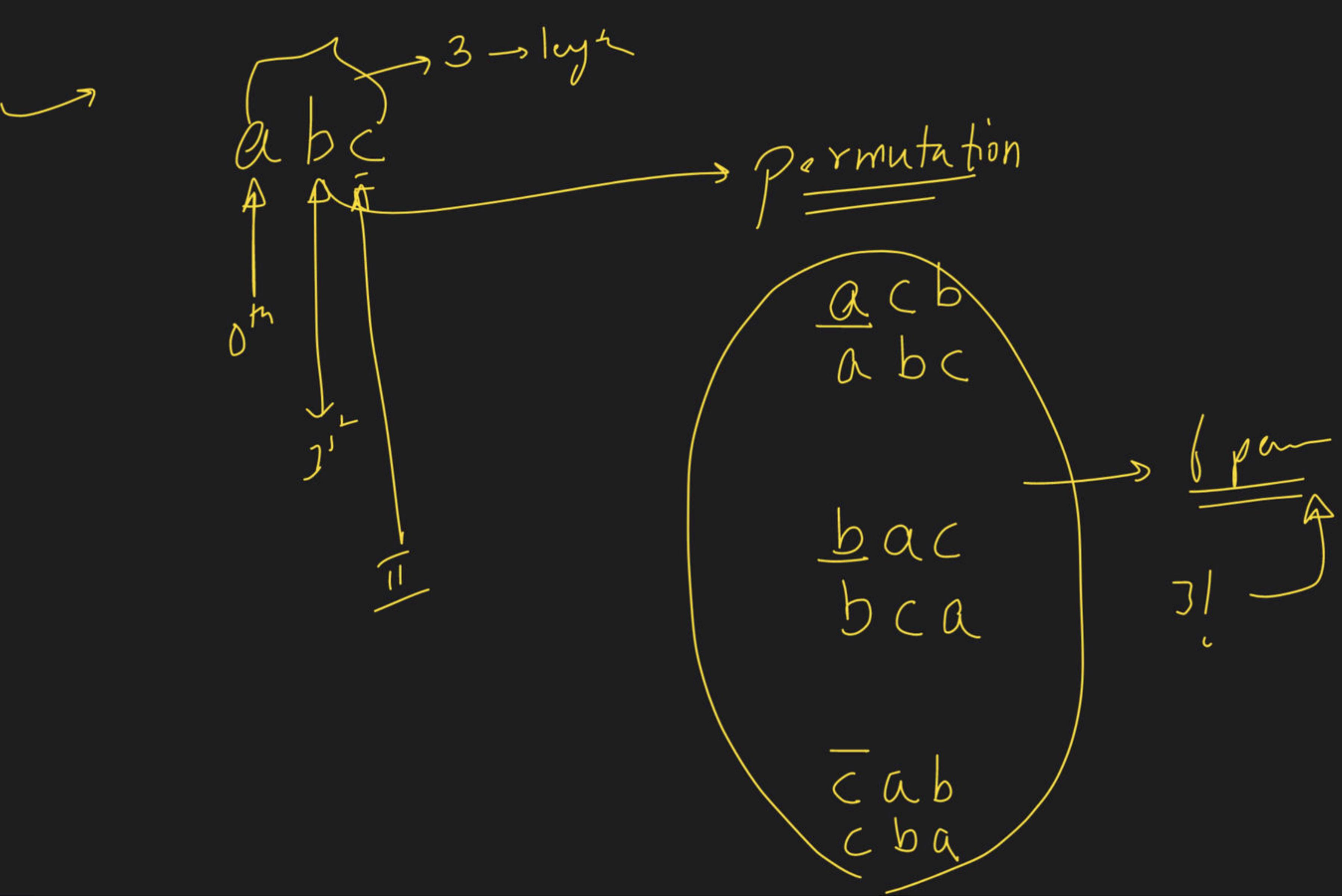
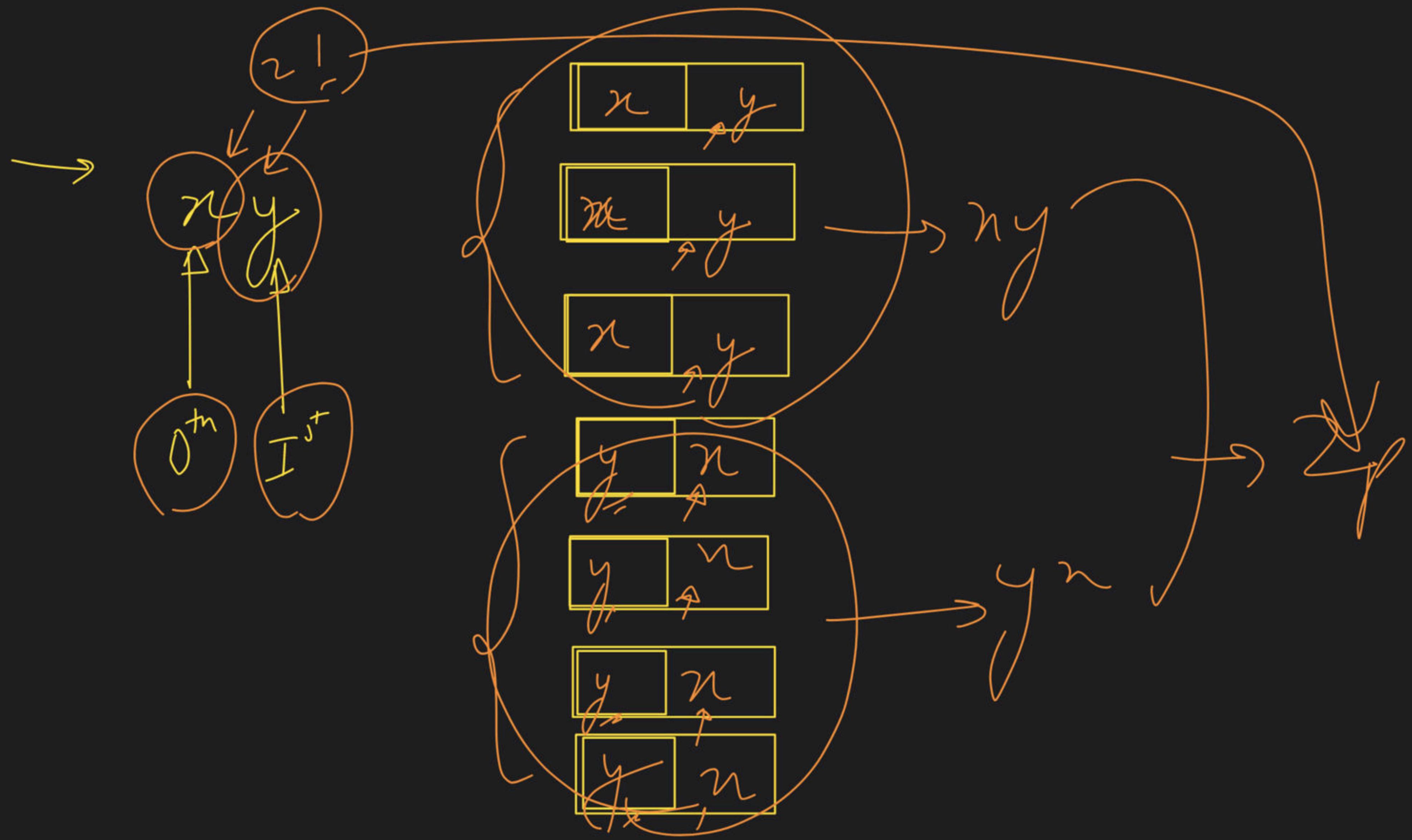
