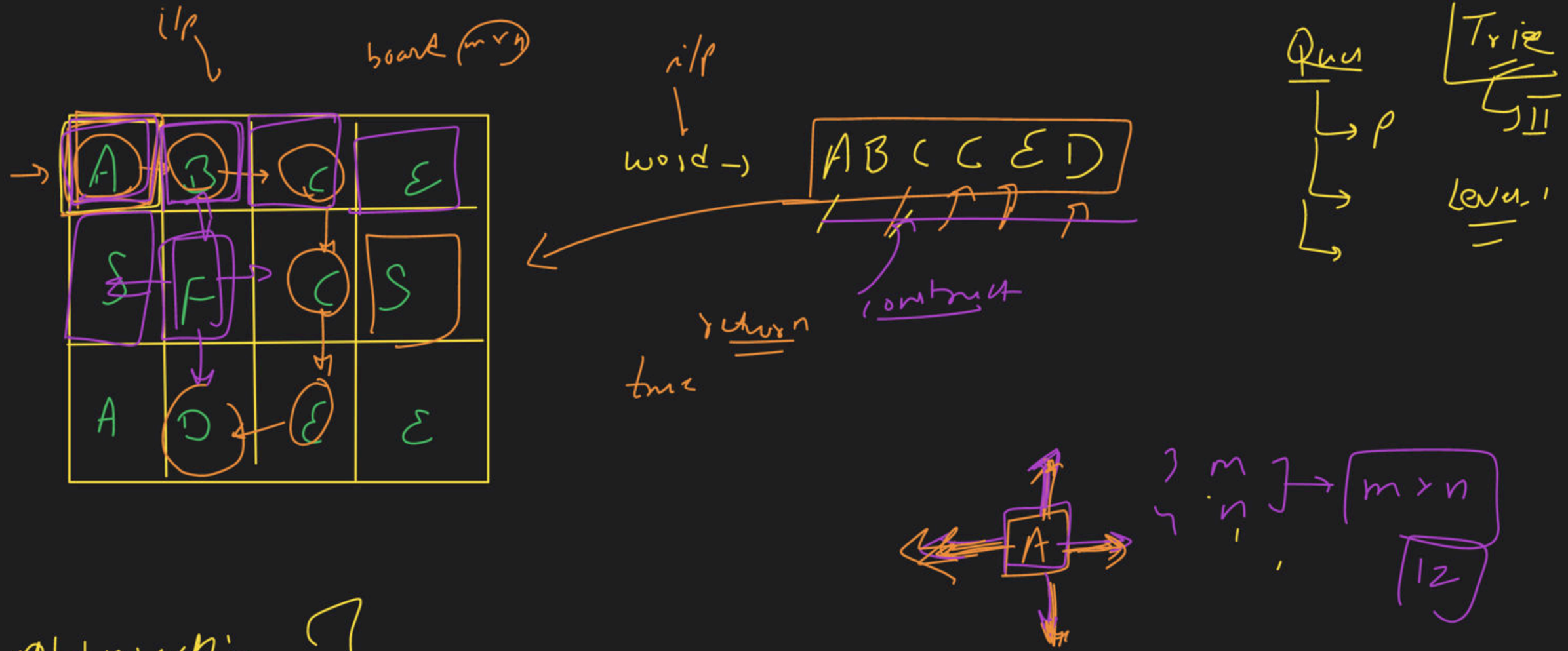
Backtracking - III & Time/Space Complexity of Recursive Algorithm

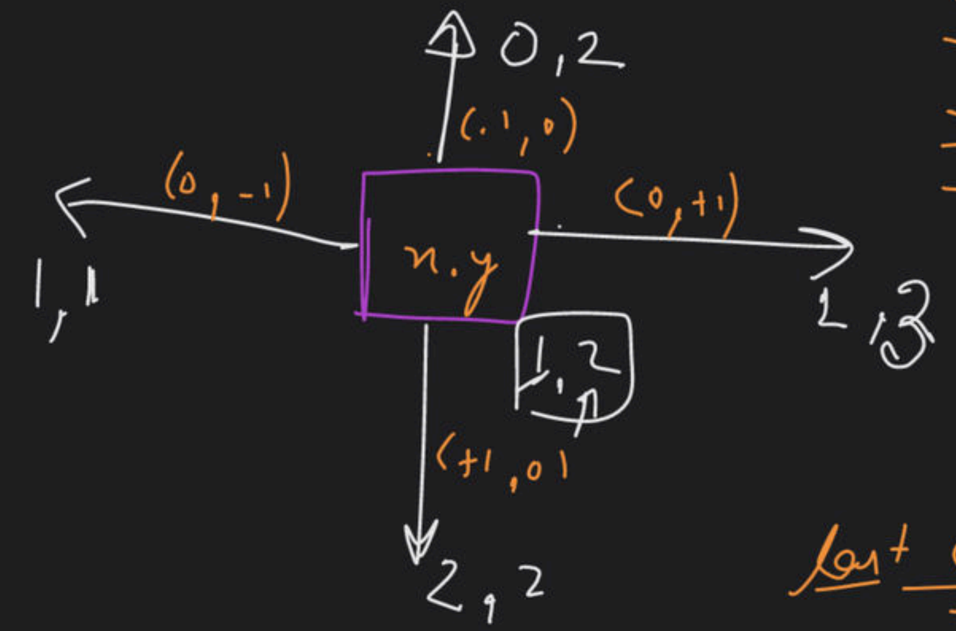
Foundation Course on Data Structures & Algorithm - Part I

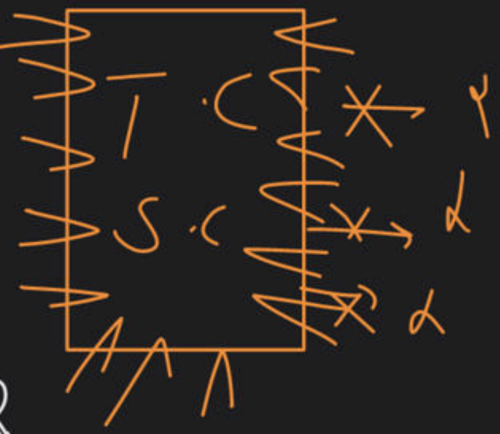
Love Babbar • Lesson 25 • June 8, 2022



Approach :-

start from each cell





- bfs = (x, y-1)
- right = (x, y+1)
- up -> (x-1, y)
- down -> (x+1, y)

last change

length - 1

map < pair < int, int > , bool > visited

visited = false

mxn 2D

→ board

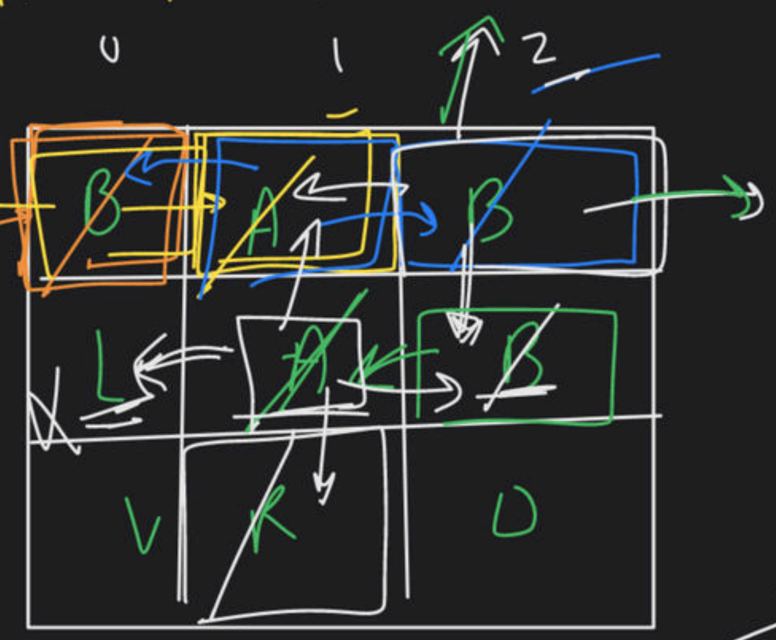
board [i] [j]

= =

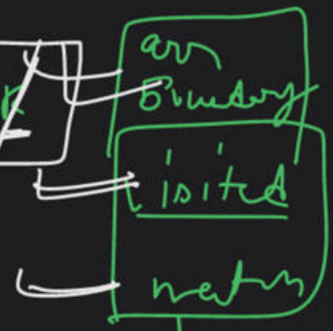
board (i) (j)

= ch;

