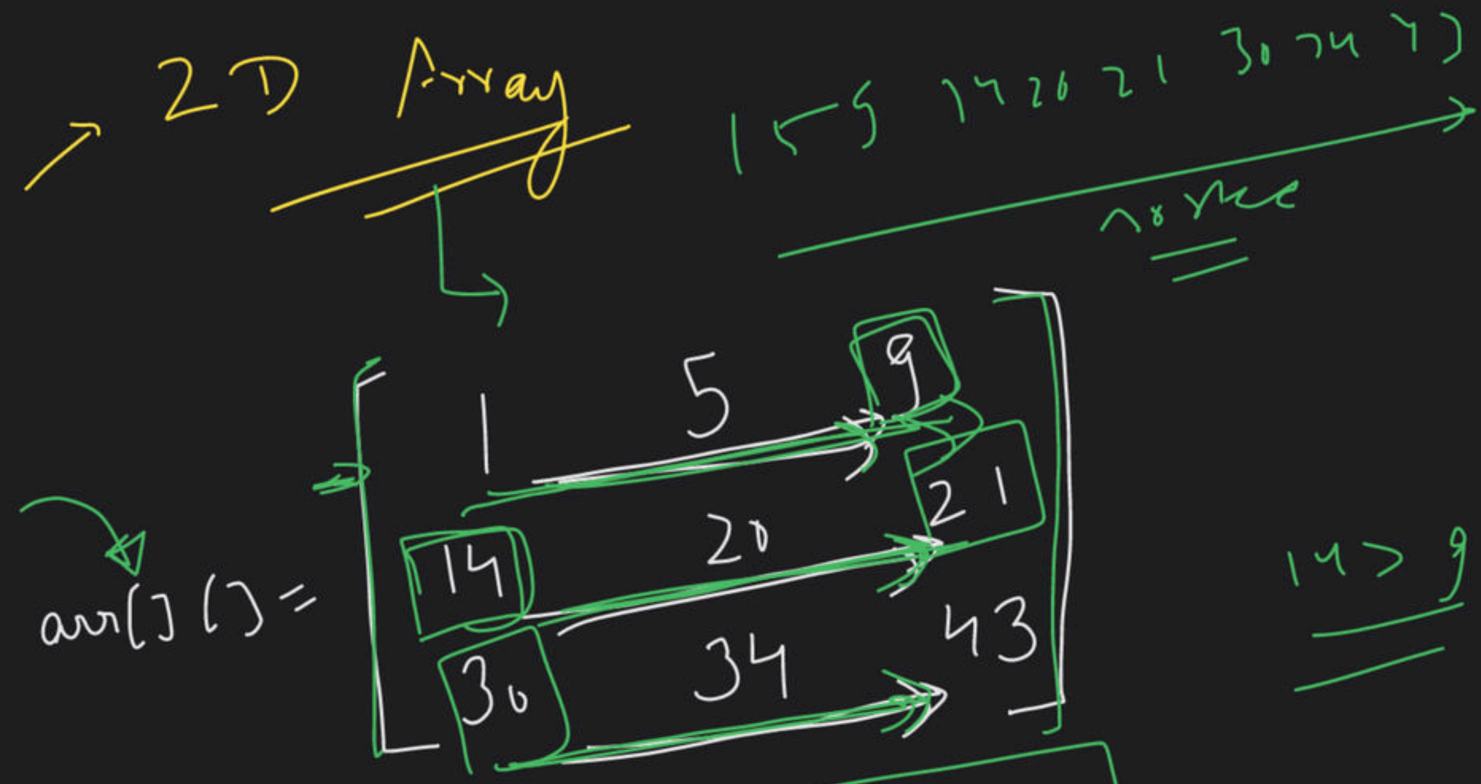
i^{th} → n

$(i+1)^{th}$ → n/2

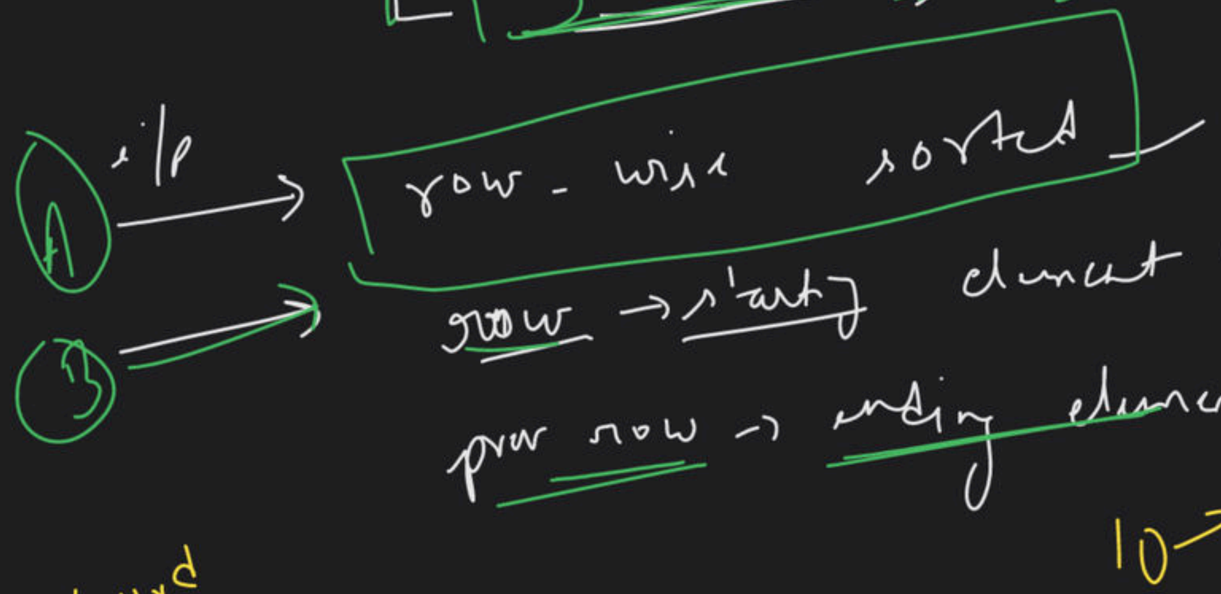
$(i+2)^{th}$ → n/4

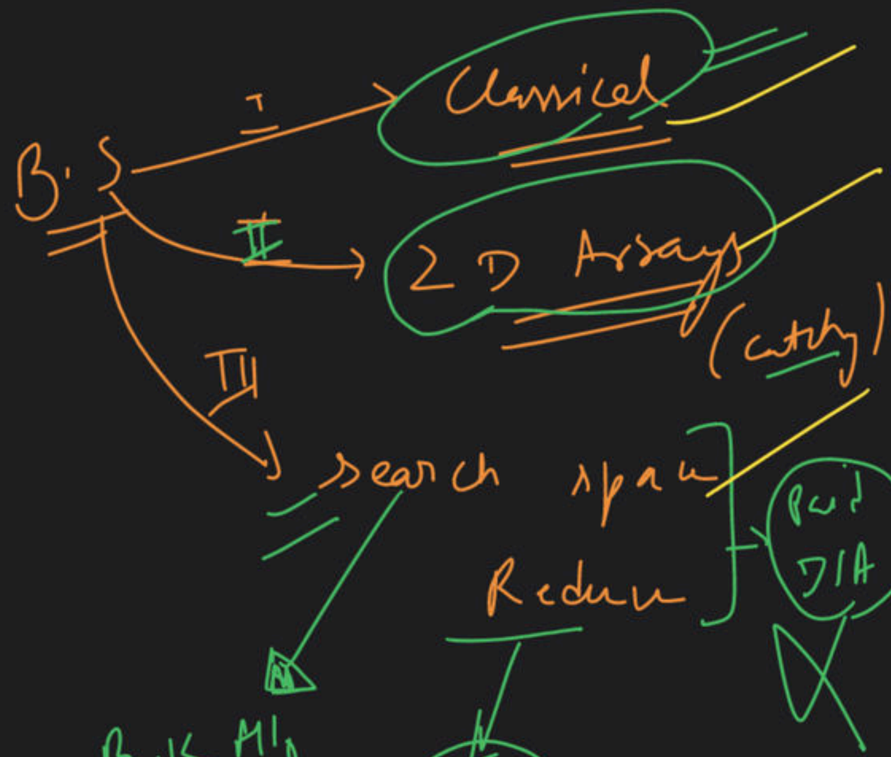
Yes/No

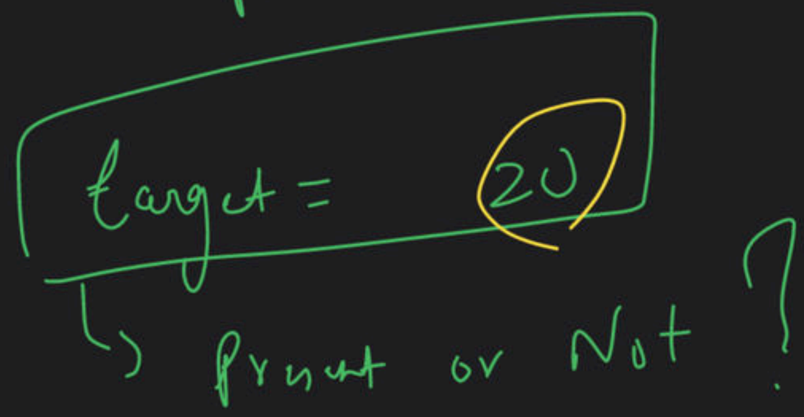
2D Array


1 5 9
14 20 21
30 34 43
arr[][] =
1 5 9 14 20 21 30 34 43
Sorted

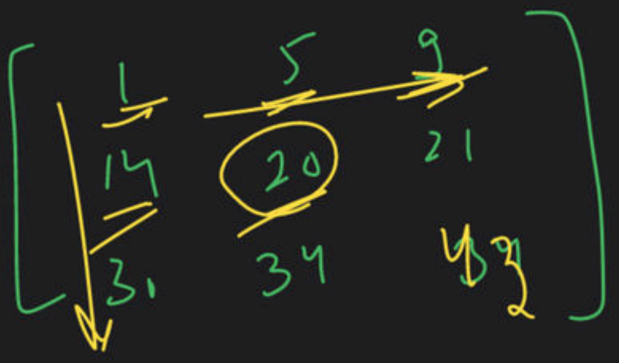
14 > 9


i/p
row - wise sorted
row → starting element
prev row → ending element
10 → 100
Wind


B.S
I
Classical
II
2D Arrays
(catchy)
III
Search space
Reduce
Book A/c
Path LOW
pair for
CP
Real Life


target = 20
Present or Not ?

#1



target = 21

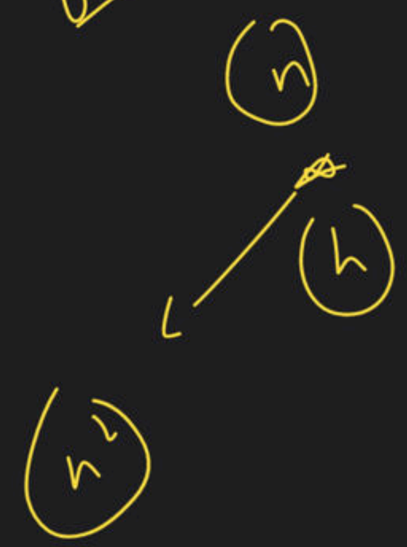