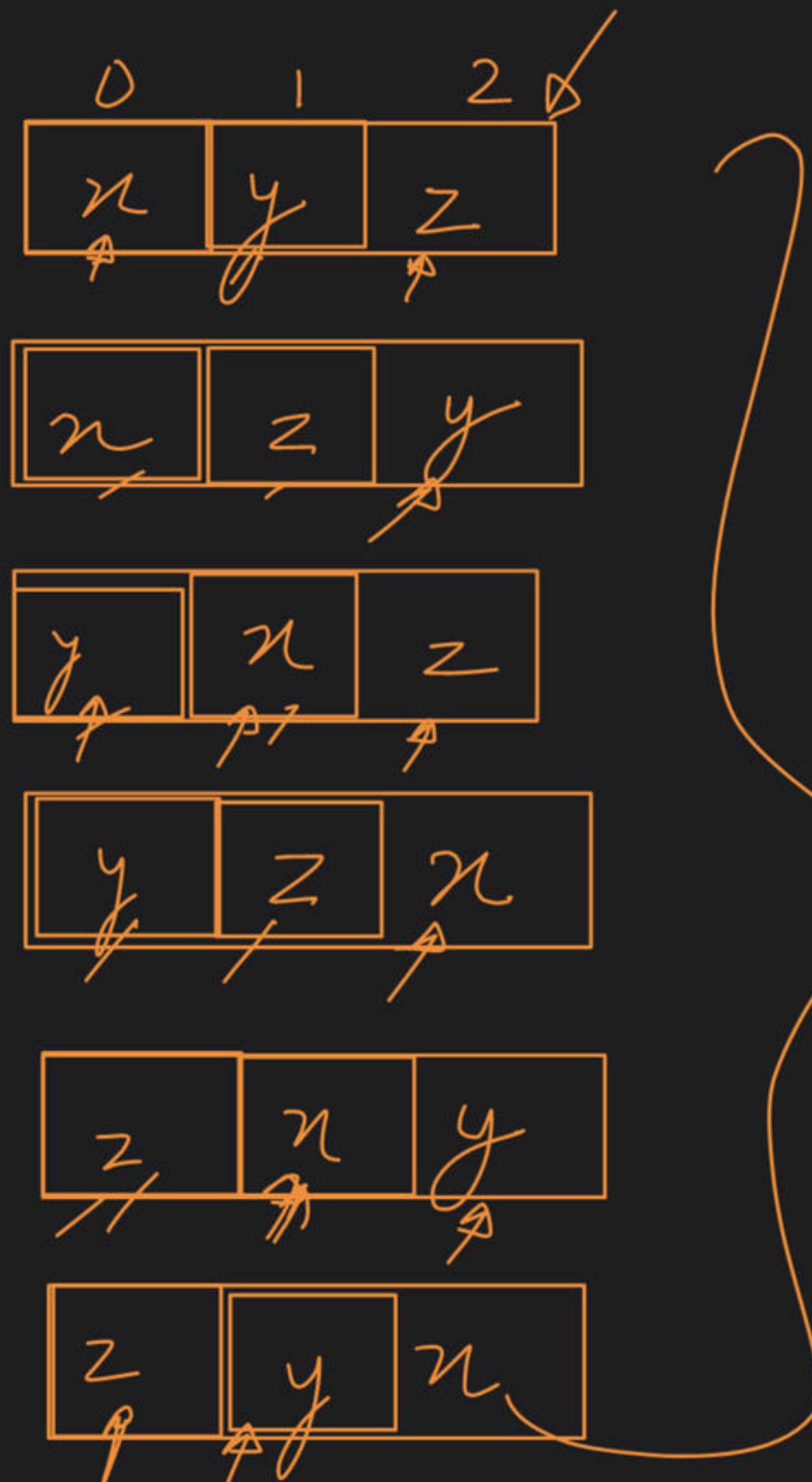