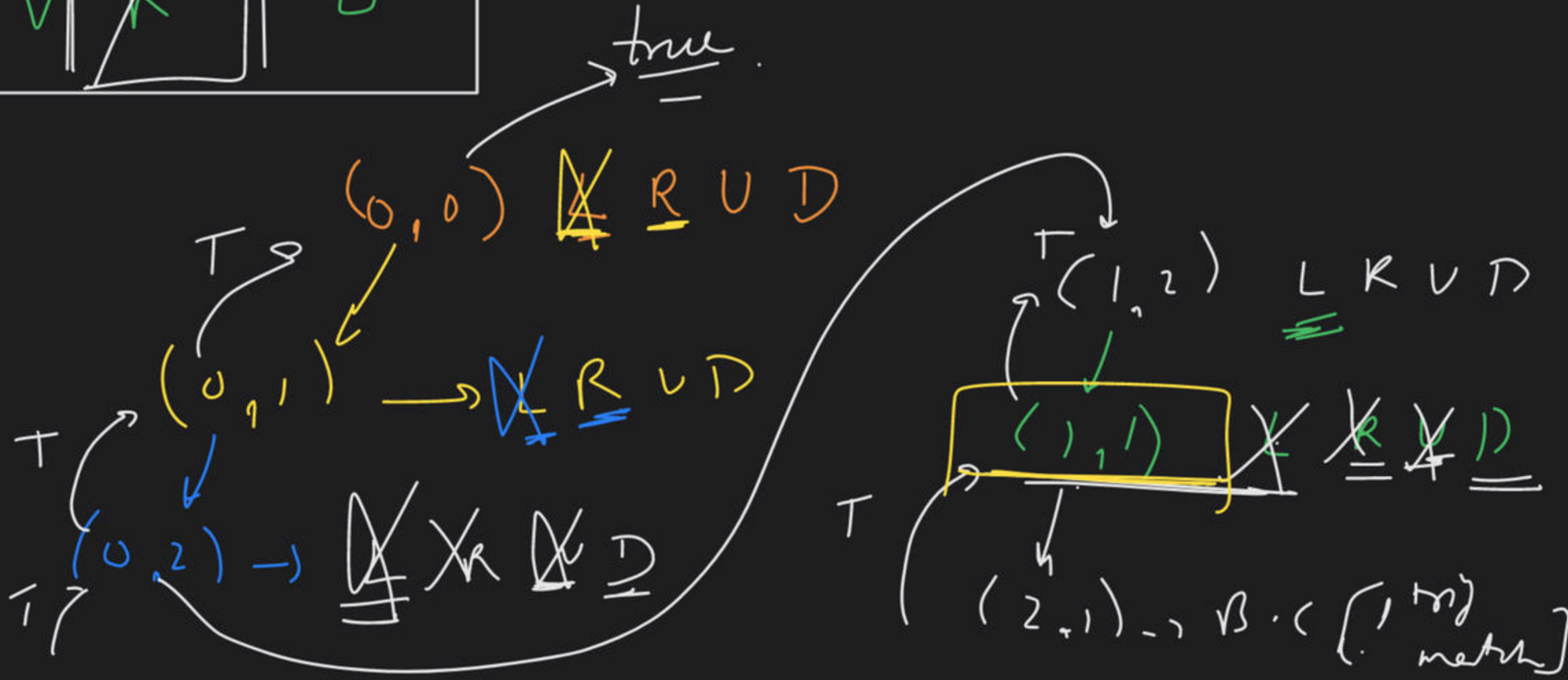
Backtracking

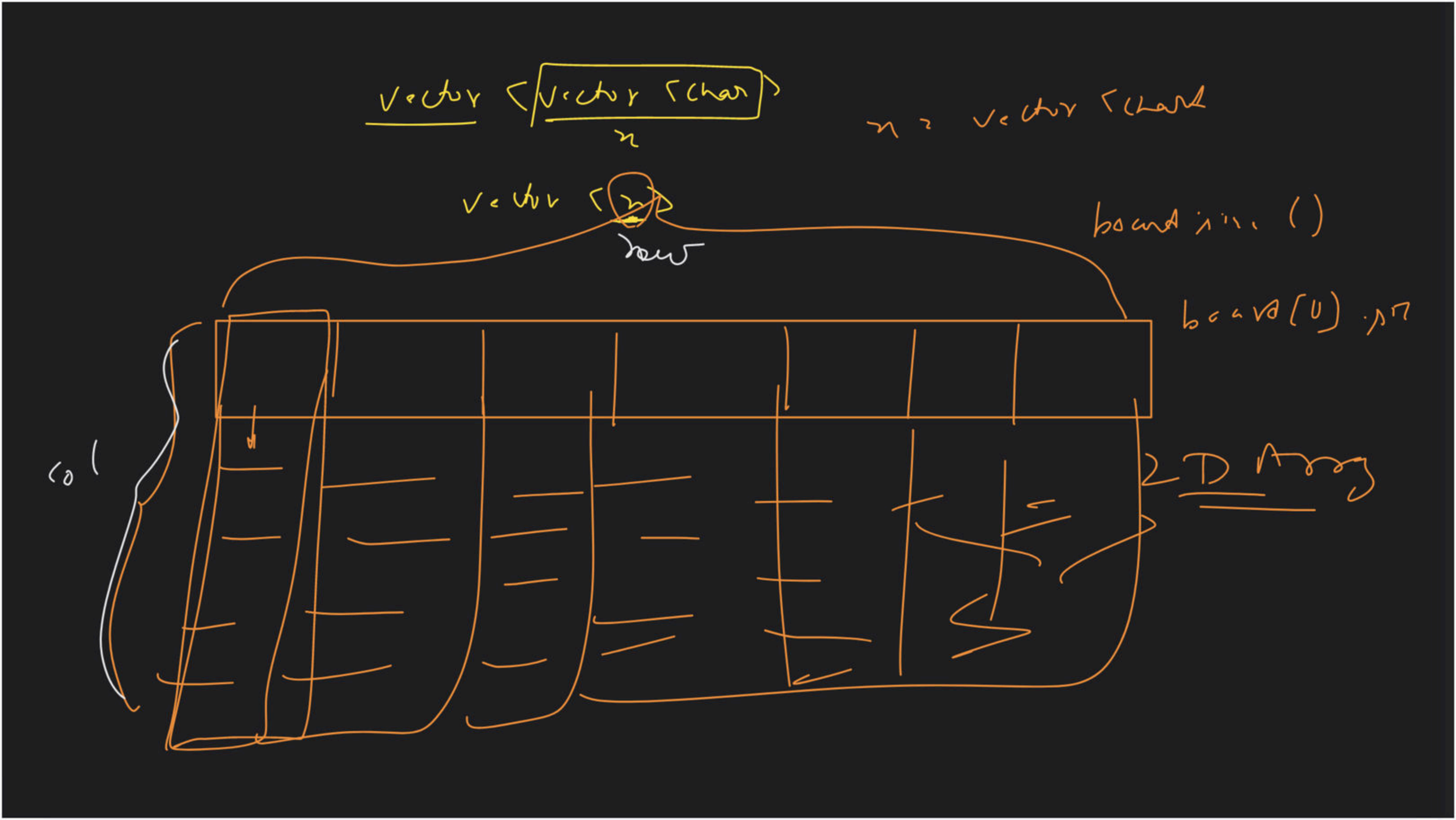


word = B A B B A E

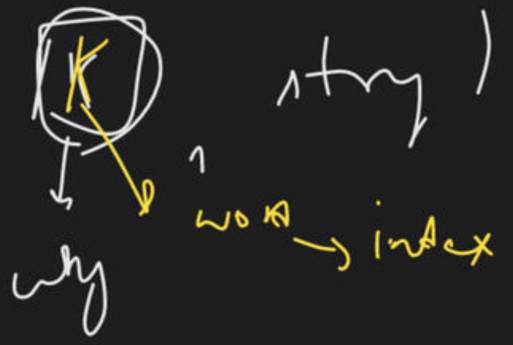


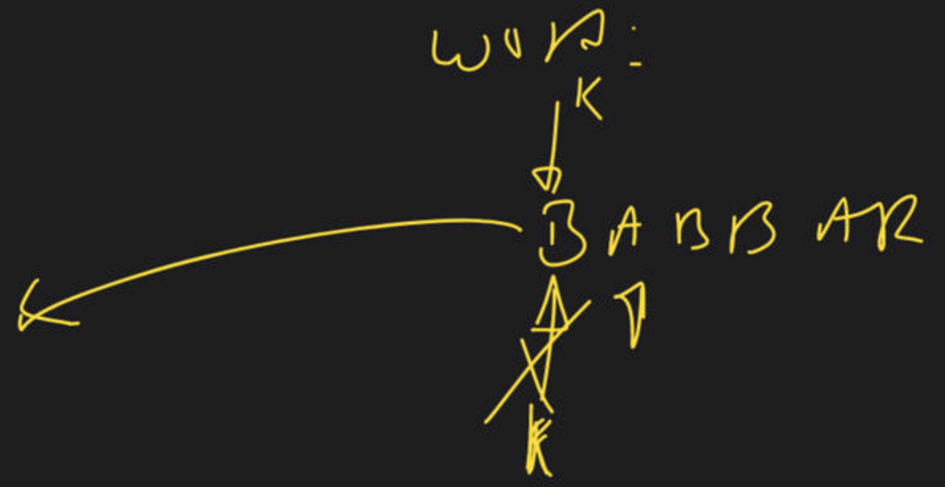
str(-> (0, 0

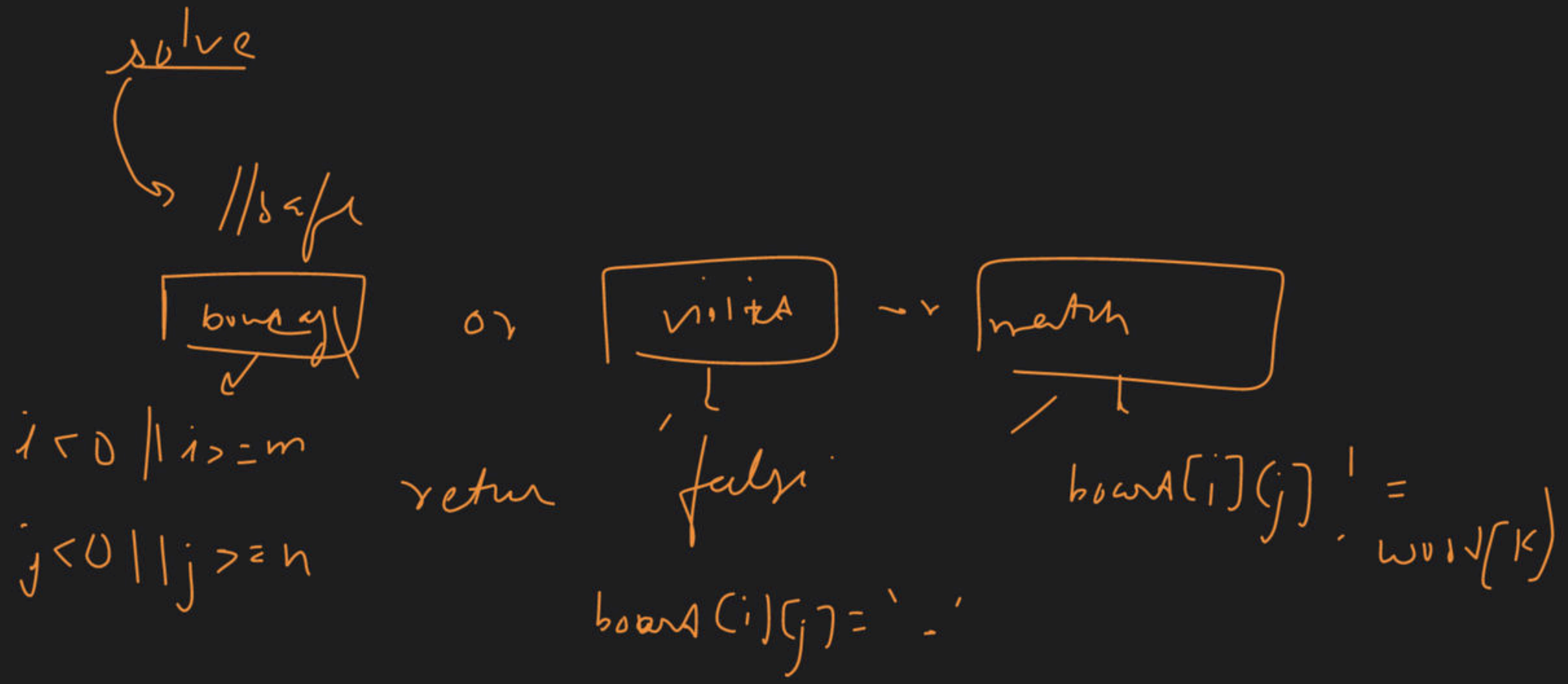



vector<vector<char>>
n
n = vector<char>
vector<n>
rows
board.size()
board[0].size()
col
2D Array

solve (board, i, j, K, word)

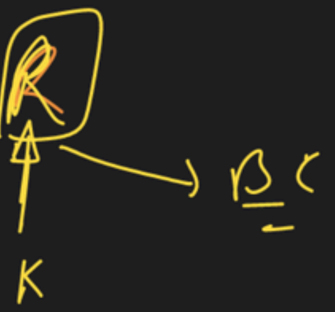


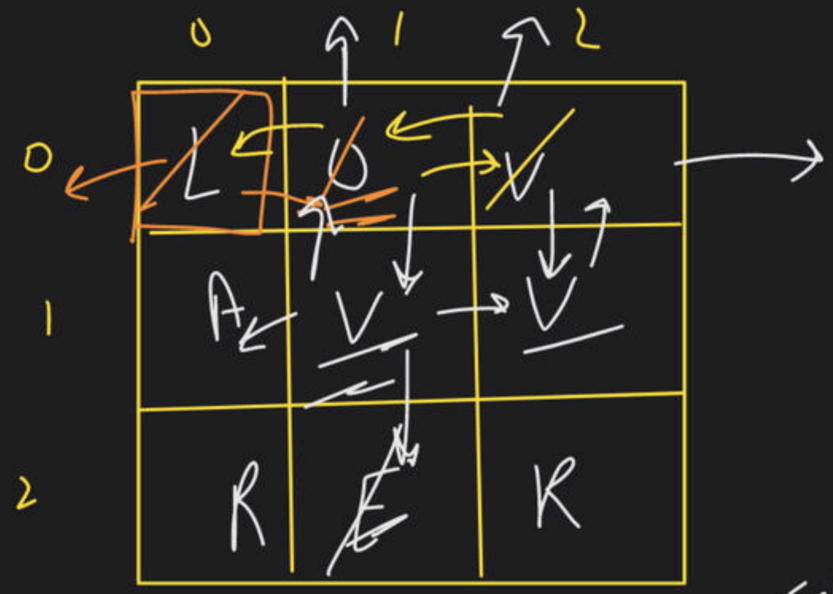



solve
//safe
boundary
or
visited
match
i<0 || i>=m
j<0 || j>=n
return
false
board[i][j] = '.'
board[i][j] !=
word[k]

55 min

word = BABBAR





word:

T

(0,0)

(0,1)

false

(0,2)

T

(1,1)

T

(2,1)

Boundary

Visited

Mark

```
bool exist ( vector<vector<char>>& board, string word)
{
    int rows = board.size();
    int cols = board[0].size();

    for (int i=0; i< rows; i++)
    {
        for ( int j = 0; j<cols; j++)
        {
            if (solve( board, i, j, 0, word))
                return true;
        }
    }
    return false;
}
```

end;

How?