Comfortable

Klothes

Noob

$T.C \rightarrow O(n^2)$

Brute force




```
for (int i=0 ; i<n ; i++)
{
    cout << "i" << i << endl
    for (int j=0 ; j<n ; j++)
    {
        cout << "j" << j << endl
        if (arr[i][j] == target)
        {
            cout << i << " " << j;
            return;
        }
    }
}
```

visualisation

i = 0
j = 0, 1, 2
i = 1
j = ...

row-wise sorted

B-H r S E

pR EN >

→ Binary Search

target = 20

m cols → log m

n rows

1	5	9
14	20	23
30	34	43

$O(\log n)$

T.C → $O(\log n + \log m)$

Algo:-

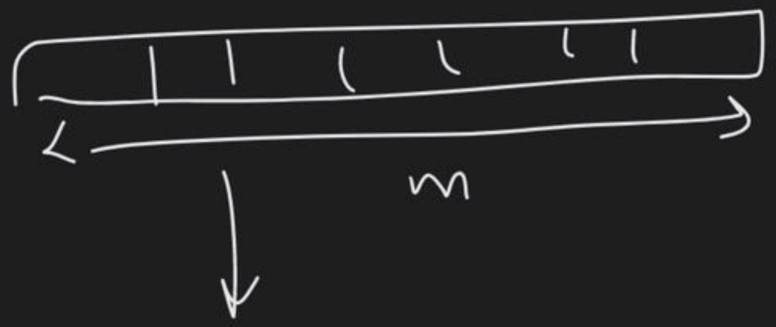
I → first find row in which number may be present