DnC && Backtracking Class - 3

Special class

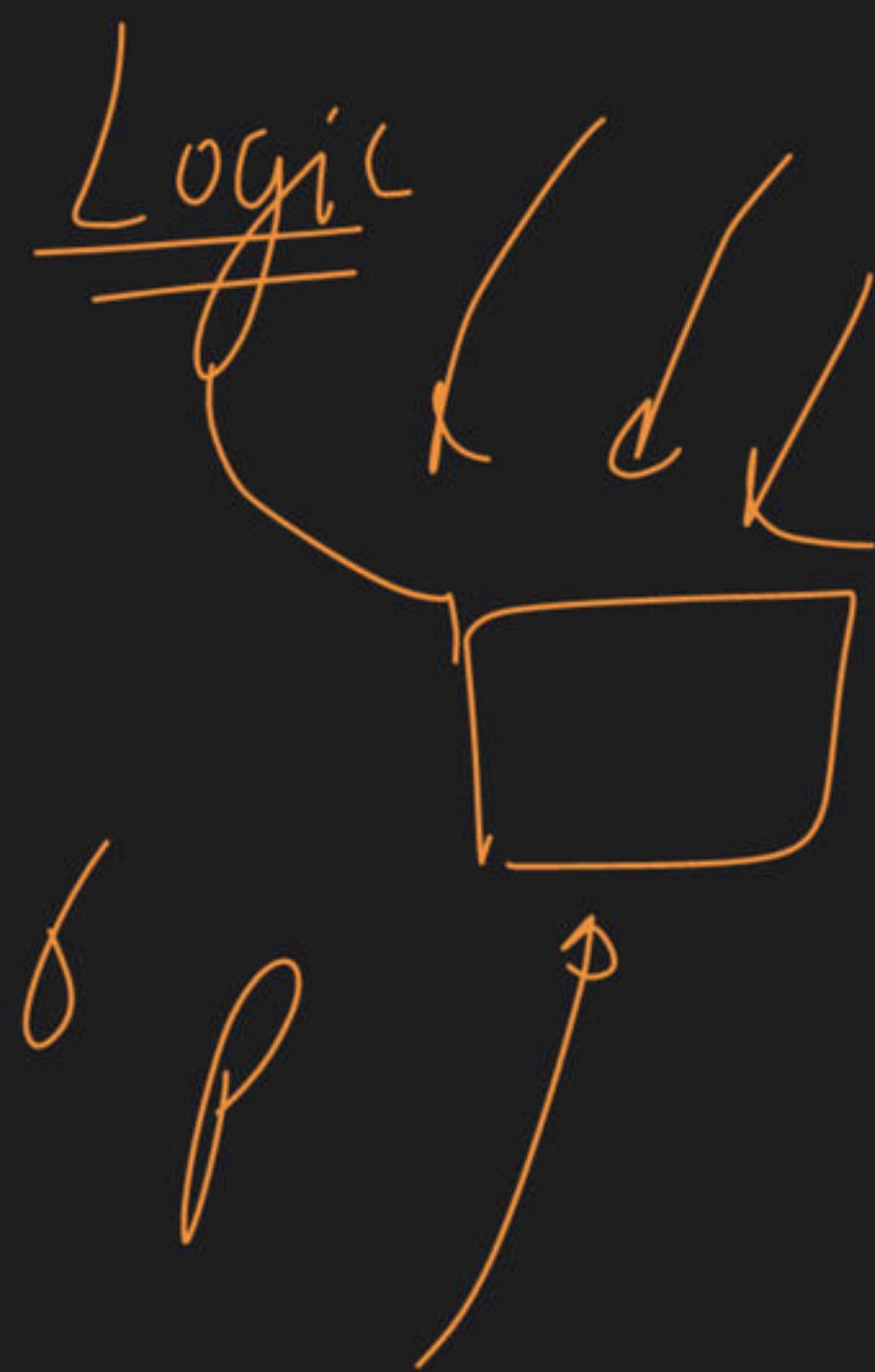


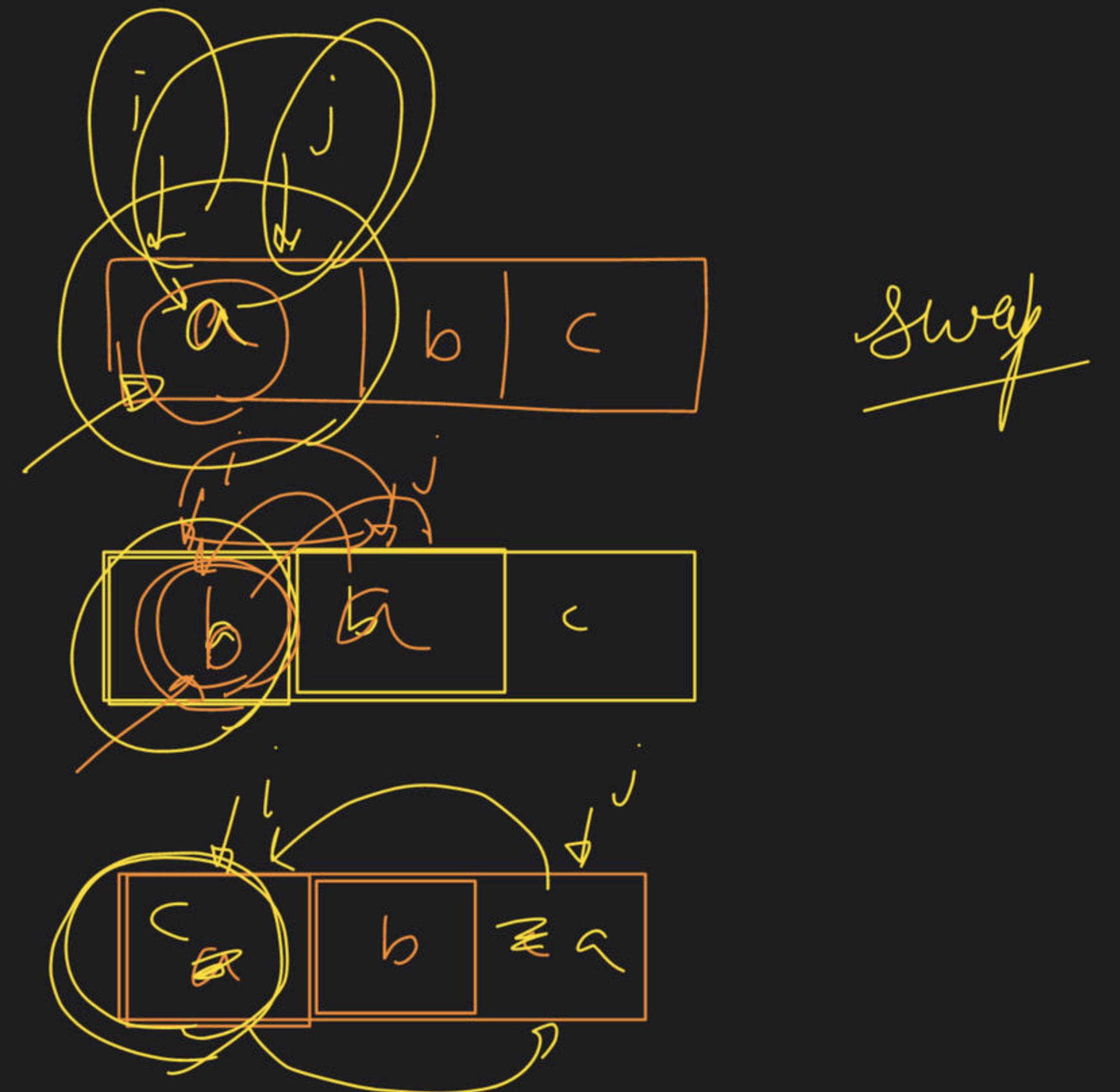