3 3 3

i<-1 ?

?

bool Solve (vector<vector<char>> & board, int i, int j, int k, string word)

{

// Boundary cases

if ((i < 0 || i >= rows || j < 0 || j >= cols) ||

(board[i][j] == '*') || (word[k] != board[i][j]))

{ return false; }

isSafe()

// B.C

if (k == word.length() - 1)

return true;

k

word.length() - 1

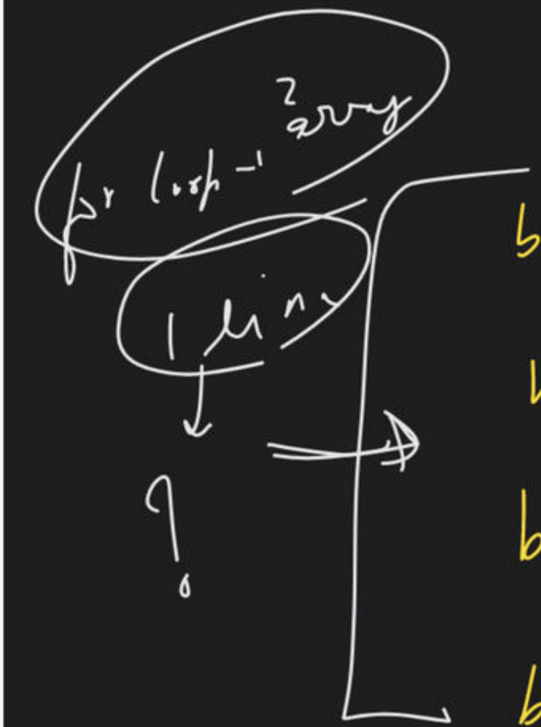
char ch = board[i][j];

// visited

board[i][j] = '*';

Success -> stroy Tent

Carry



// LRUD

bool left = if (isSafe(i,j-1) solve(board, i, j-1, k+1, word);

bool right = if (isSafe(i,j+1) solve(board, i, j+1, k+1, word);

bool up = if (isSafe(i-1,j) solve(board, i-1, j, k+1, word)

bool down = if (isSafe(i+1,j) solve(board, i+1, j, k+1, word)

// Backtrack

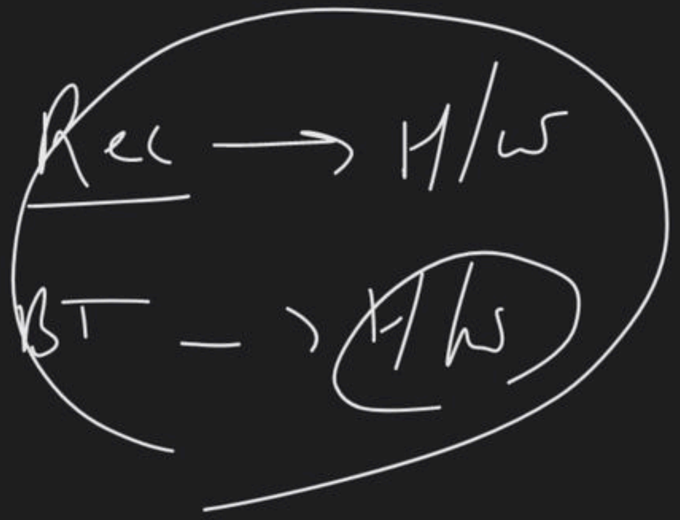
board[i][j] = ch;

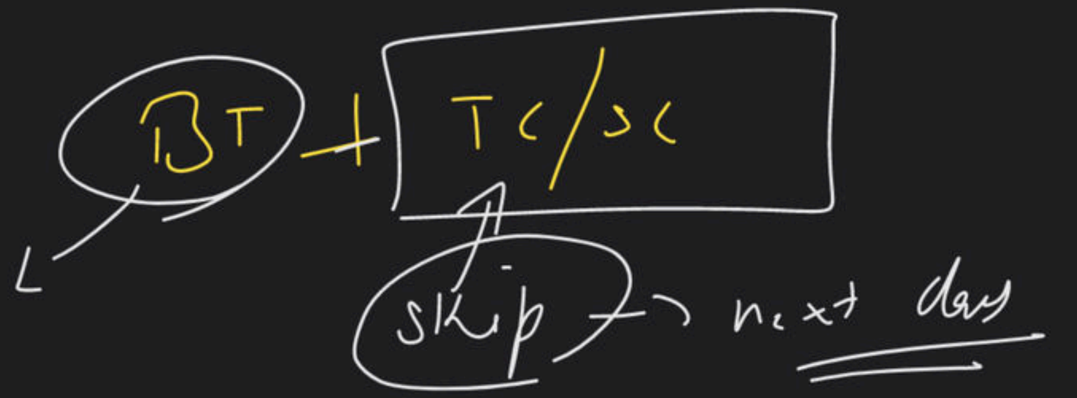
return

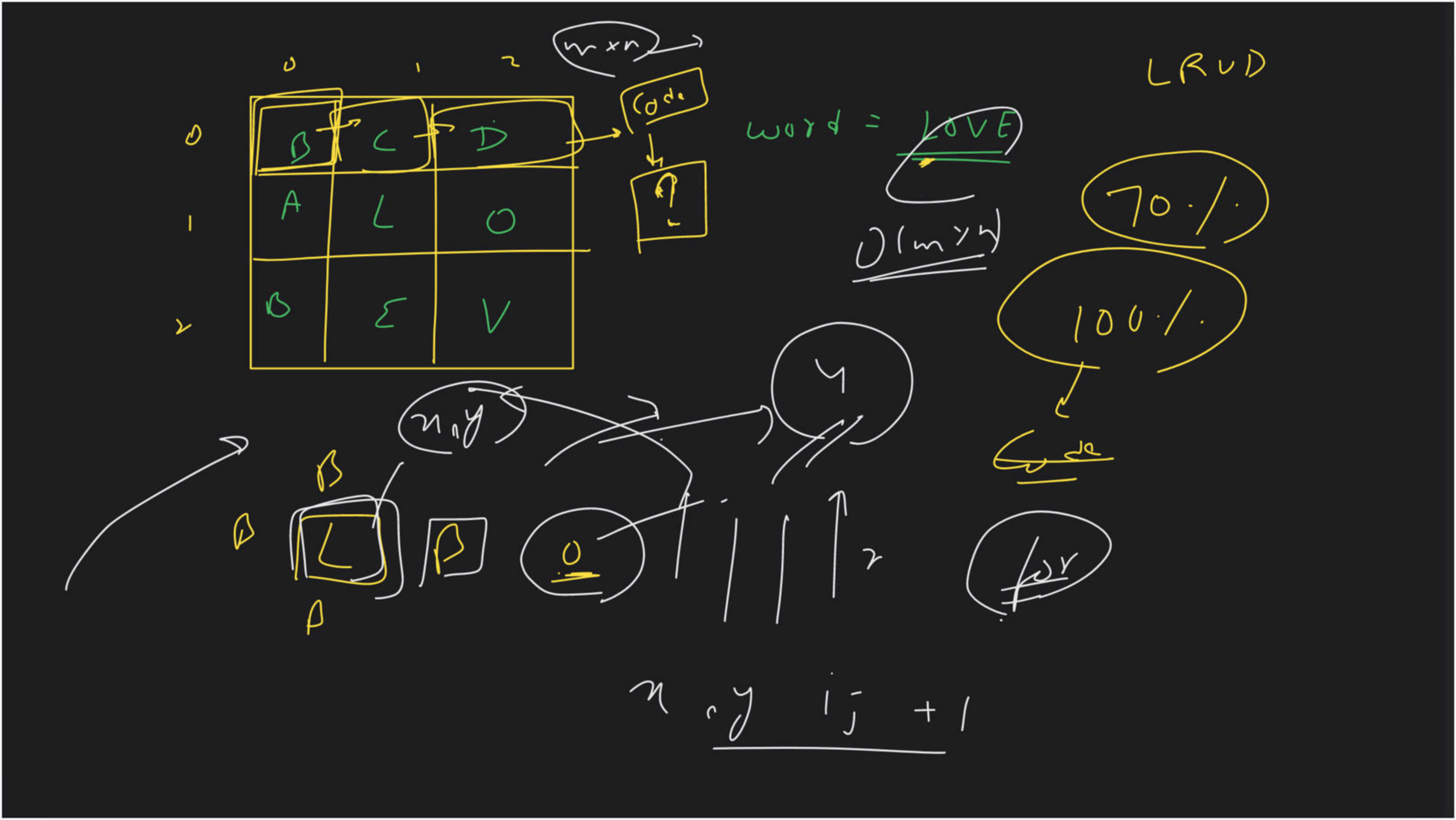
}

left or Right or down or up;

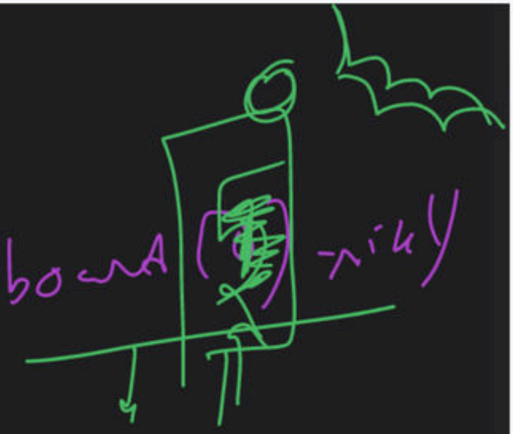
Student






m x n
Code
?
word = LOVE
O(m x n)
LRUD
70 %
100 %
Code
B
C
D
A
L
O
E
V
x, y
y
L
O
for
i j +1

```
int rows = board.size()
int cols = board[0].size()
```