II → if found, apply Binary Search on that row to find target

rows - n
↓
B.S
↓
$O(\log n)$ - clear

row =

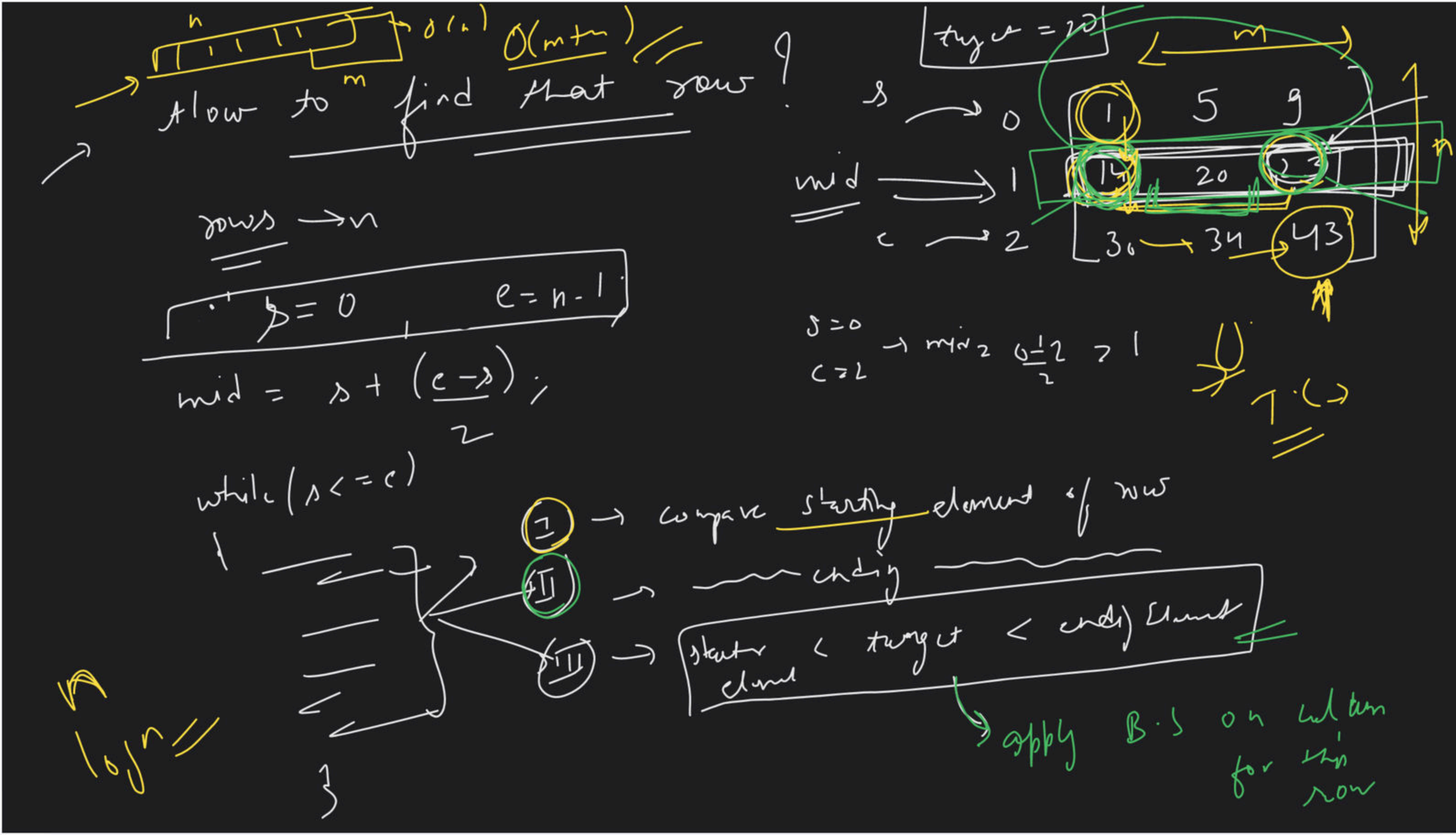
m cols

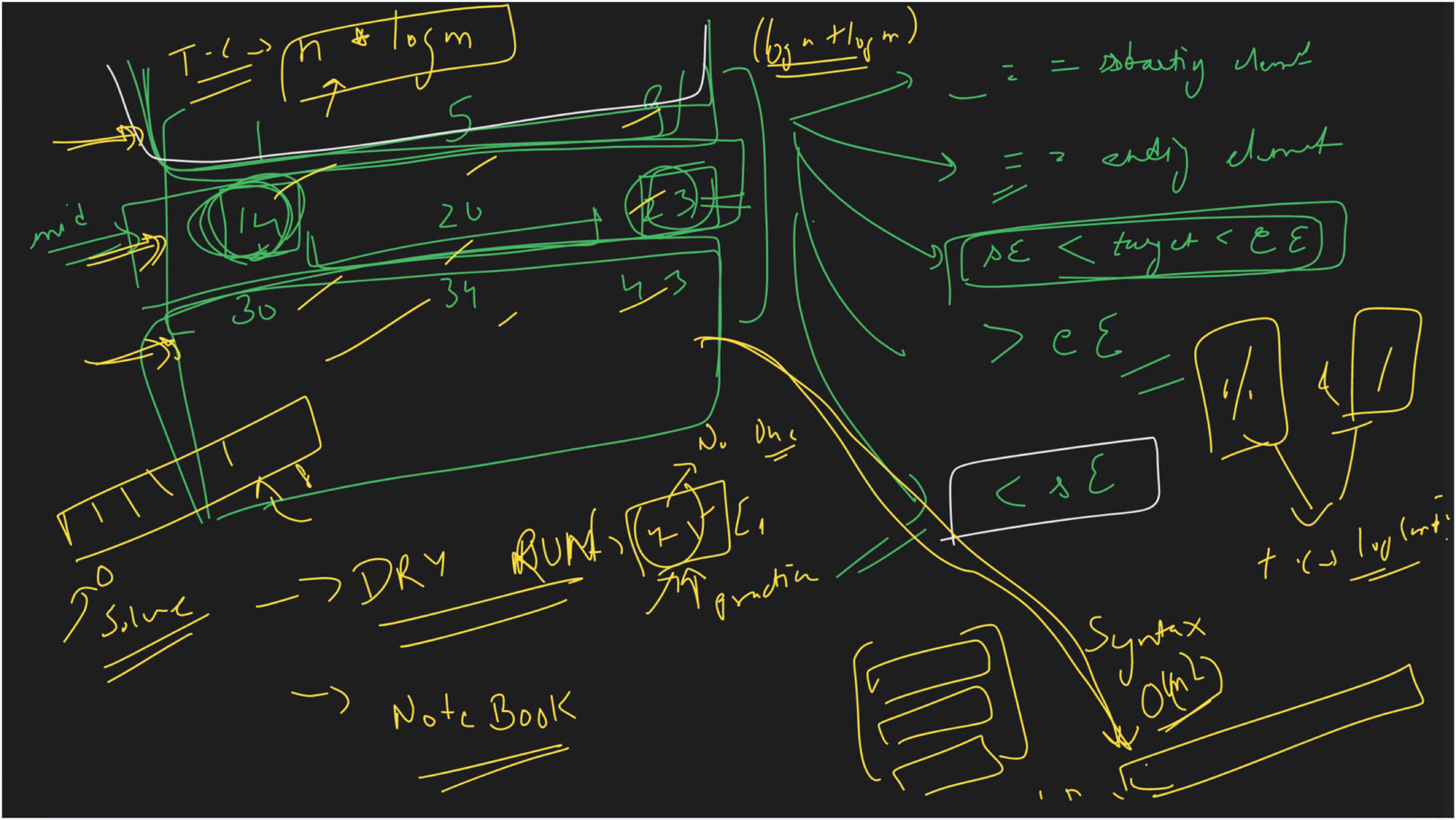


B.S

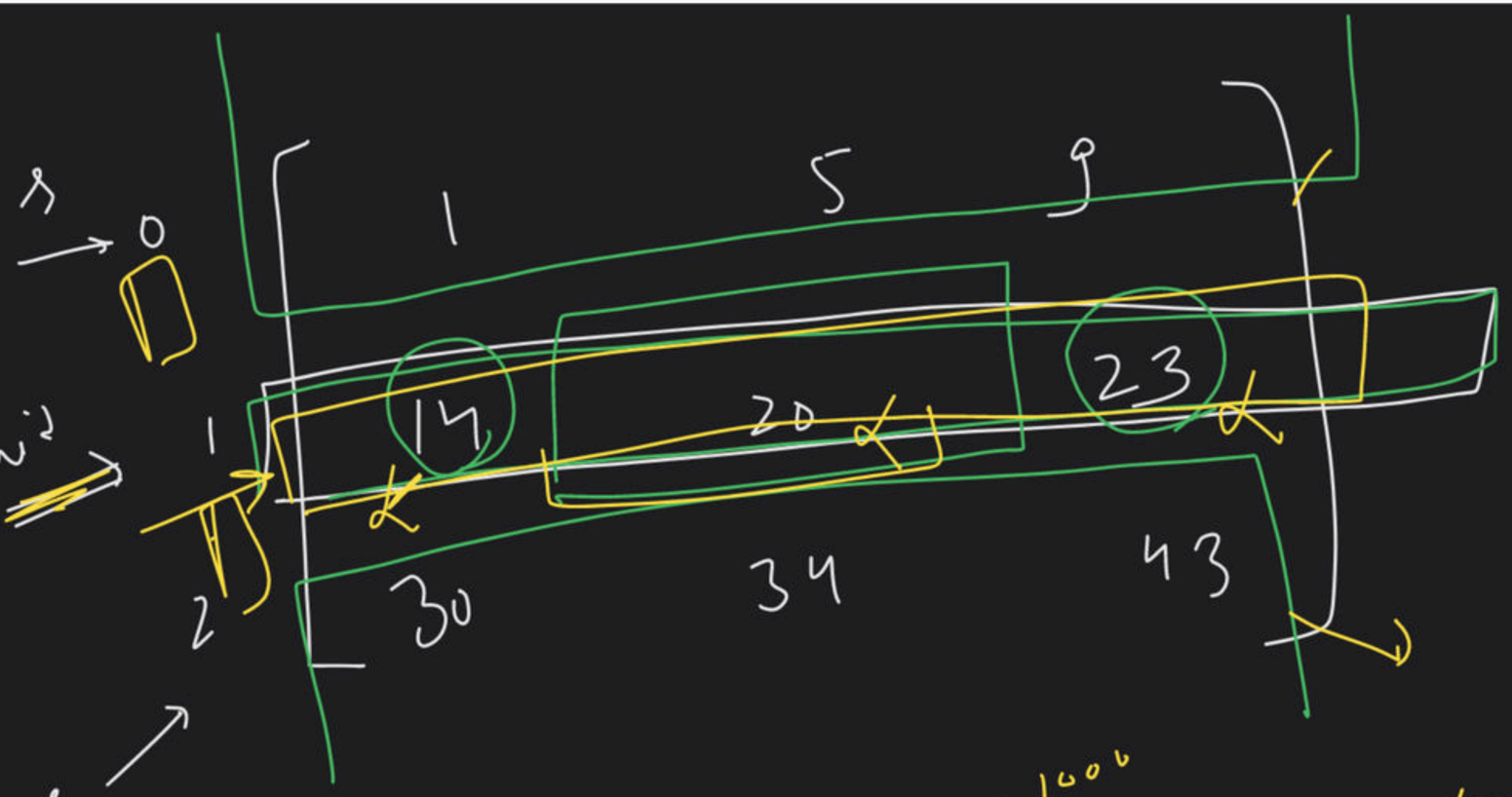
T.C -> logm

O(log n + log m)


n
m
O(n)
O(m+n)
How to find that row ?
rows → n
s = 0
e = n-1
mid = s + (e-s)/2;
while (s <= e)
target = 20
m
s → 0
mid → 1
e → 2
1
5
9
14
20
23
30
34
43
s=0
e=2
mid = 0+2/2 = 1
T.C
I → compare starting element of row
II → ending
III → start < target < end element