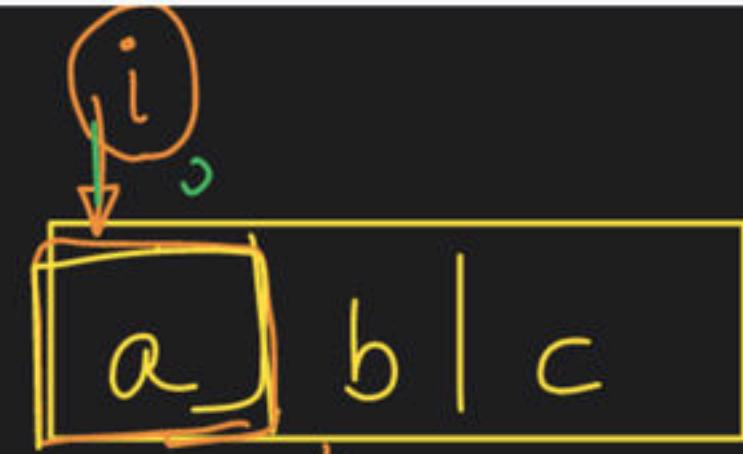


$3! \rightarrow 1$

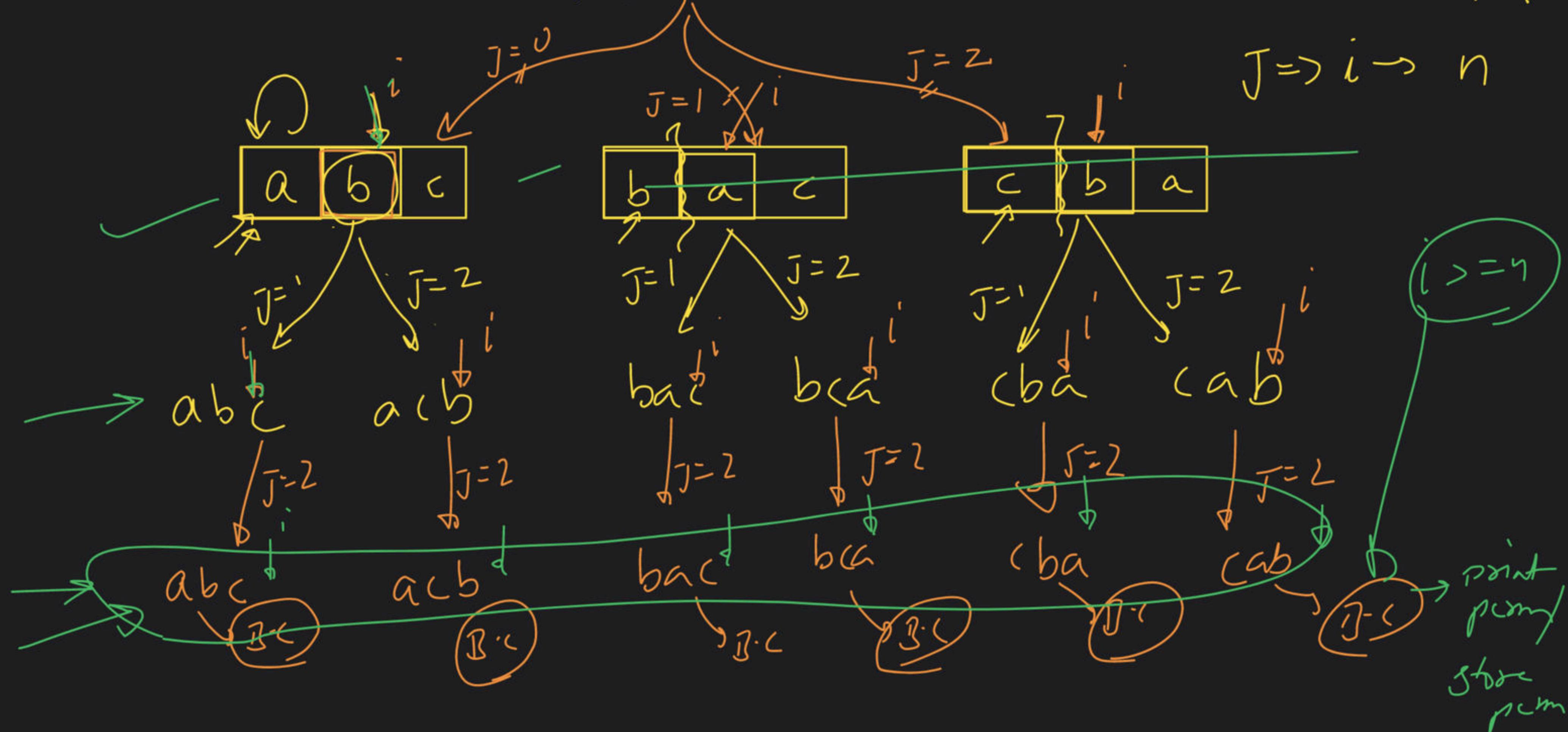
$3 \times 2 \times 1 \rightarrow 6$