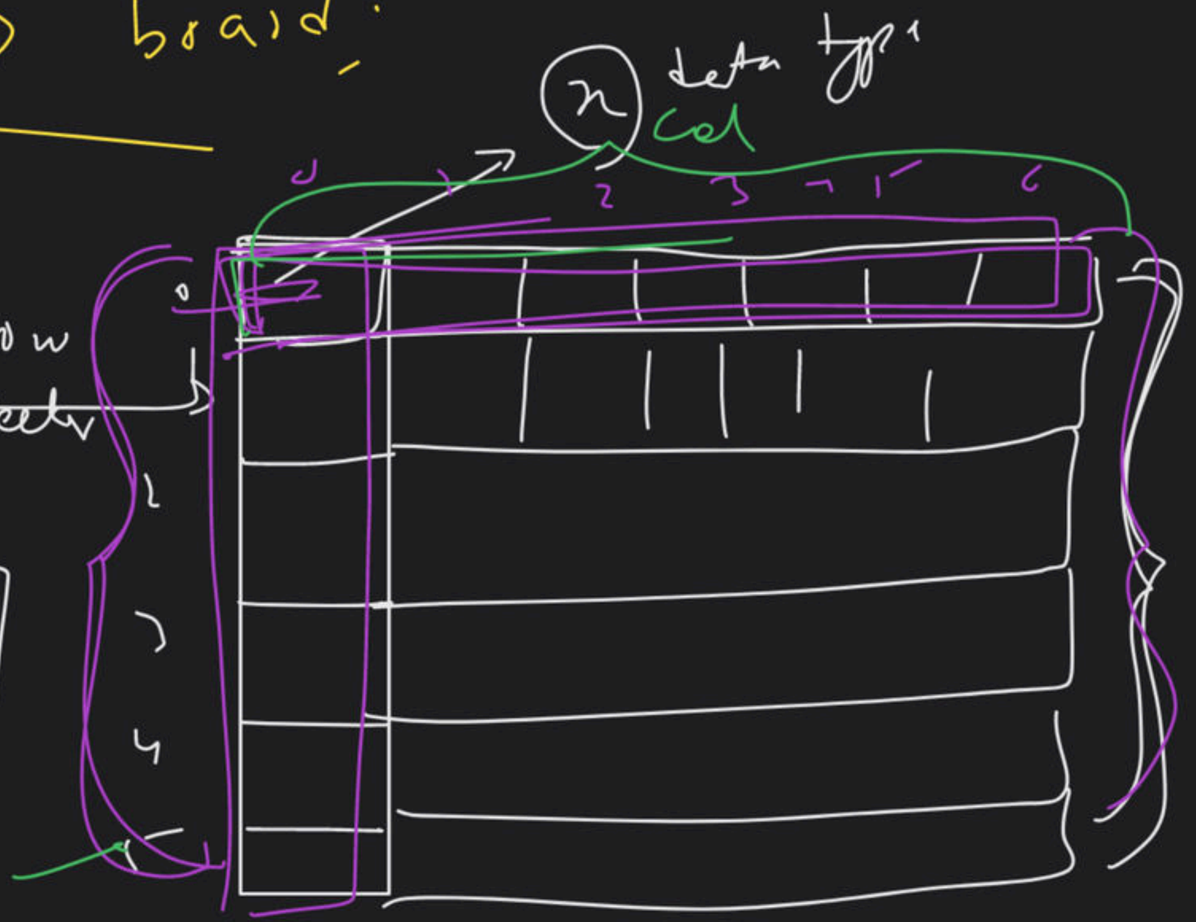

```
vector<vector<char>> board;
```

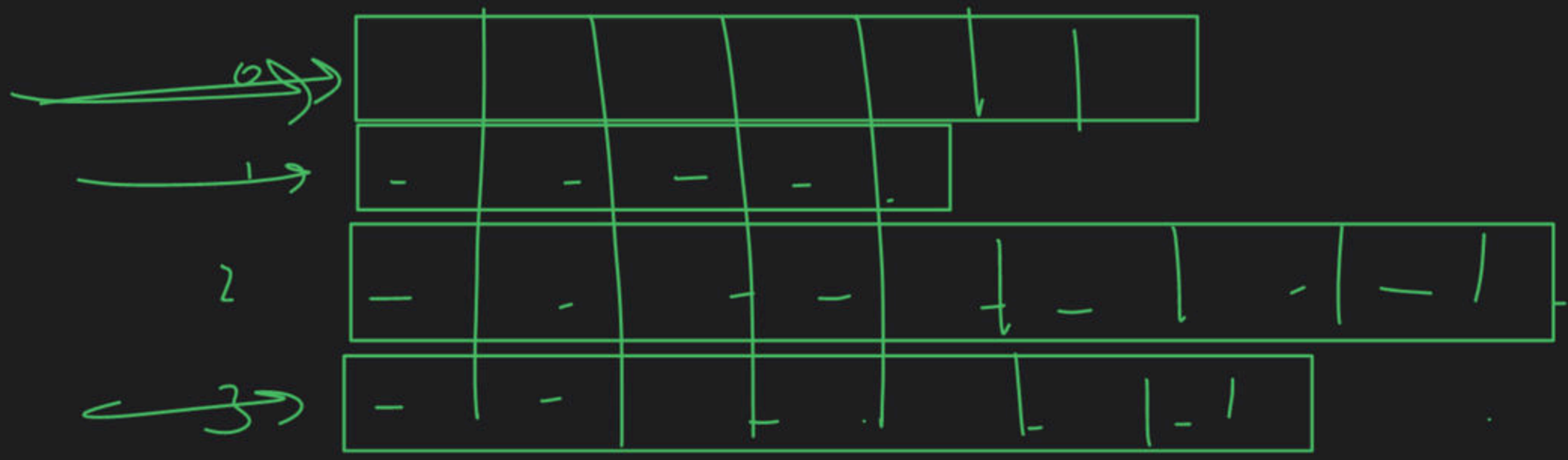
×

```
vector<int> board
```

Ankit

n = vector<char>





STL

Video

Next DS

for (int i = 0; i < board.size(); i++)

{

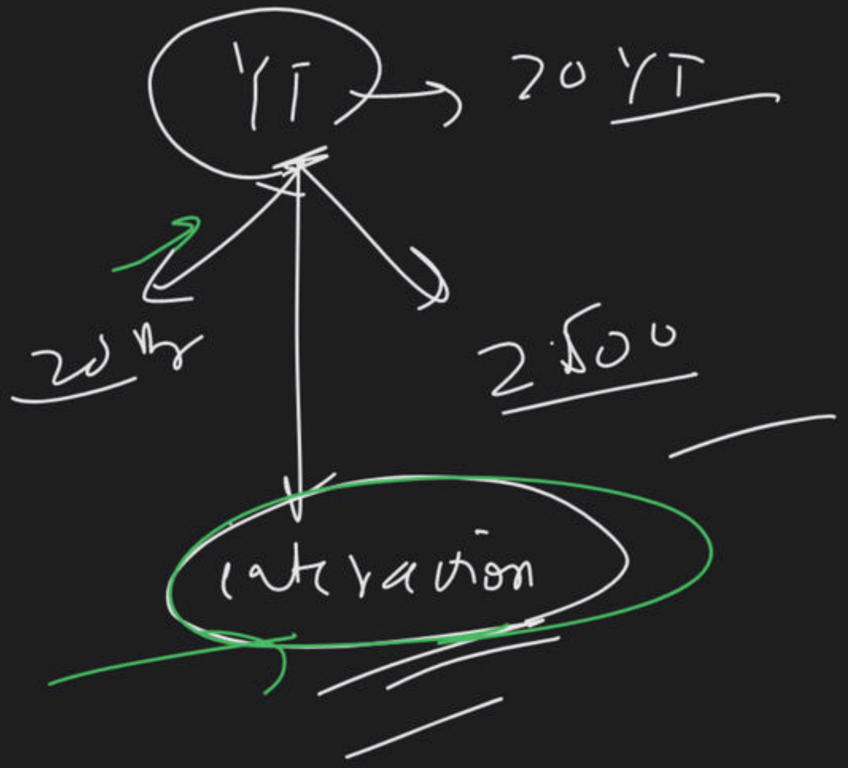
int col = board[i].size();