apply B.S on that for this row
log n



T.C -> n * log m
(log n + log m)
mid
1
5
9
14
20
23
30
34
43
== starting element
== ending element
SE < target < EE
> EE
< SE
No One
practice
Solve
-> DRY RUN
-> NoteBook
Syntax
O(n^2)



target = 30

Python

(I) → Upper Part

(II) → Lower Part

(II) → Row K ander

(I) → SE ✗

(II) → EE ✗

(III) SE <= TARGET ✓

$n = 2^{100}$ $m = 2^{1000}$

$\log(n) + \log(m) = 1100$

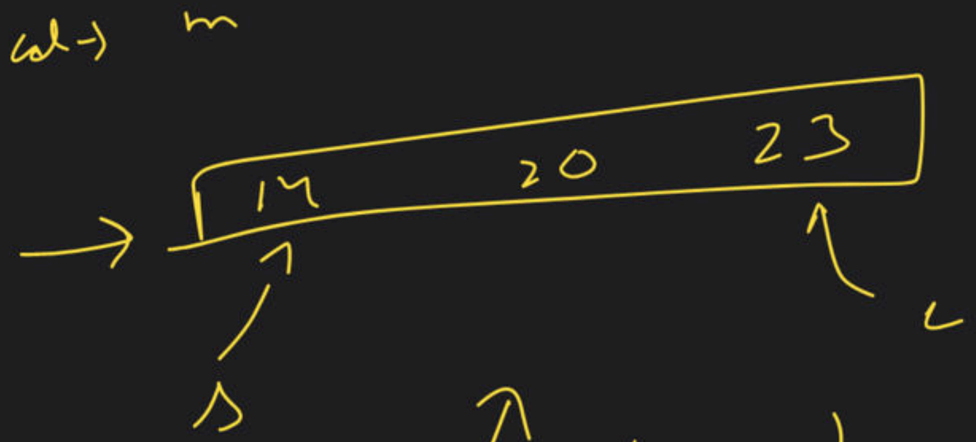
$\log(n \cdot m) \le 2 \cdot 2^{1100} = 1100$

0, 1

YT-120

$\log(n \cdot m) = \log n + \log m$

YT



B.S → O(log m)