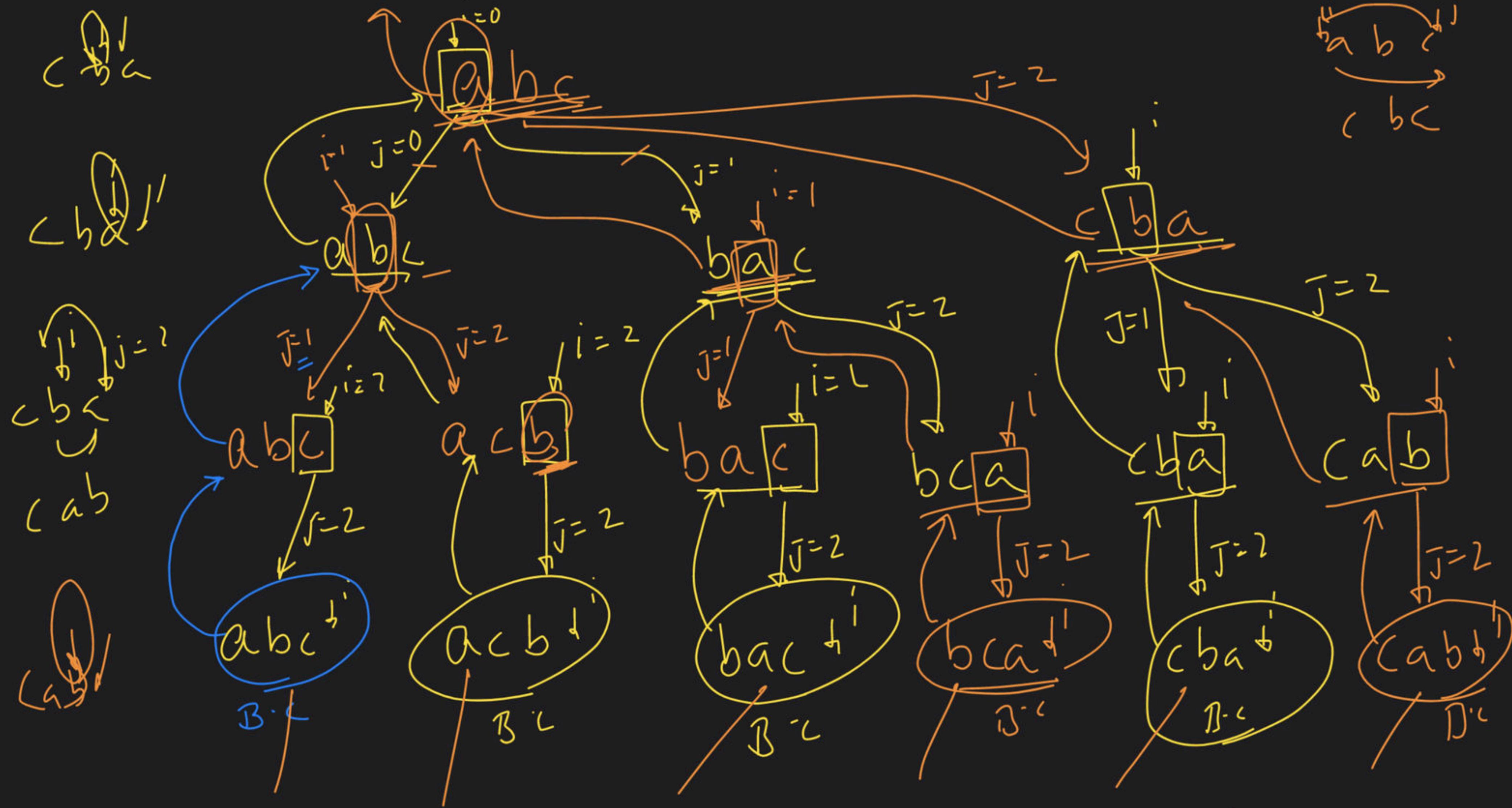






$J \rightarrow \emptyset \rightarrow h \times$

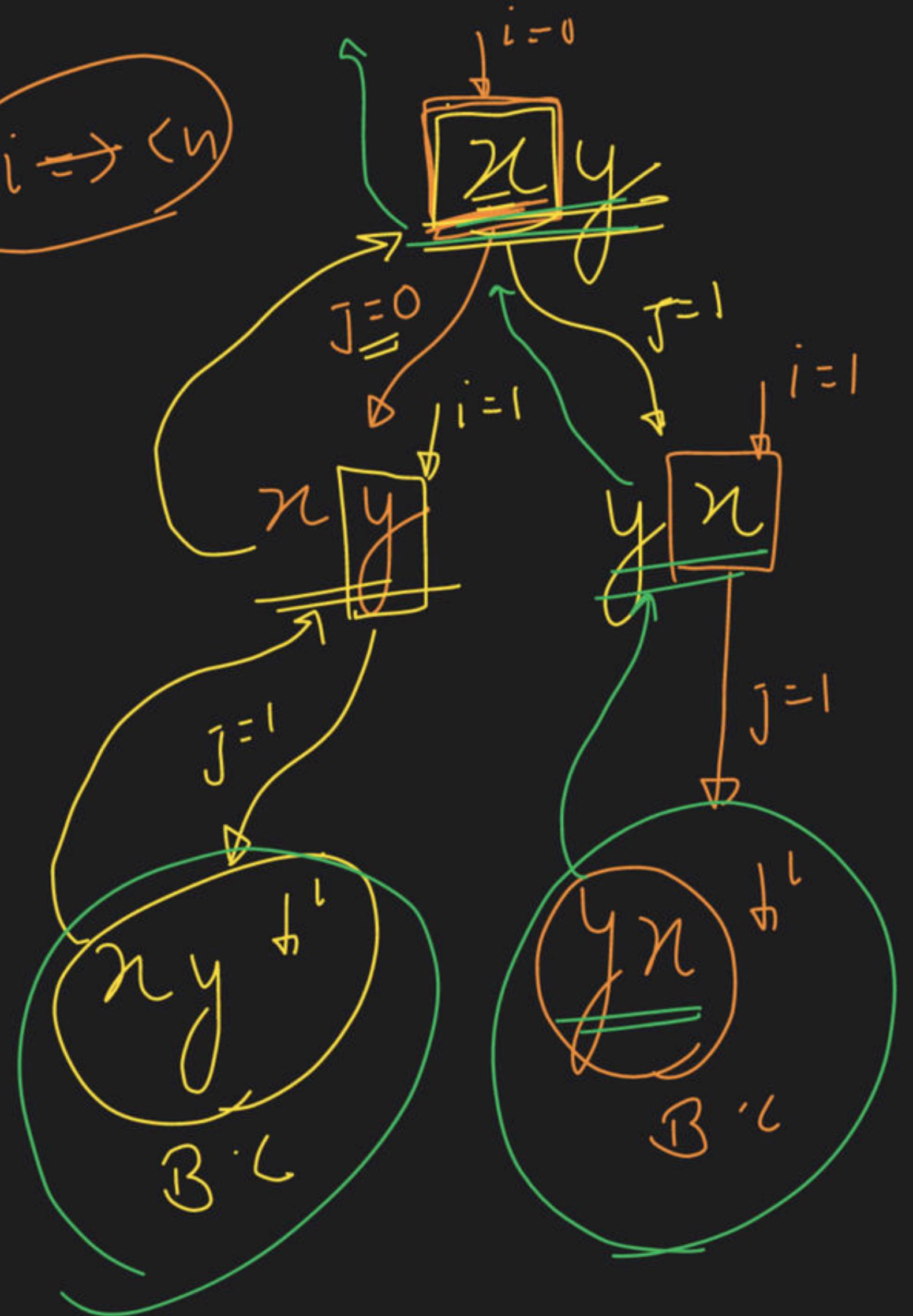