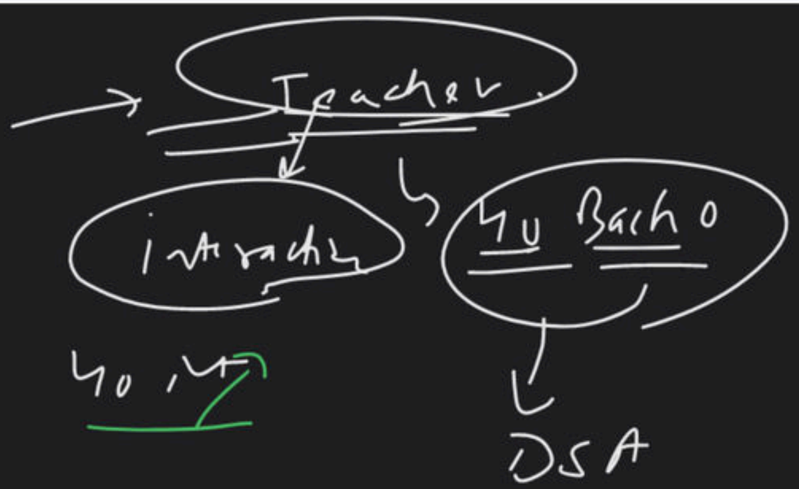
cout << "hello" << col

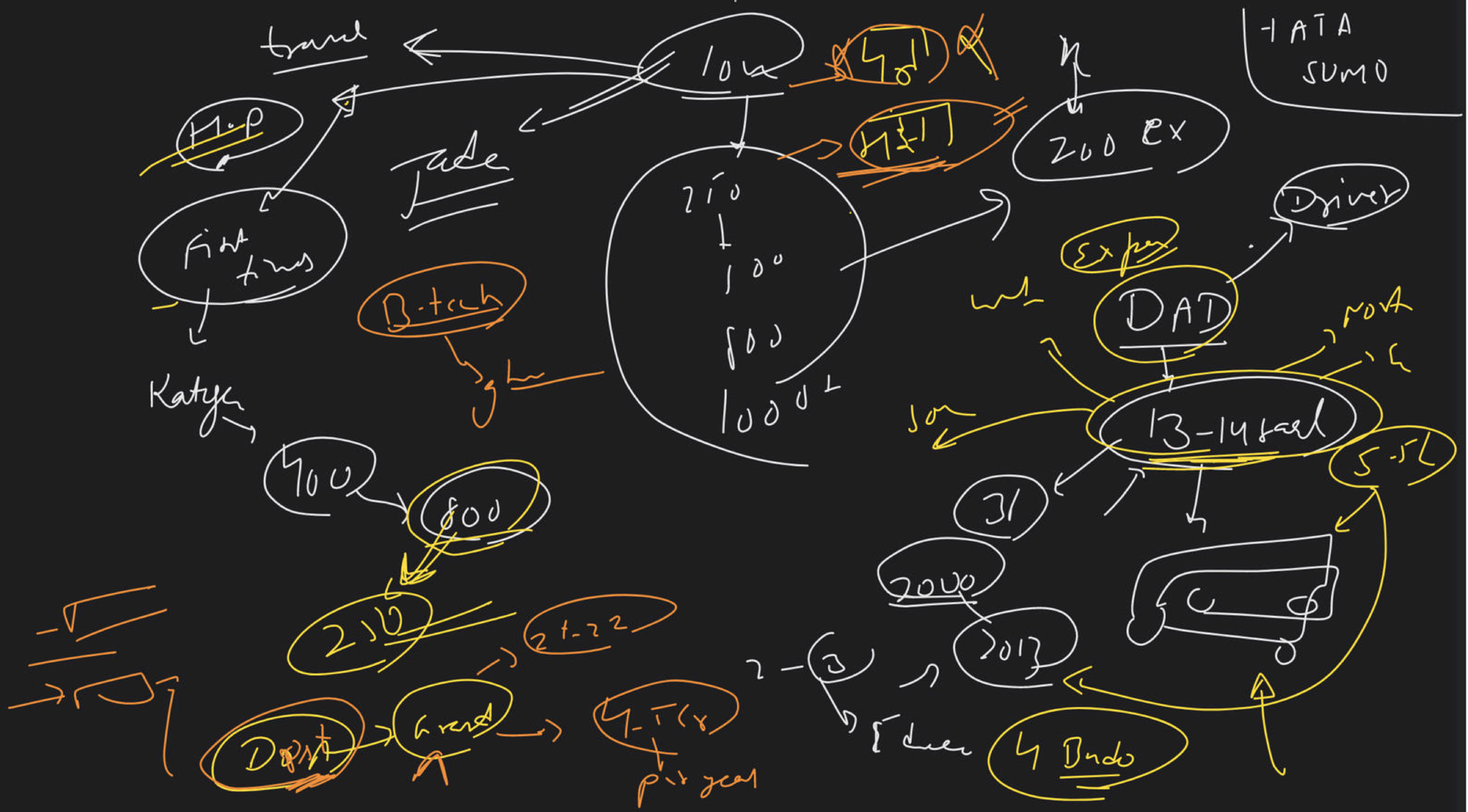
}

Rec

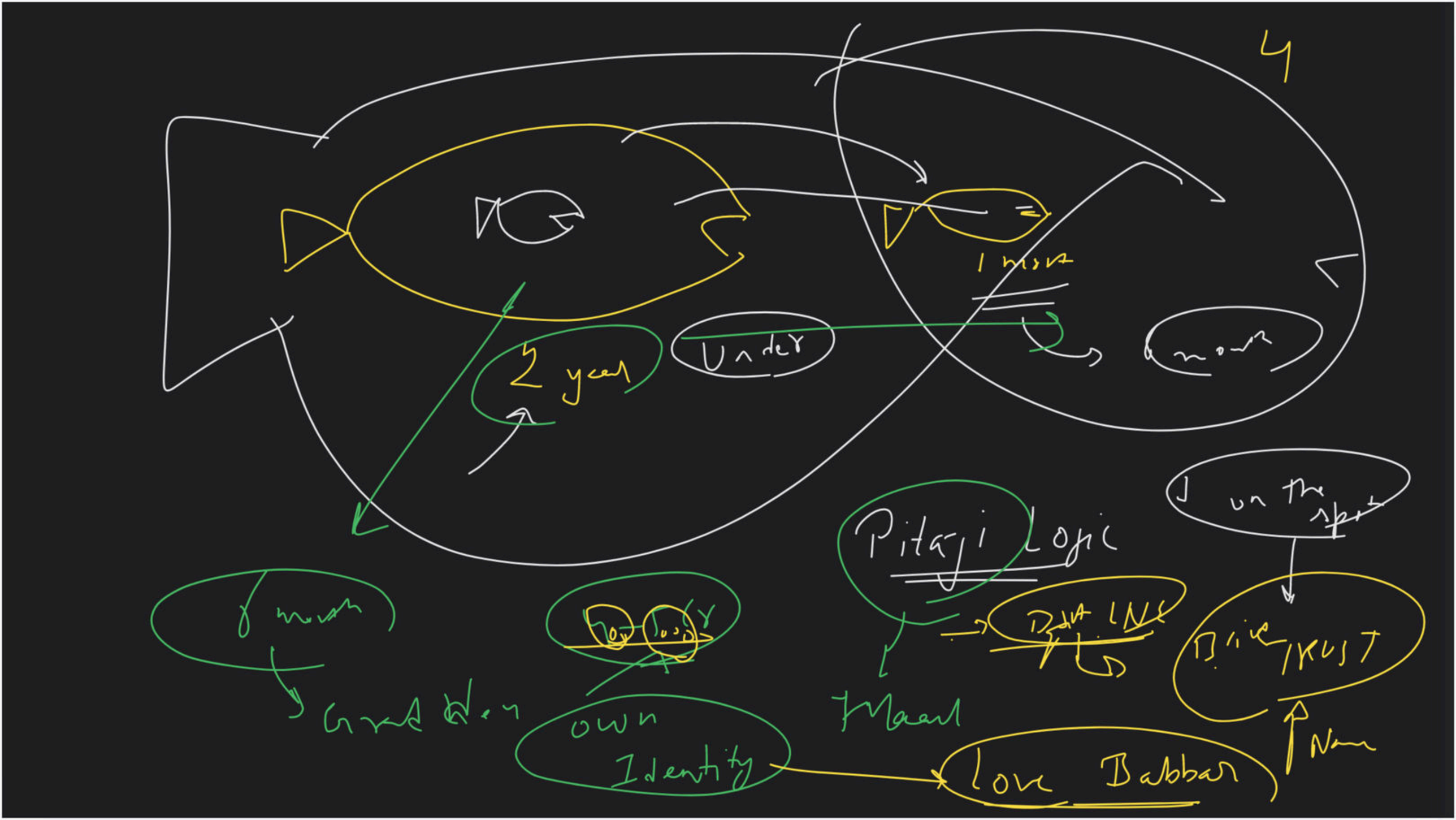
Backtracking

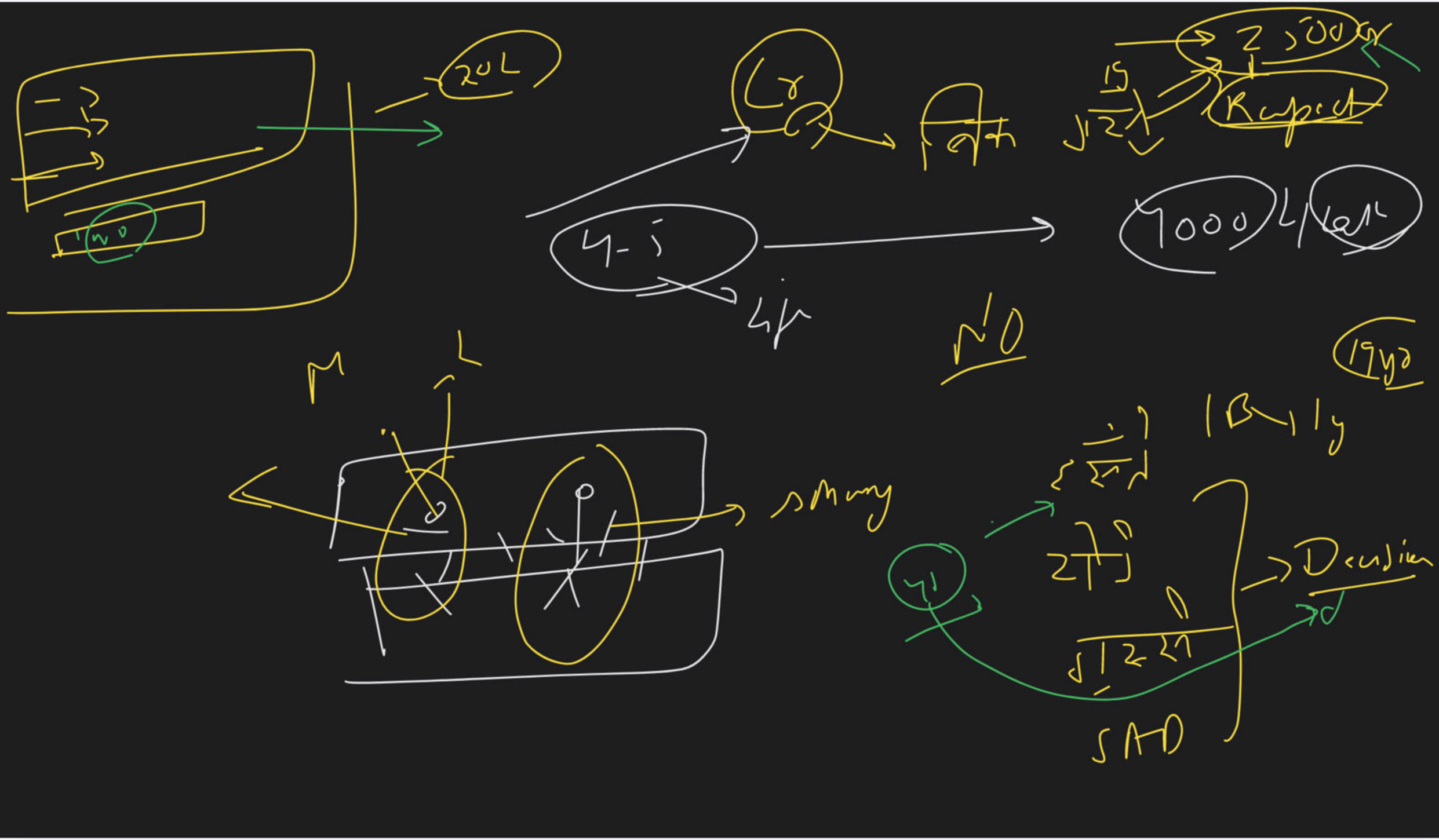

YT
20 YT
20 Mr
2500
interaction

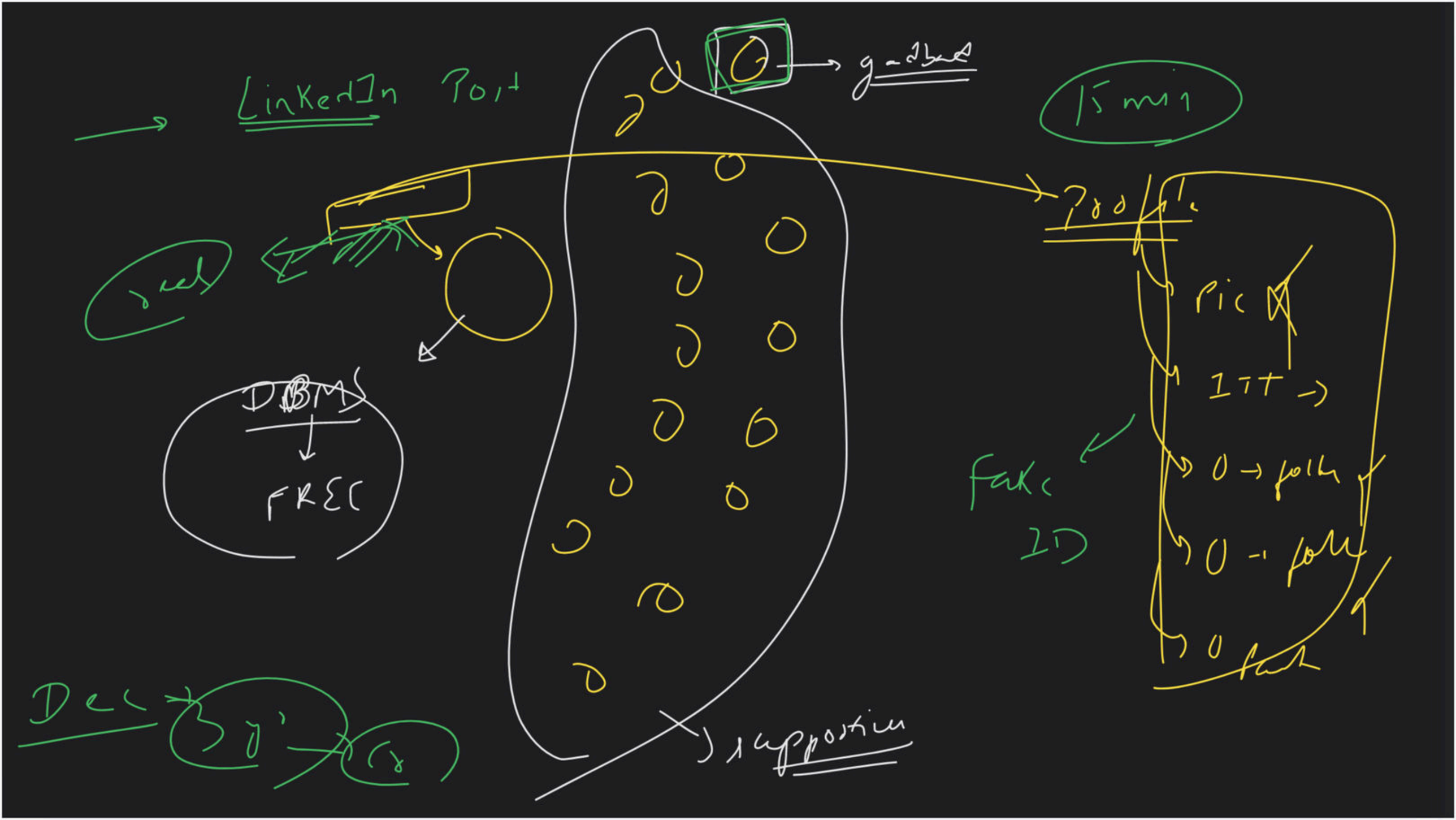

Teacher
interaction
40 M
Mu Bacho
DSA



travel
10th
TATA
SUMO
200 ex
Driver
DAD
13-14 saal
31
2000
2013
B.tech
400
800
21-22