(1) now - find out

s = 0 & e = n-1 = 3-1 = 2

e = 2

mid = $\left(\frac{0+2}{2}\right)_2$ = 1

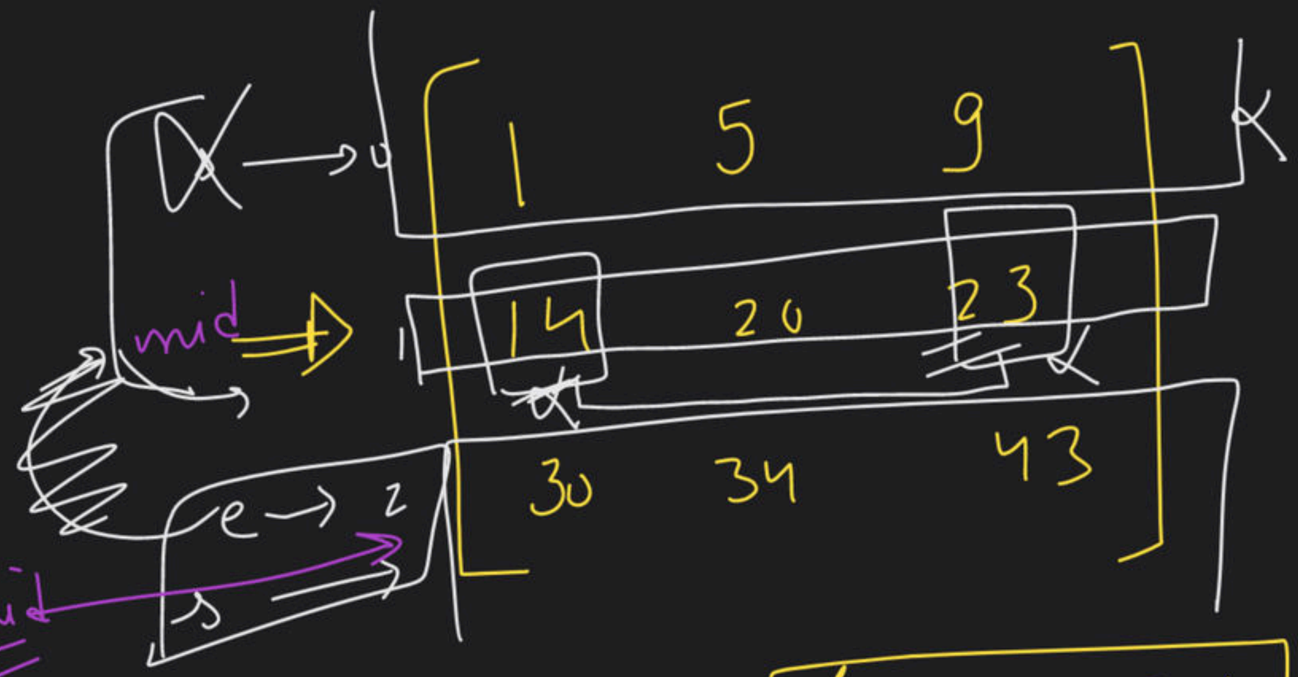
→ 14 == 34 → F

→ 23 == 34 → F

→ 14 < 34 < 23 → F

Up → 34 < 14 → F

Down → 34 > 23 → TRUE



target = 34

Up → e = mid - 1

Down → s = mid + 1

s = 0 = 1+1 = 2

s = 2 → mid = $\left(\frac{s+e}{2}\right)$

e = 2

= $\frac{2+2}{2}$ = 2

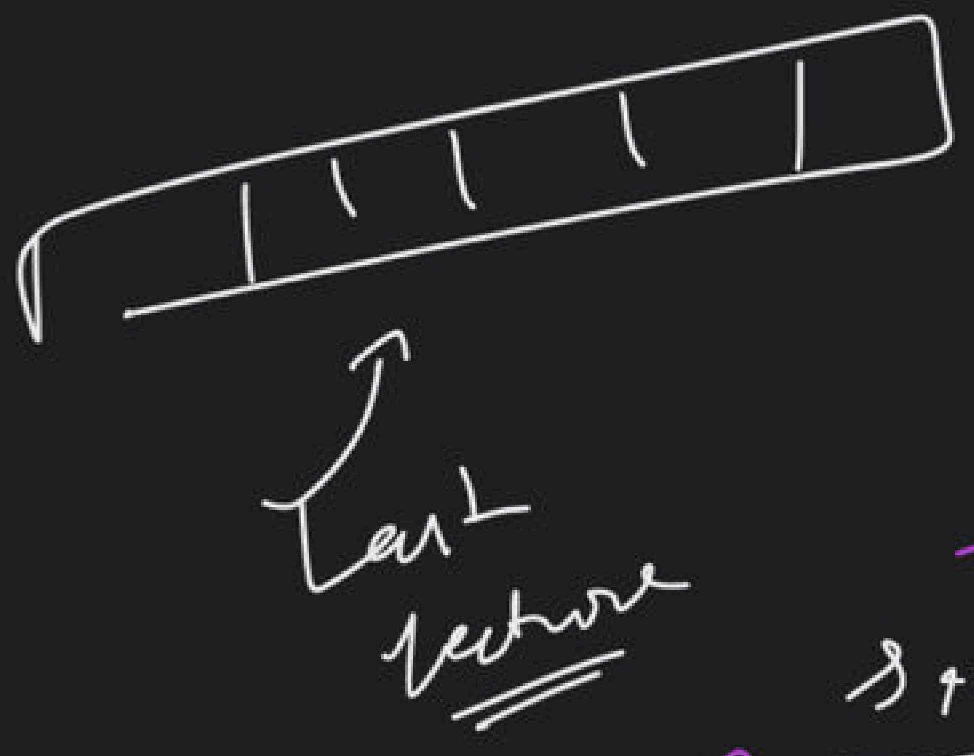
30 == 34 → F

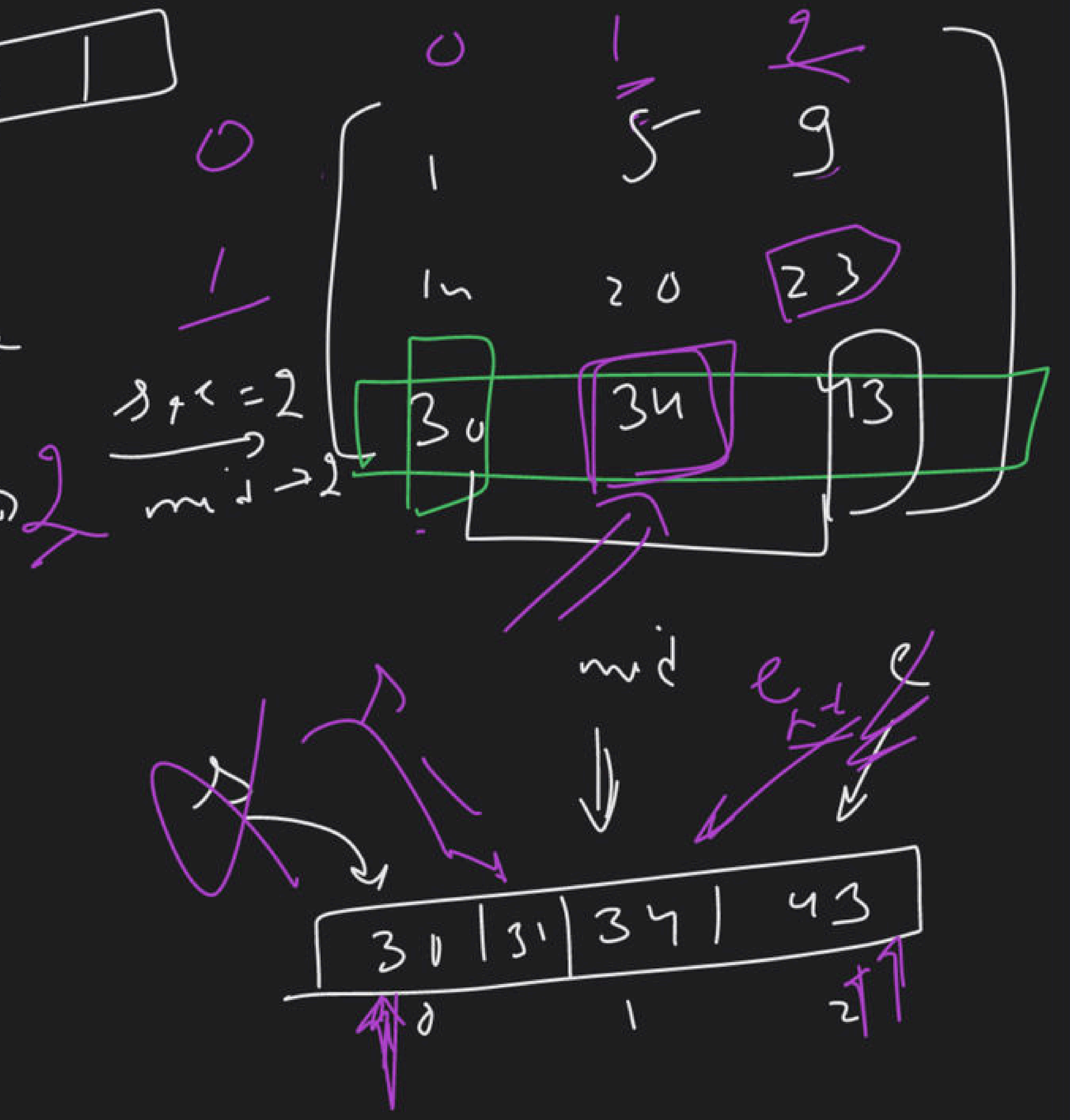