ny

$j \rightarrow i \Rightarrow ny$

$i \downarrow j$
 ny
 y^n



ny
 y^1, y^n

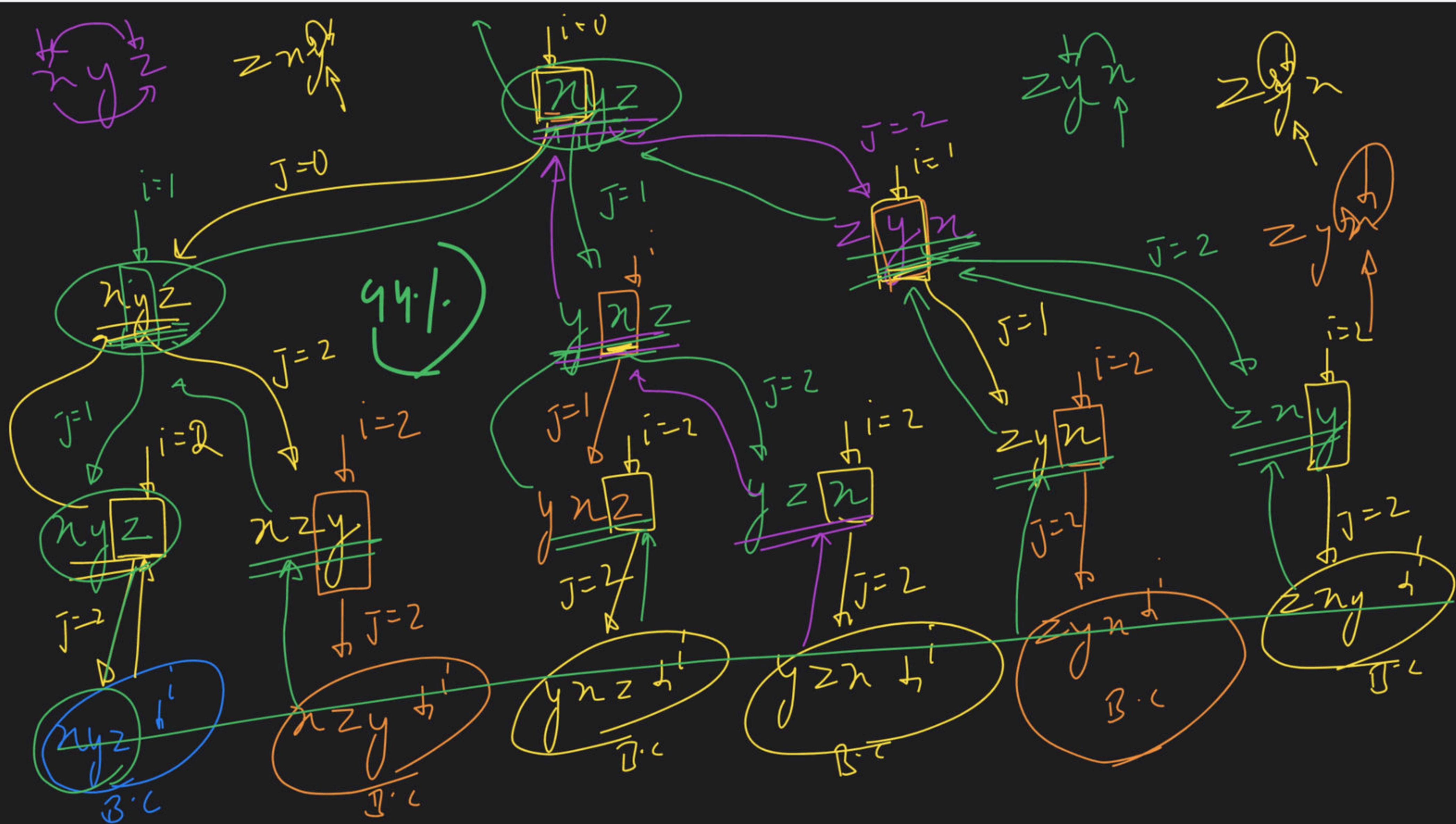