4
2 years
Under
Pitaji Logic
Own
Identity
love Babbar




LinkedIn Post
DBMS
FREE
Fake
ID
Pic
Init
0 → fetch
0 → fetch
0 fail

Learning

2 - 3

2B → Life
→ 1 Lakh

100000

17 yo
Kaatya
0 yr

333000

3.50L

500 Cr

1 Friend → PAPA

downfall

20

1L

10

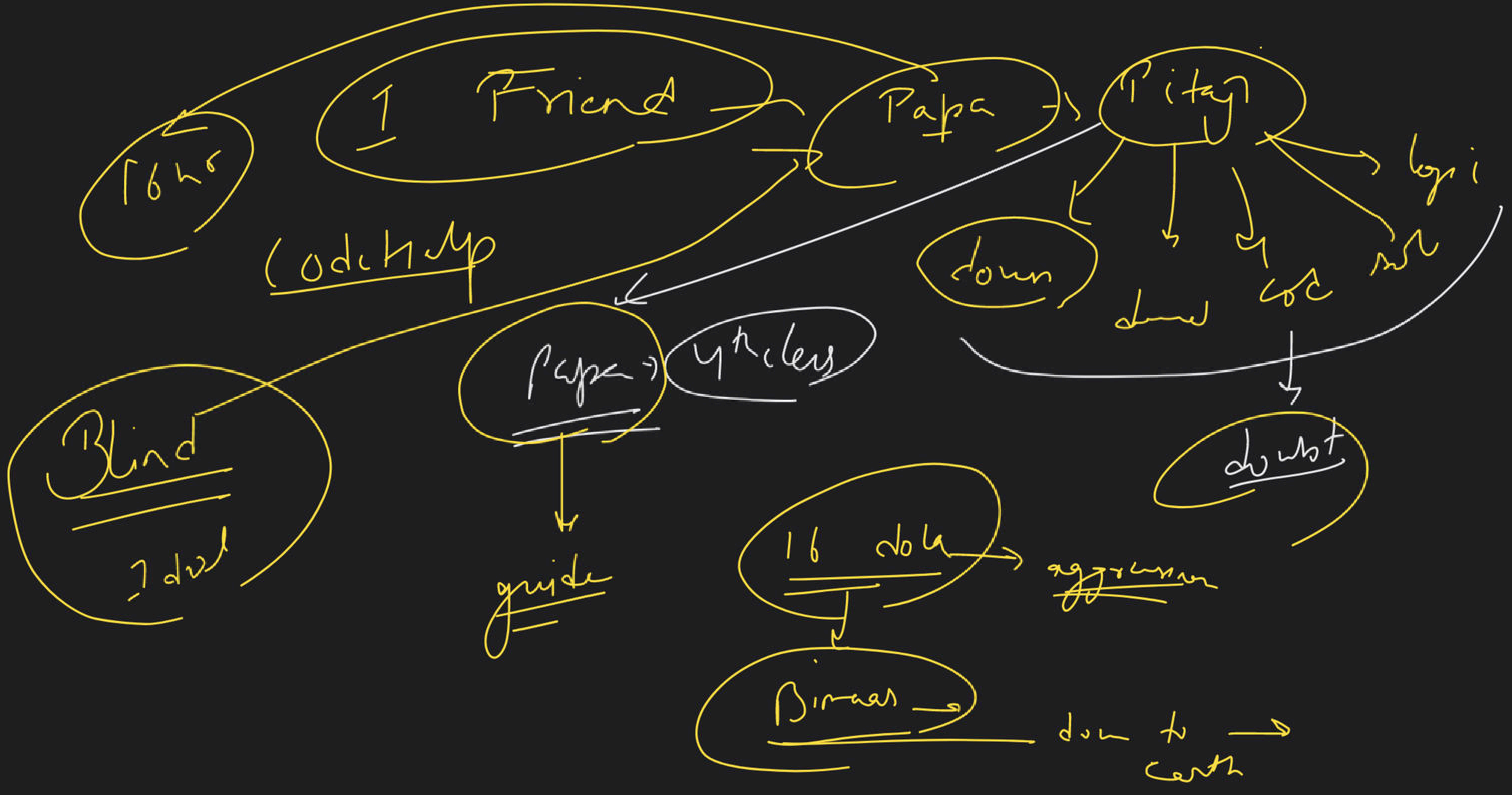
1

20

ghar walo ko pata de rhi hai

5000 → Kharch kr

10<8


I Friend
Papa
Blind
Papa
guide
down
doubt
aggression
down to earth

1st placement → 14.4 LPA

23 LPA