43 == 34 → F

30 < 34 < 43 → TRUE

T T

row = 2





(B) s = 0, e = 2

mid = 1

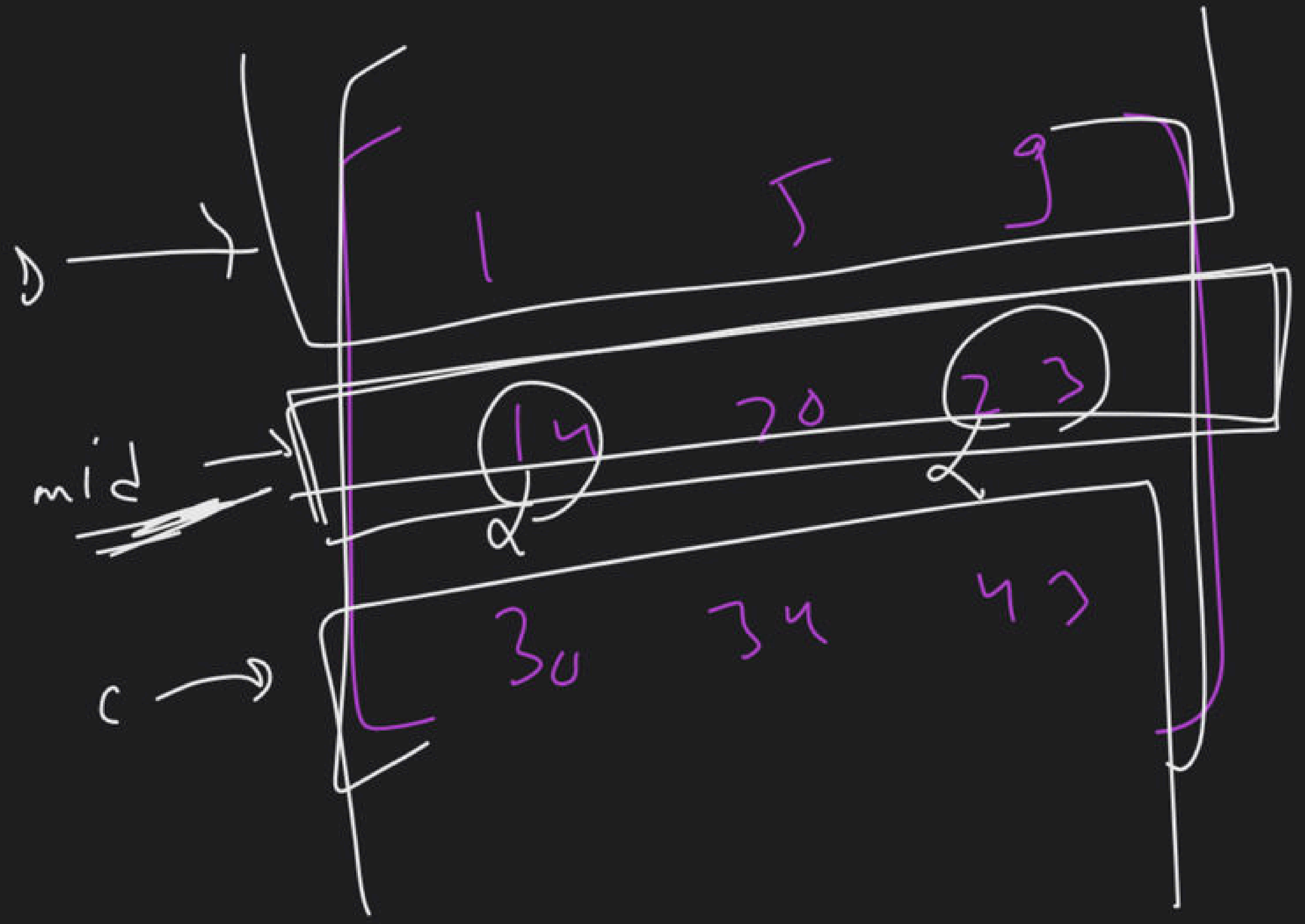
34 == 34 → TRUE → Return TRUE

14 <= Target <= 23

(I) → B.S

(II) → UP

(III) Down



mainc #

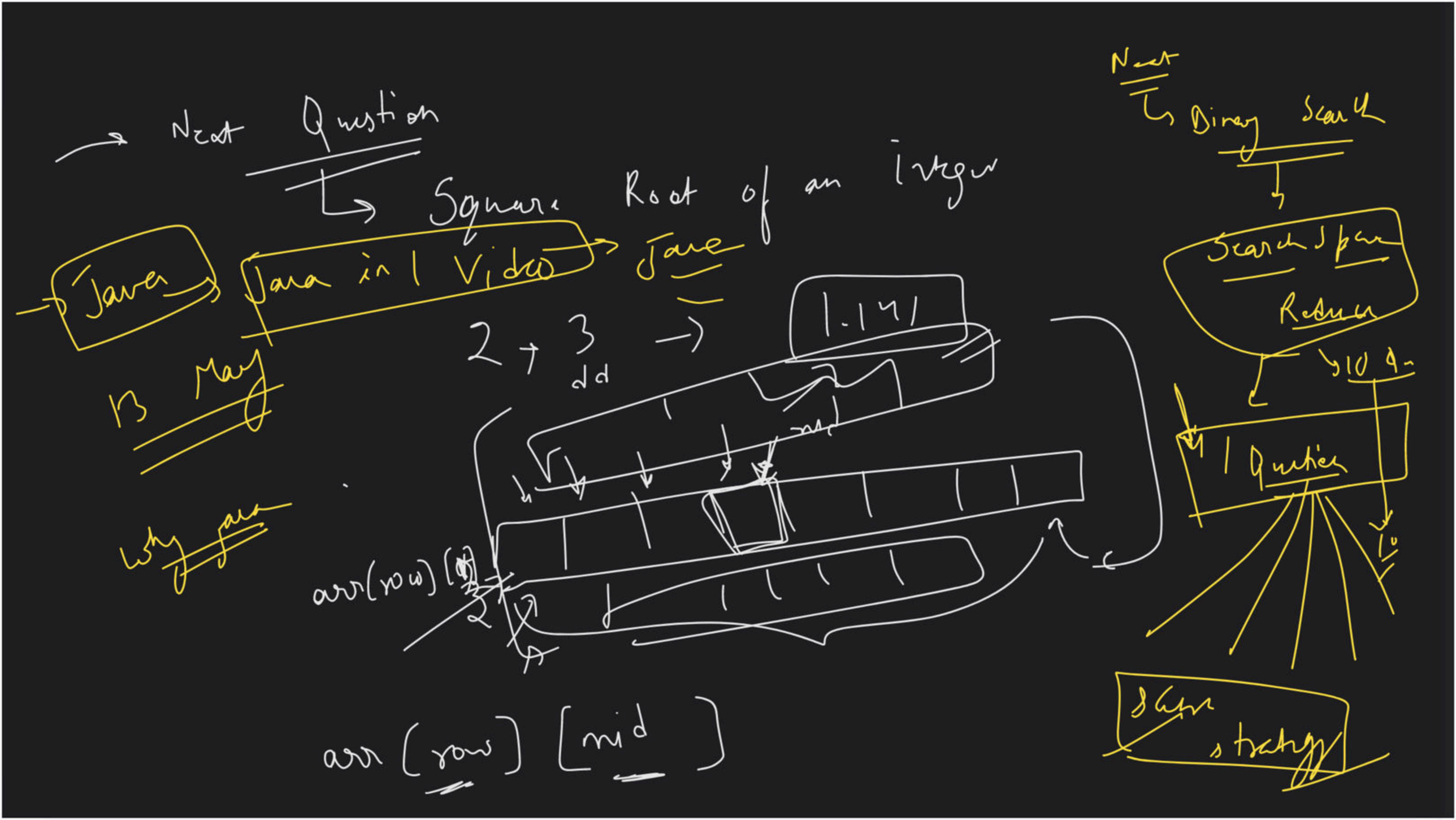
% 4 1 → soln → YT

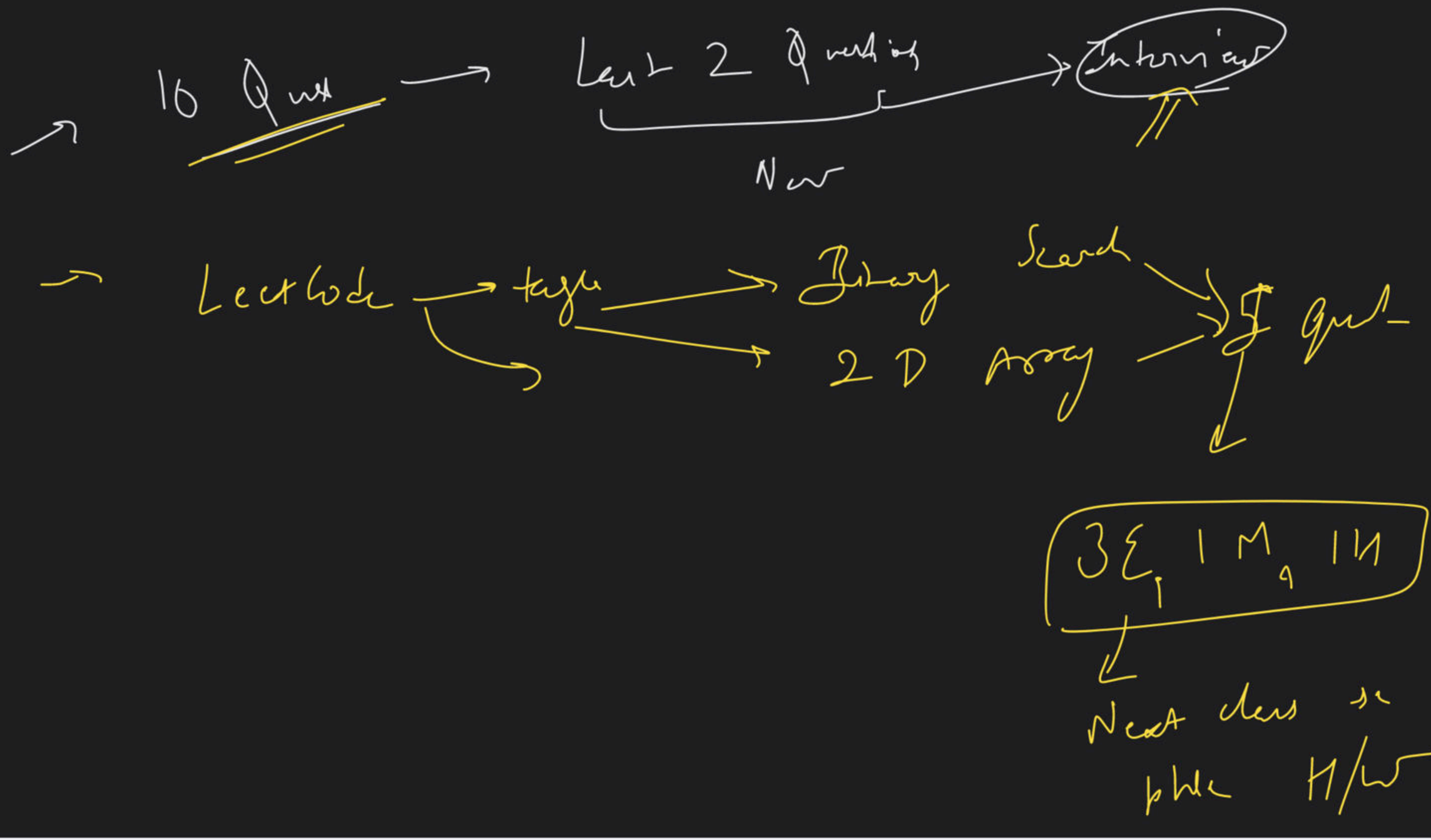
Aune #

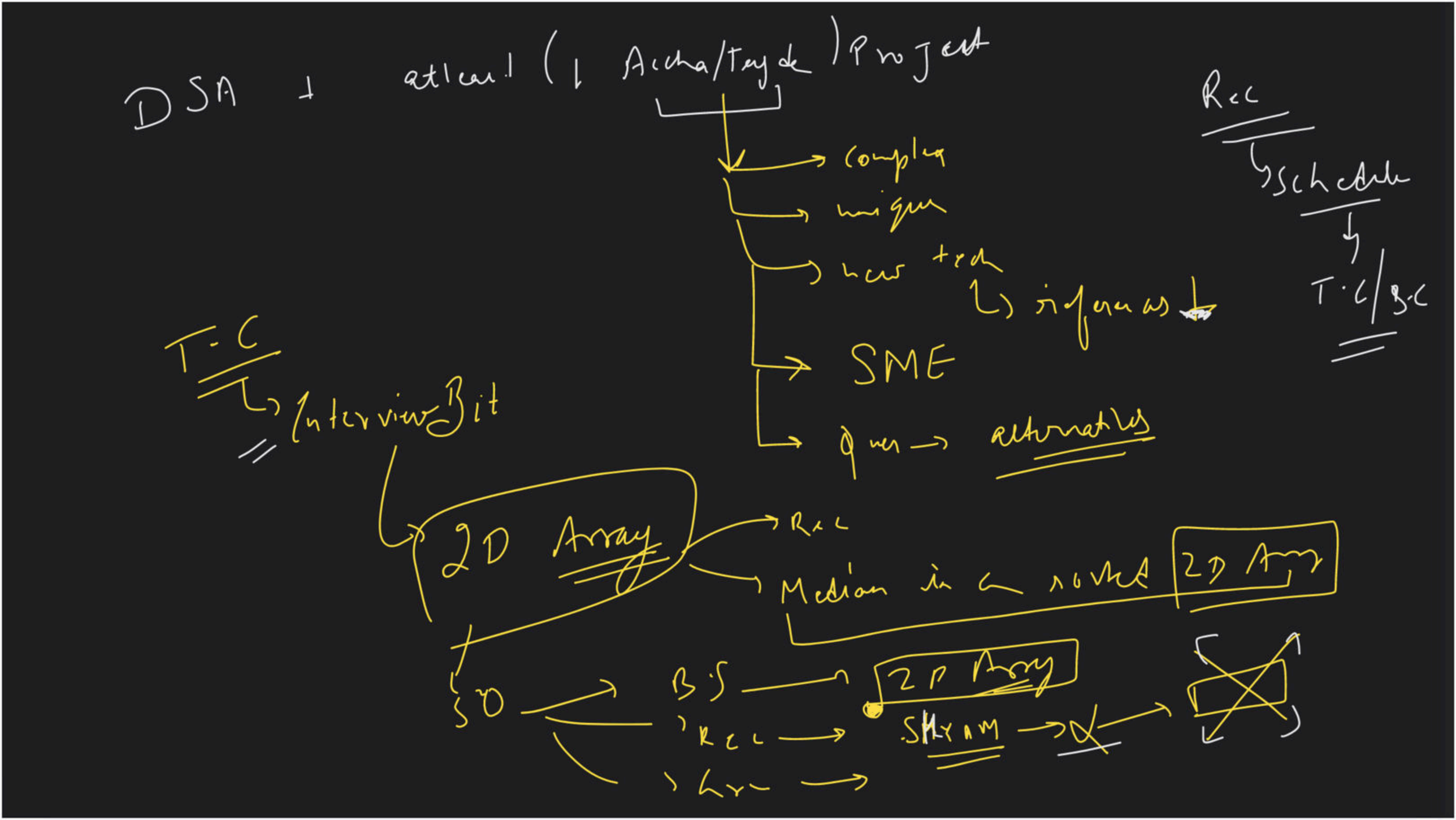
Final Year Roadmap

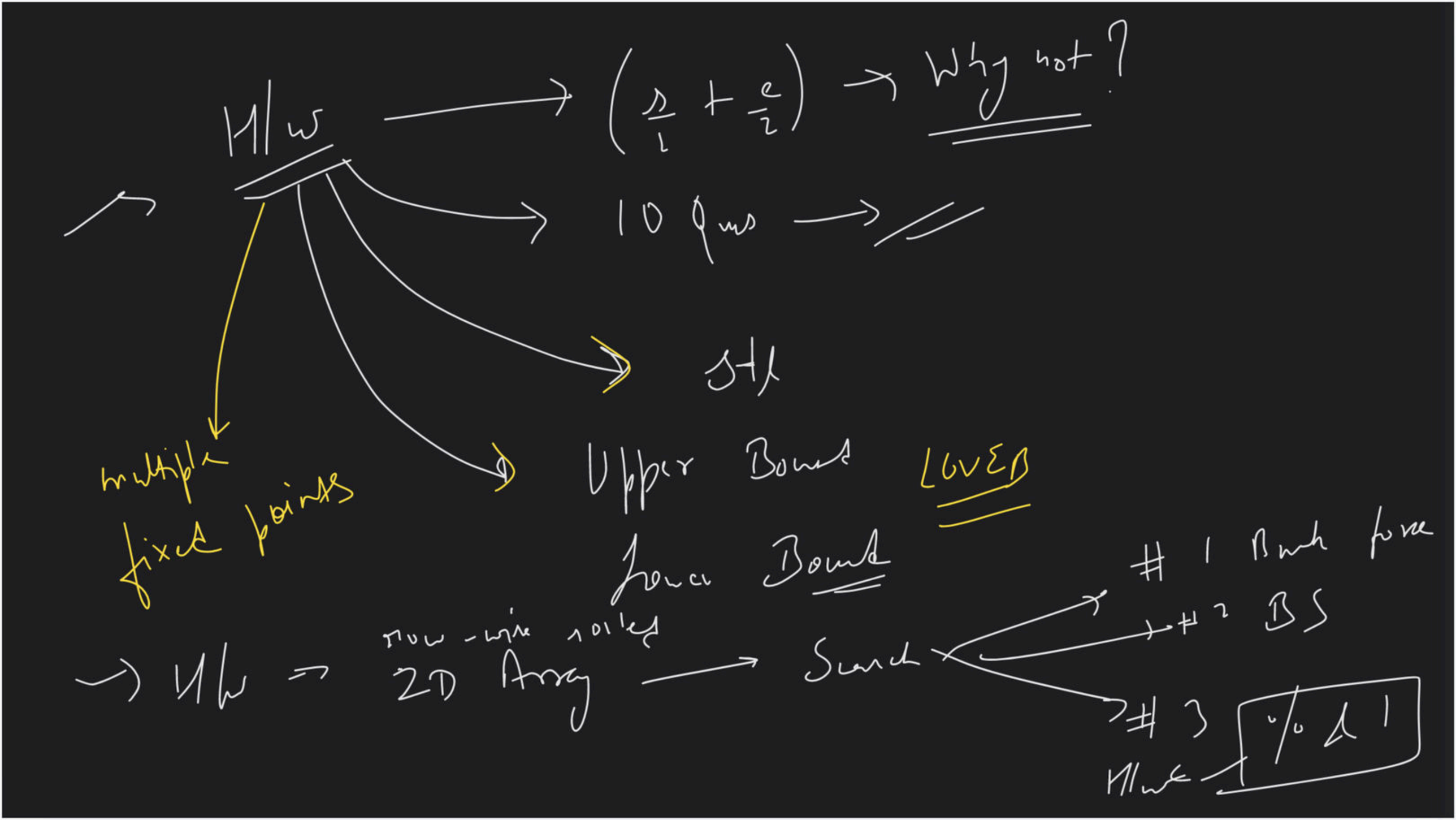