$y = ny$

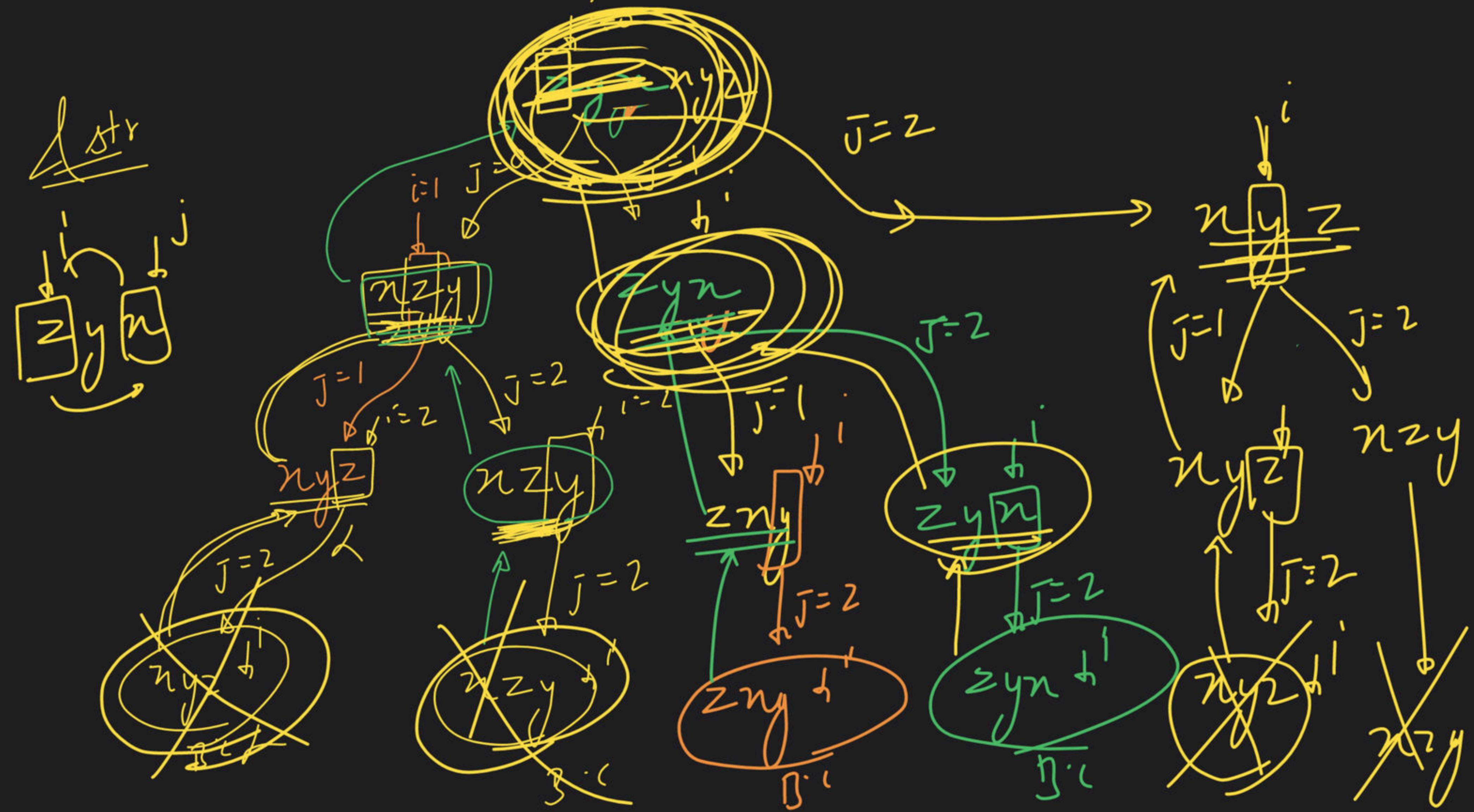
$ny = ny$

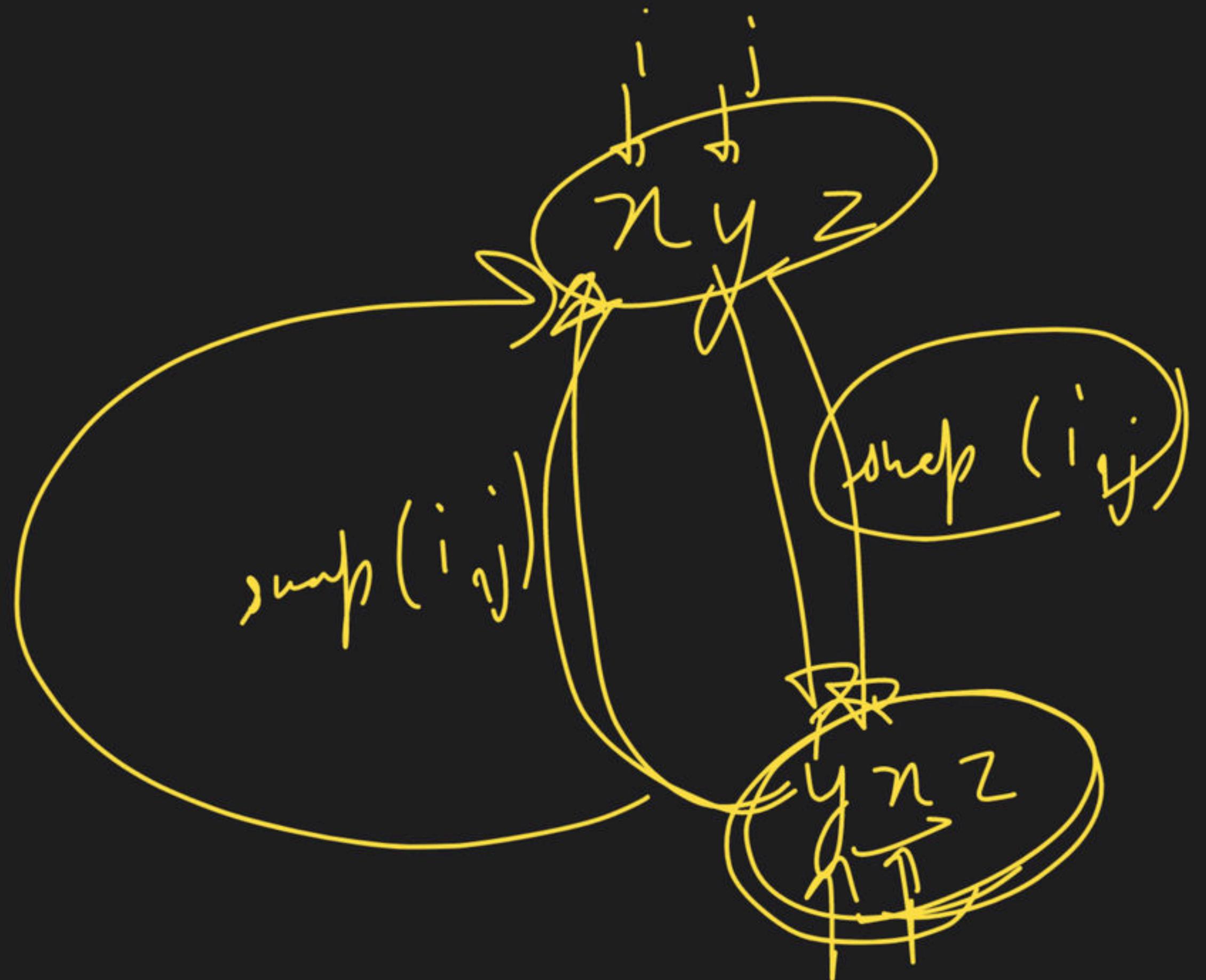
y^1, y^n

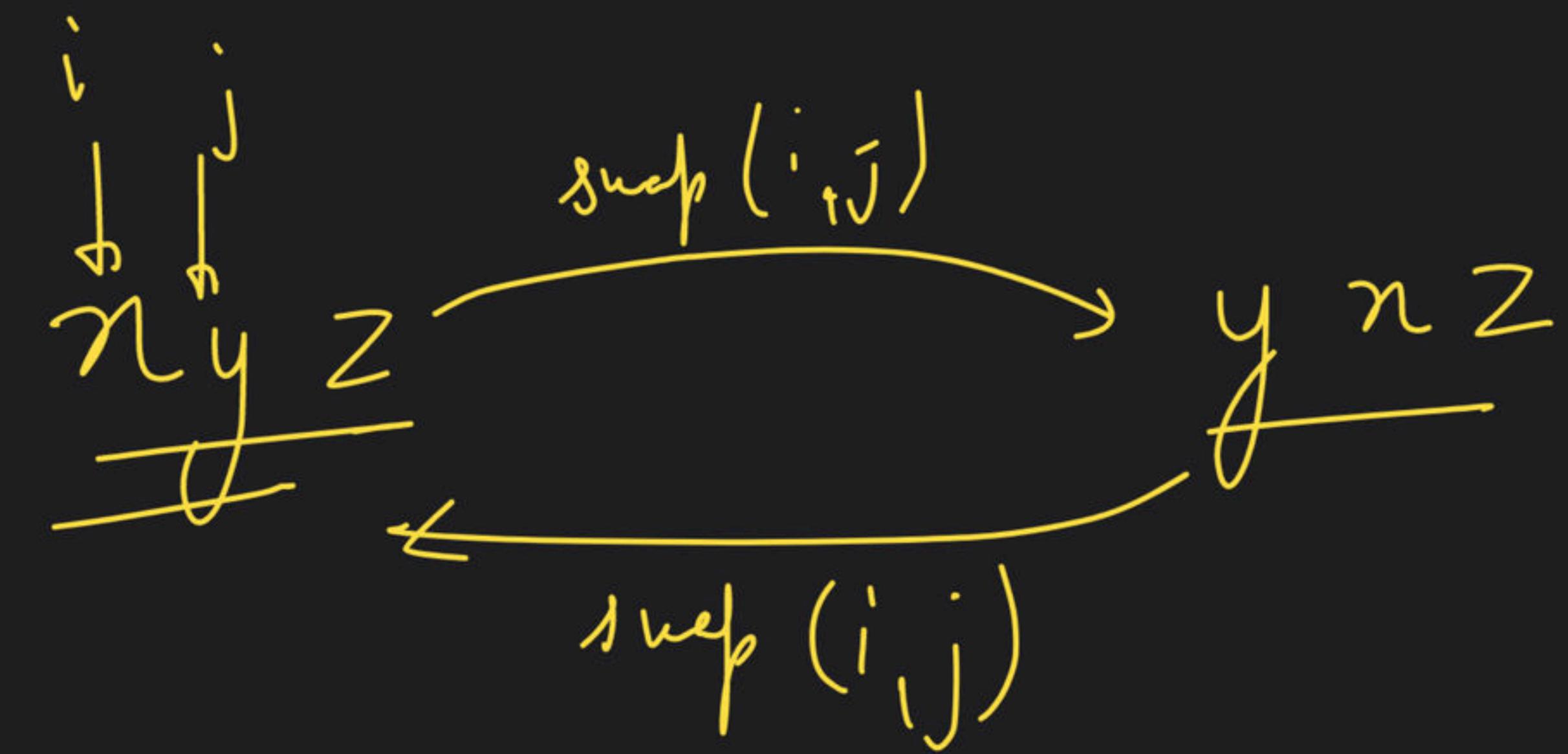