ab apni to lag gyi

25000

ab aphe jaiso ki lagwani hai

exp

Amu

1300

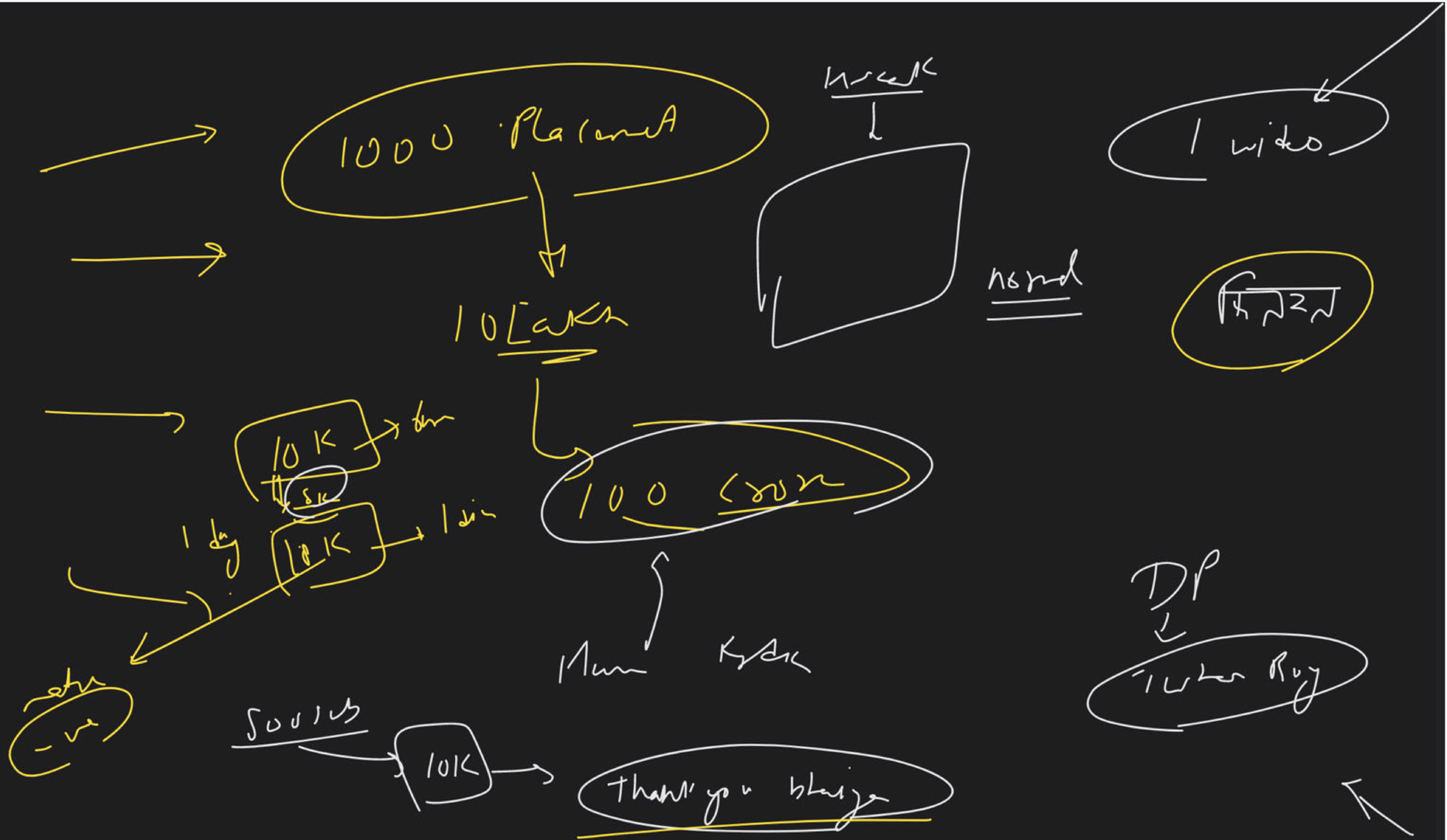
5000

1000

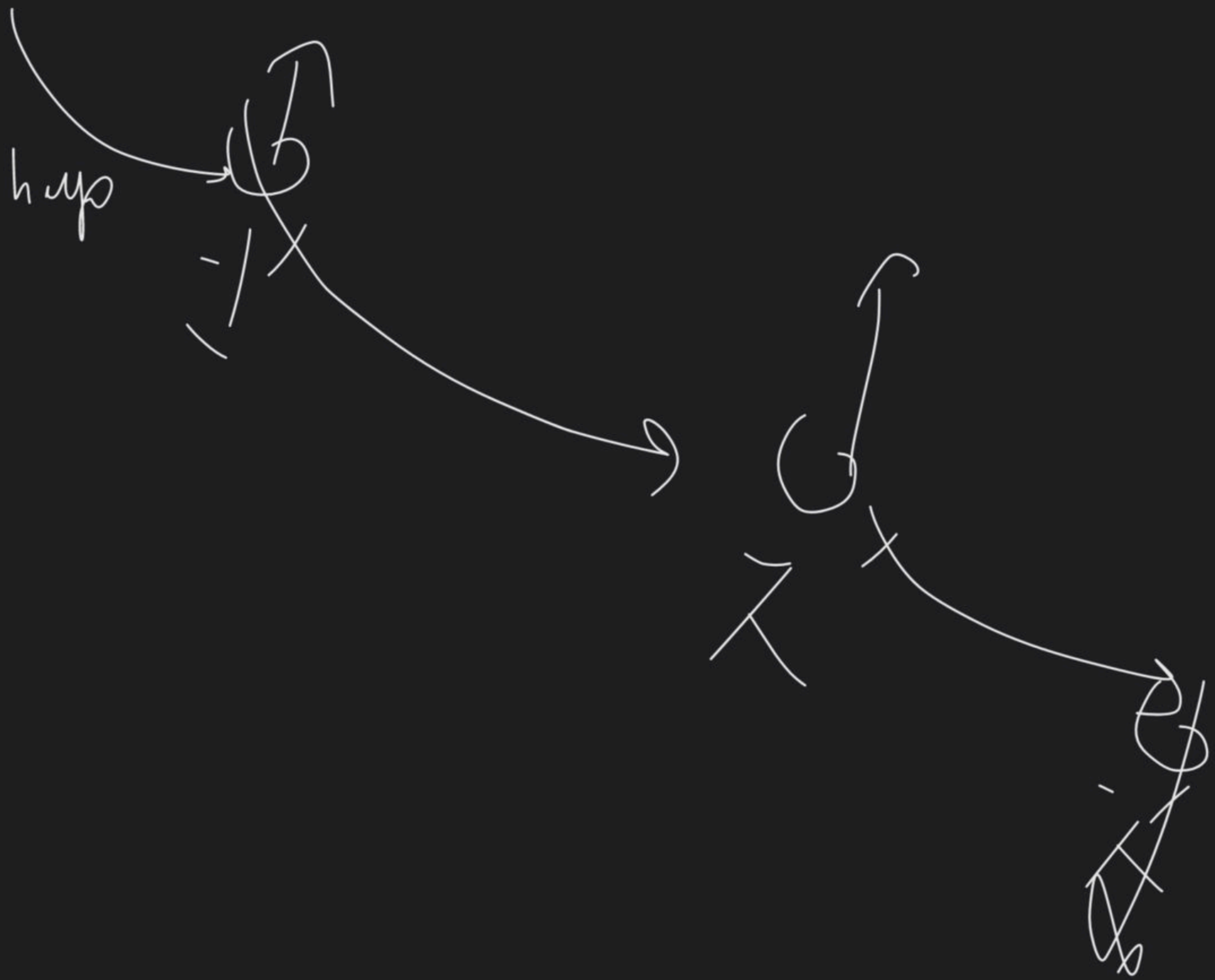
ICE → 7-8

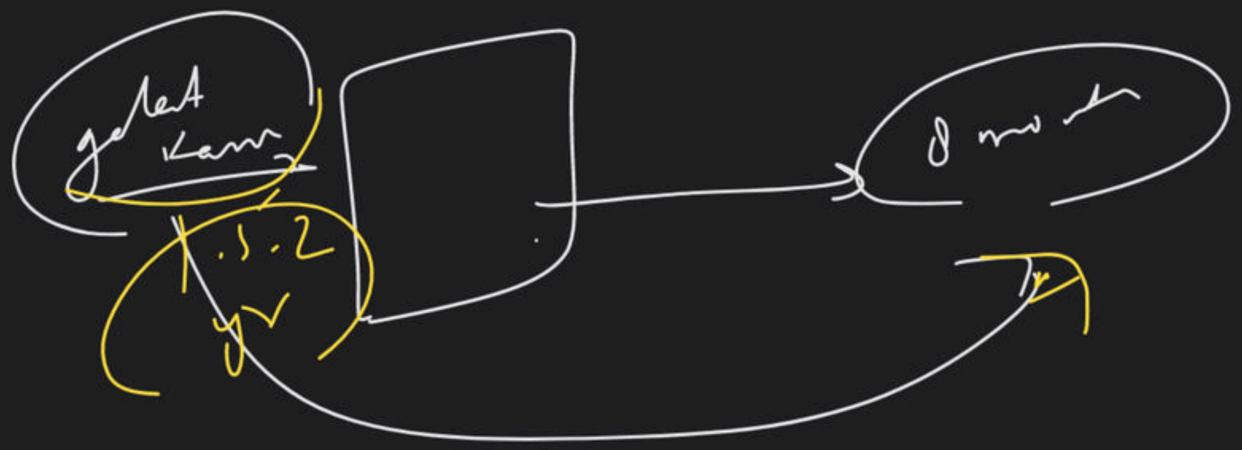
Brand

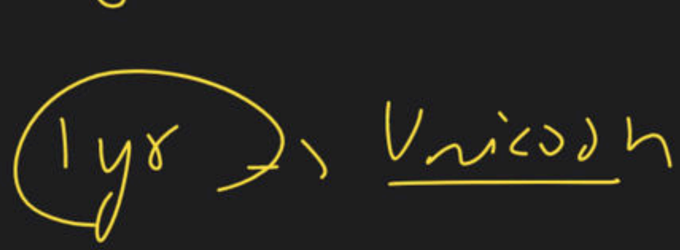
10 x 12

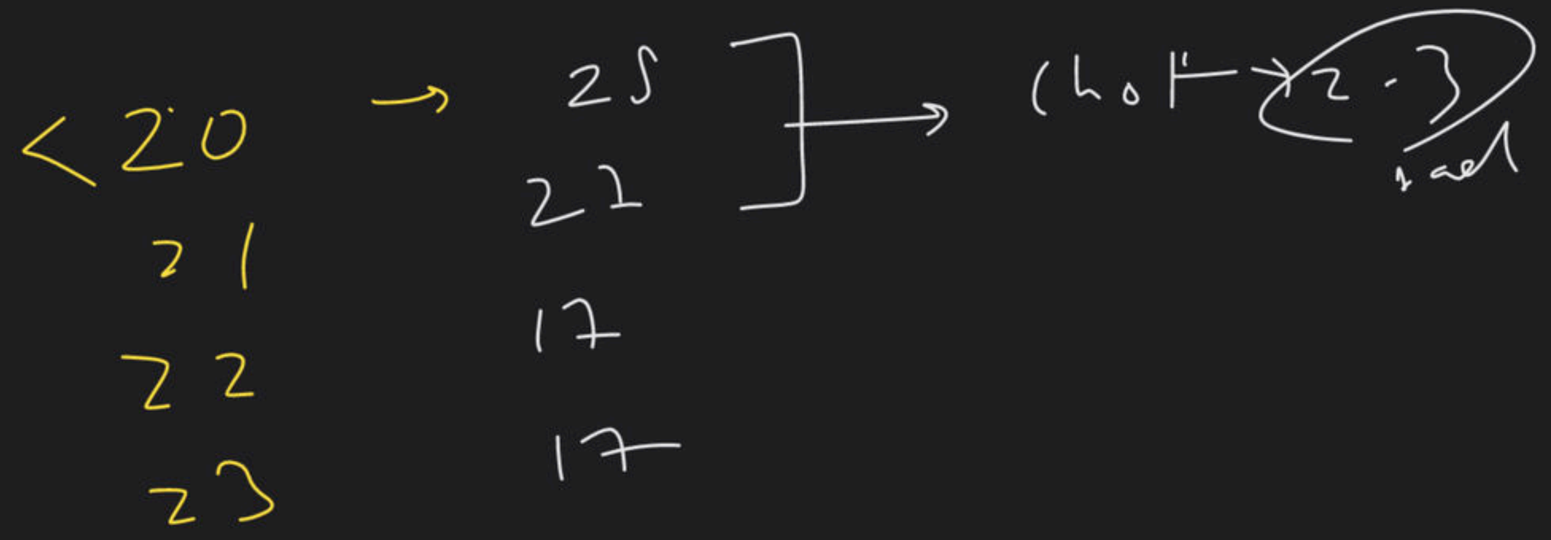


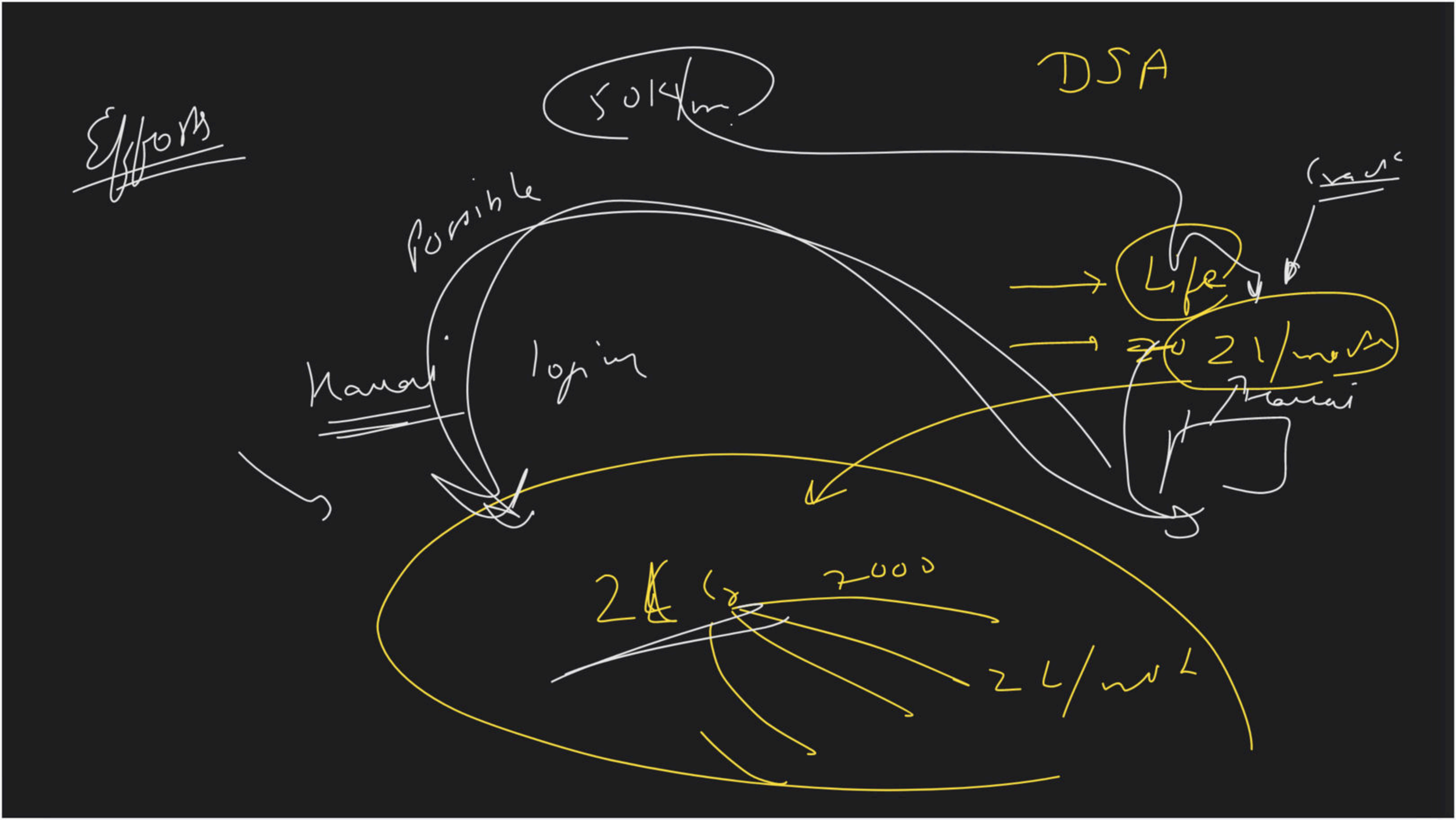
1000 Placement
10 Lakh
100 crore
10K
5K
10K
1 day
1 min
-ve
Sources
10K
Thank you bhaiya
1 week
normal
1 video
DP