15 min

Queries


Next Question
Square Root of an Integer
Java
Java in 1 Video
13 May
Why Java
2 + 3
add
1.171
arr[row][mid]
arr [row] [mid]
Next
Binary Search
Search Space
Return
Mid
Question
Java
strategy


10 Ques
Least 2 Questions
Interview
Now
LeetCode
topic
Binary
2D Array
Search
3 E, 1 M, 1 H
Need class 12 pdf

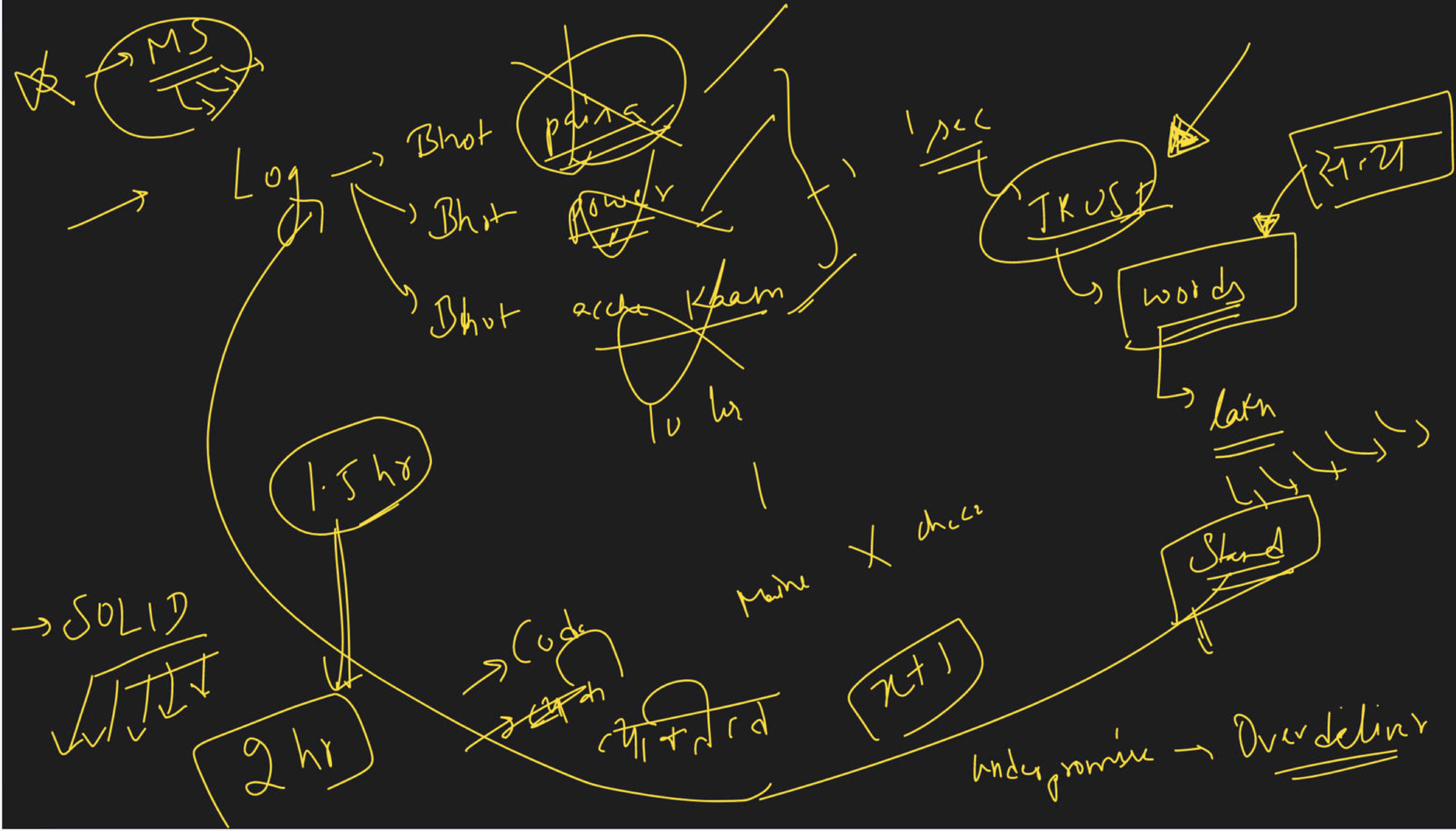

DSA + attend (1 Accha/Tayde) Project
complex
unique
new tech
references
SME
ques → alternatives
Rec
Schedule
T.C/B.C
T.C
InterviewBit
2D Array
Rec
Median in a sorted 2D Array
2D Array
Rec
Shyam


H/w
$(\frac{s}{2} + \frac{e}{2})$
Why not?
10 ms
s+1
multiple fixed points
Upper Bound
LOWER
Lower Bound
H/w
row-wise sorted 2D Array
Search
#1 Brute force
#2 BS
#3
Time
% 1

→ DBMS → 1 May

15 May 30 May

यहाँ से


MS
Log
Bhot pains
Bhot power
Bhut acche Kaam
10 hr
1 sec
TRUST
words
Stand
1.5 hr
2 hr
SOLID
Code
n+1
Underpromise → Overdeliver