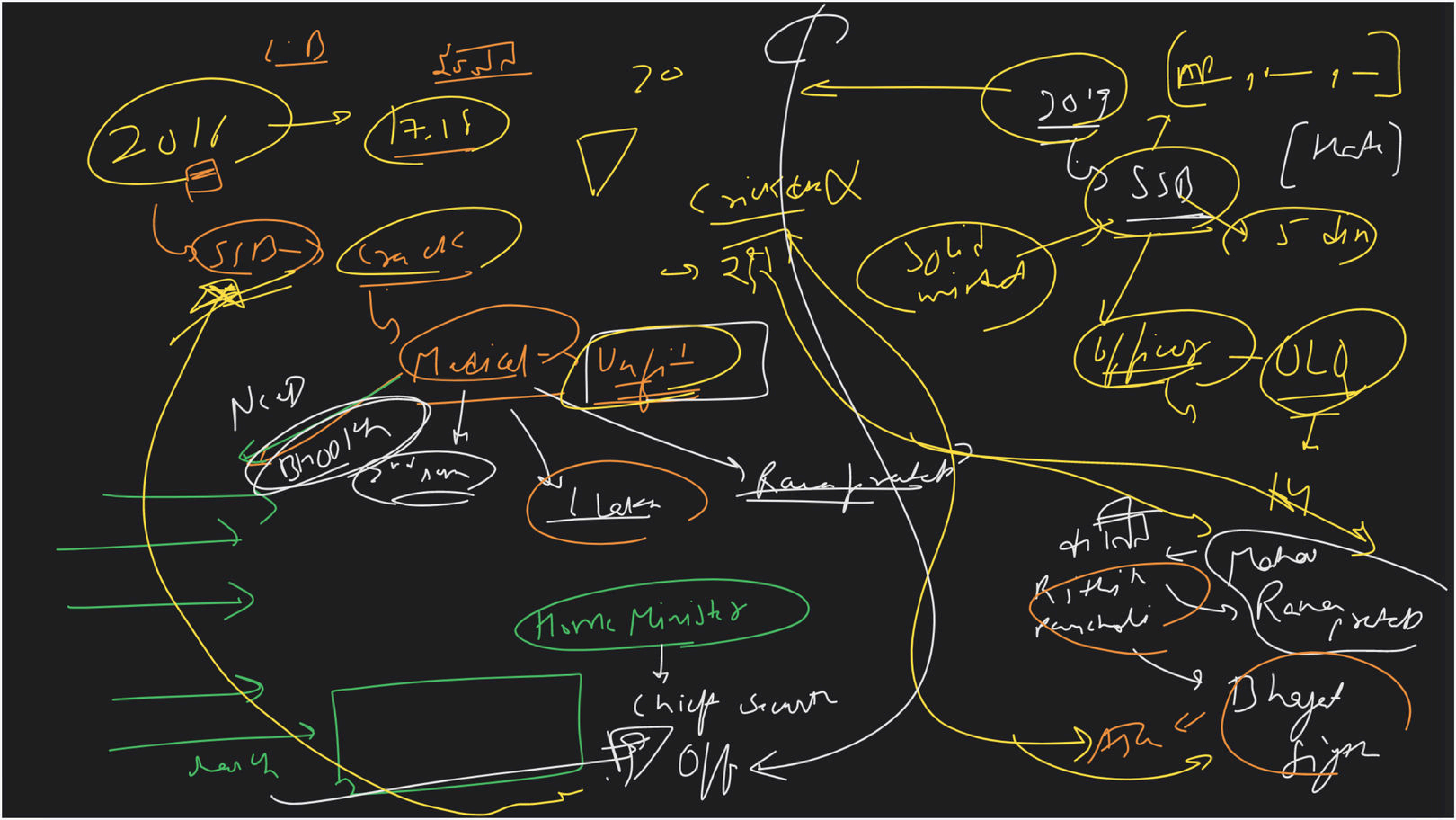

hup

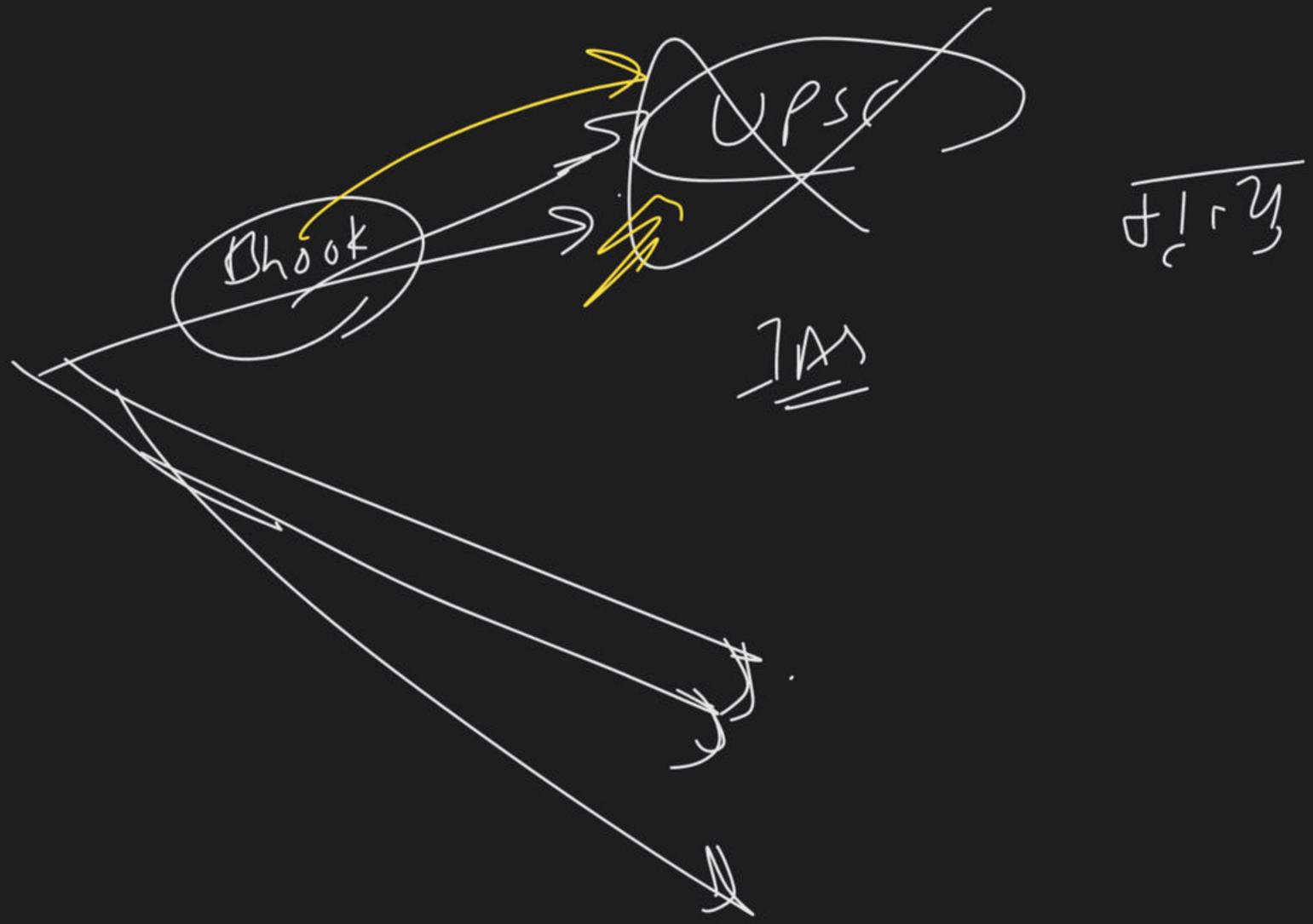

select
Name
1.1.2

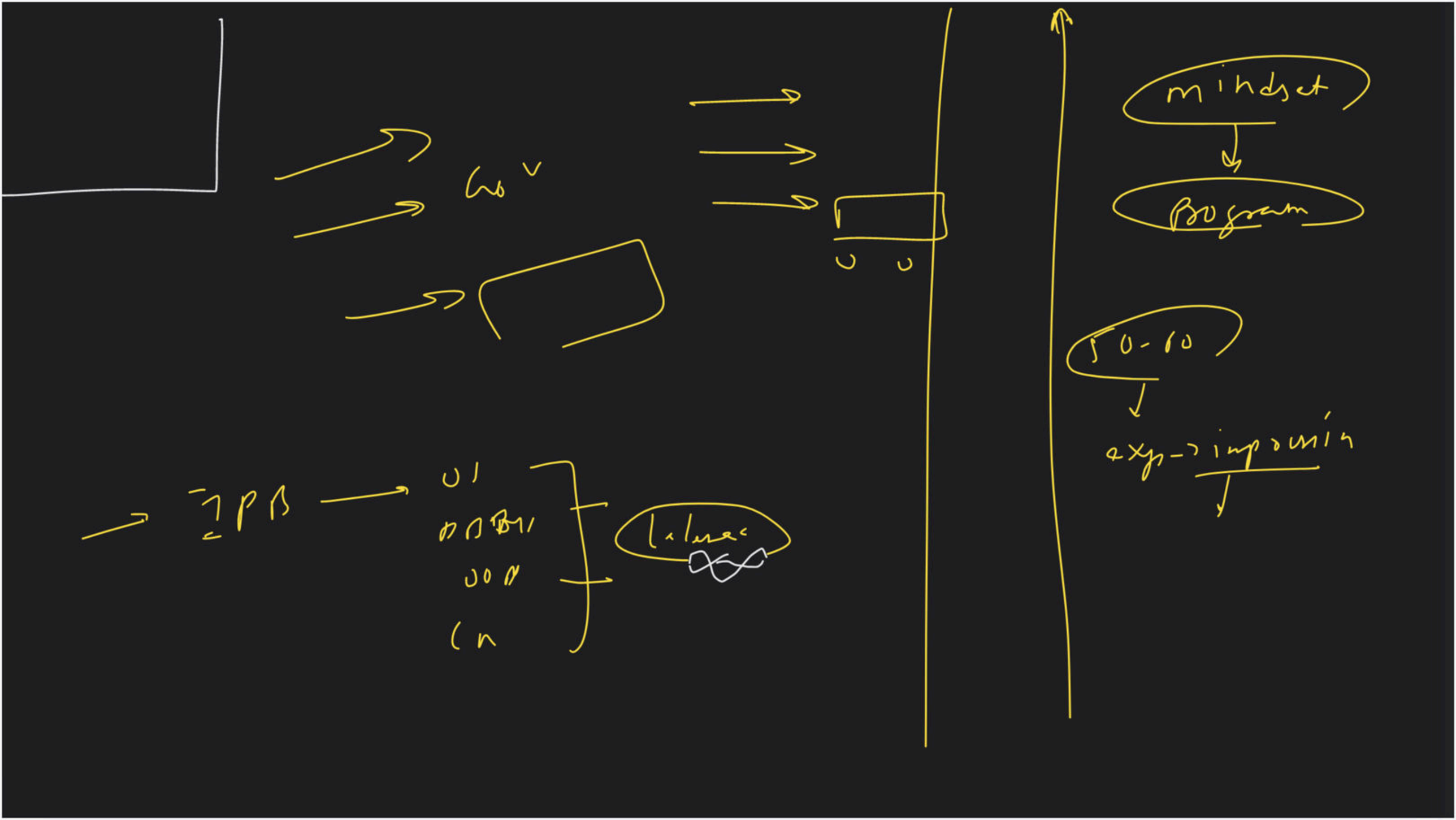

Unicode

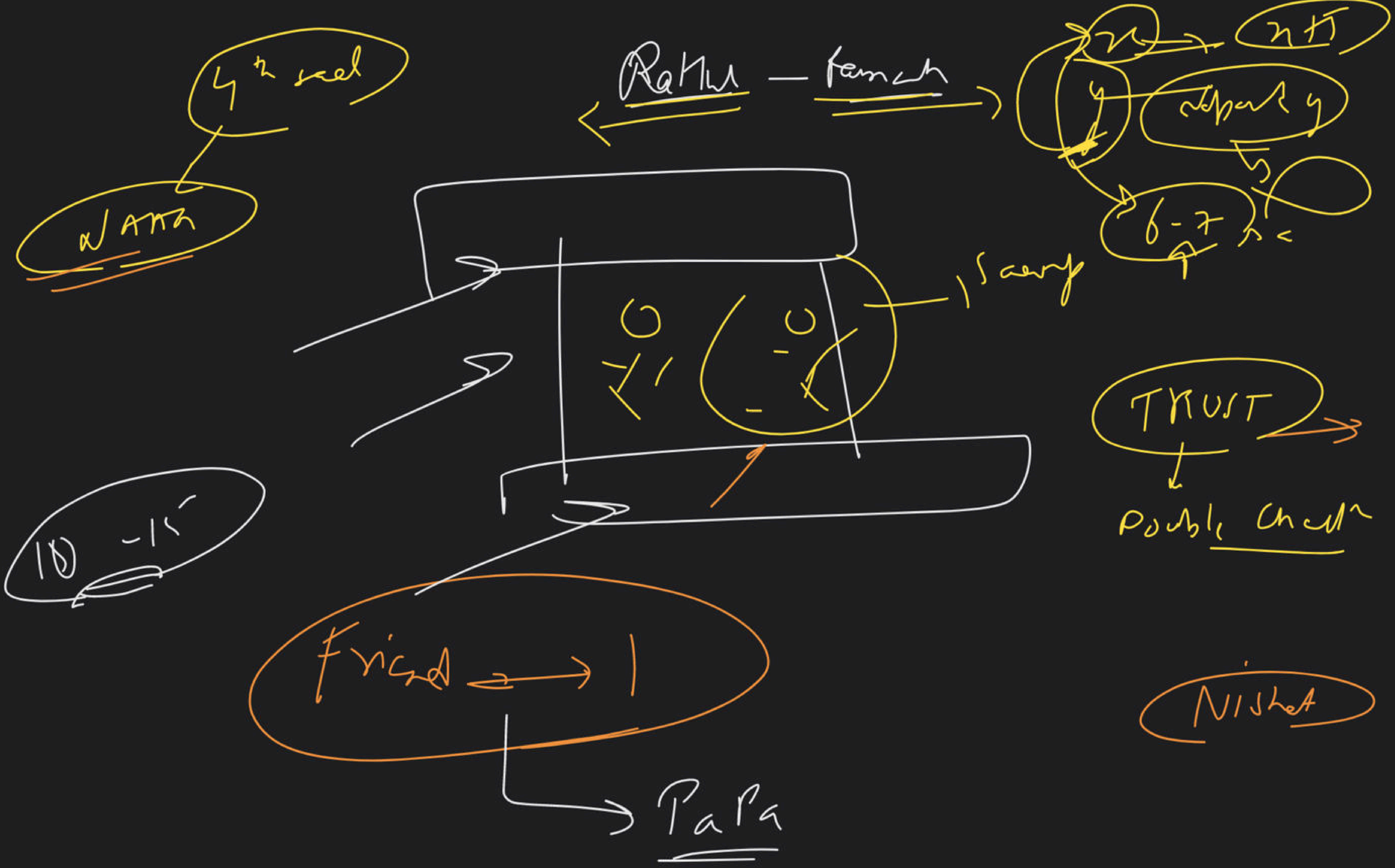

< 20
21
22
23
25
22
17
17
Hot
2.3


Efforts
50 LPA
DSA
Possible
Manual
login
Life
2021
2000
2L/mnth

 2016
17.18
20
SSB
Crack
Medical
Unfit
Need
2019
SSB
Officer
OLQ
Home Minister


Bhook
UPSC
IAS


mindset
Program
for 0-10
exp-> improving


Rahul
10 -15
6-7
TRUST
Double Check
Friend → 1
PaPa
Nishat

90%

true