y^n

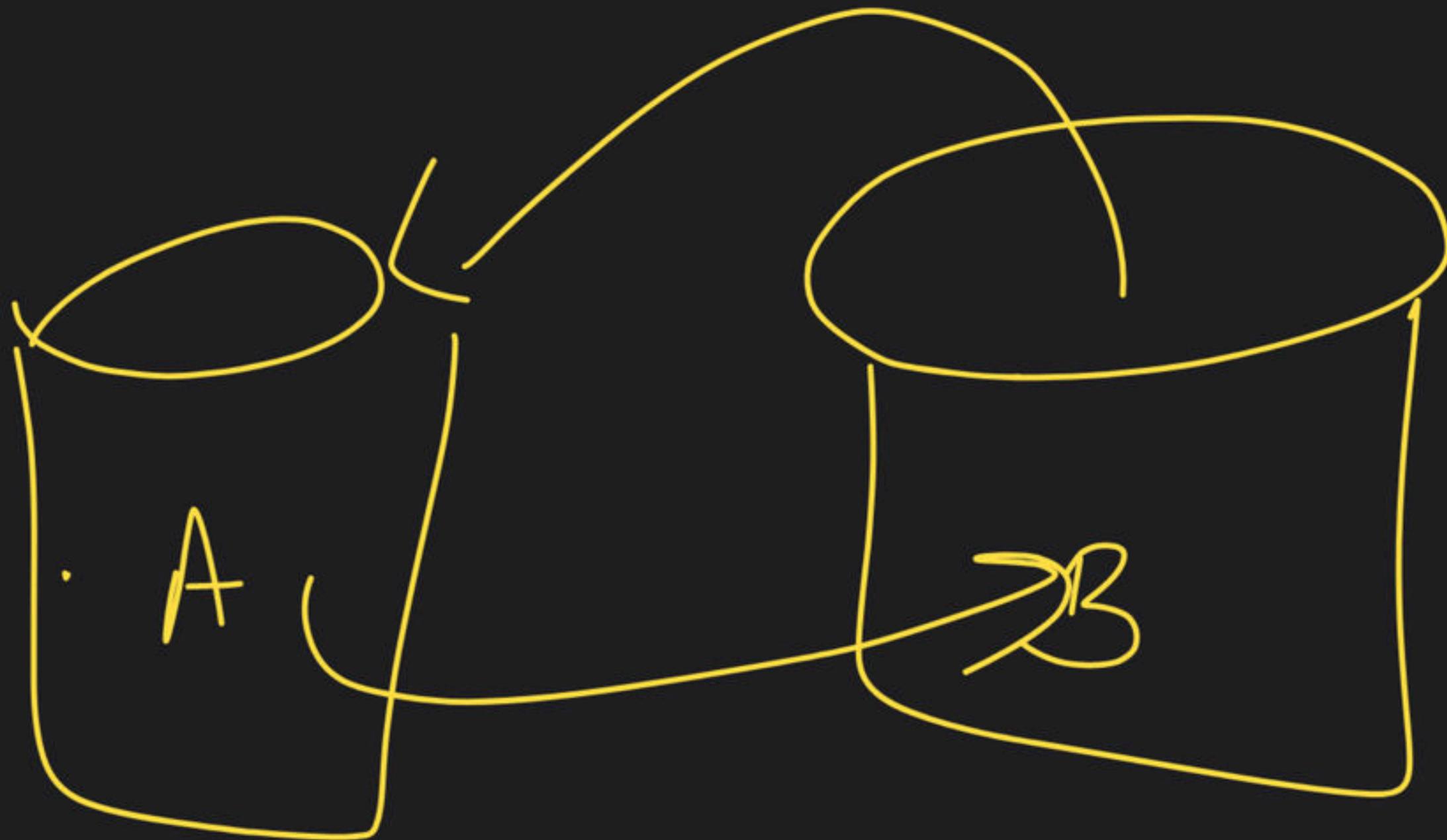
$y^n \rightarrow y^n$





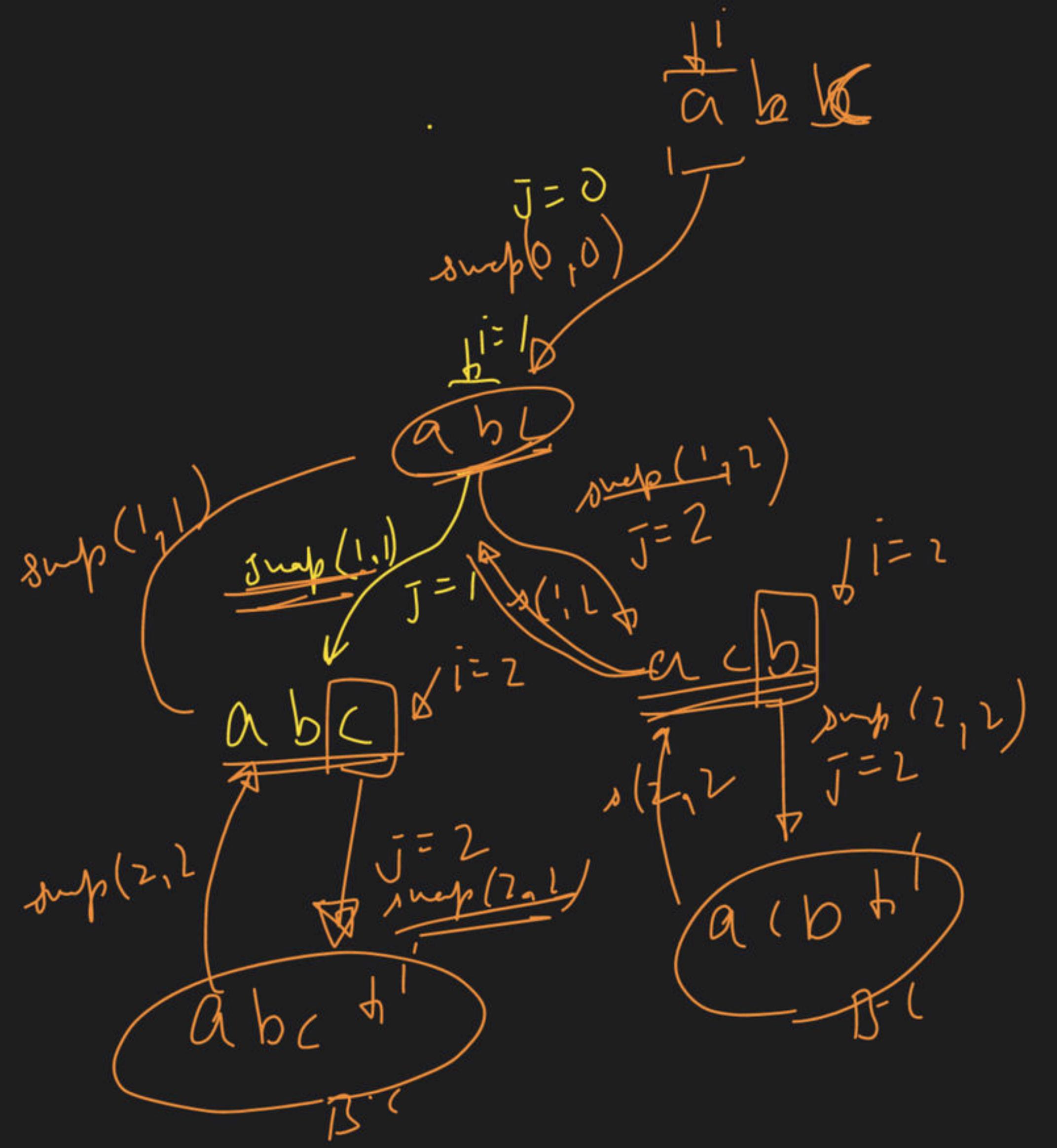


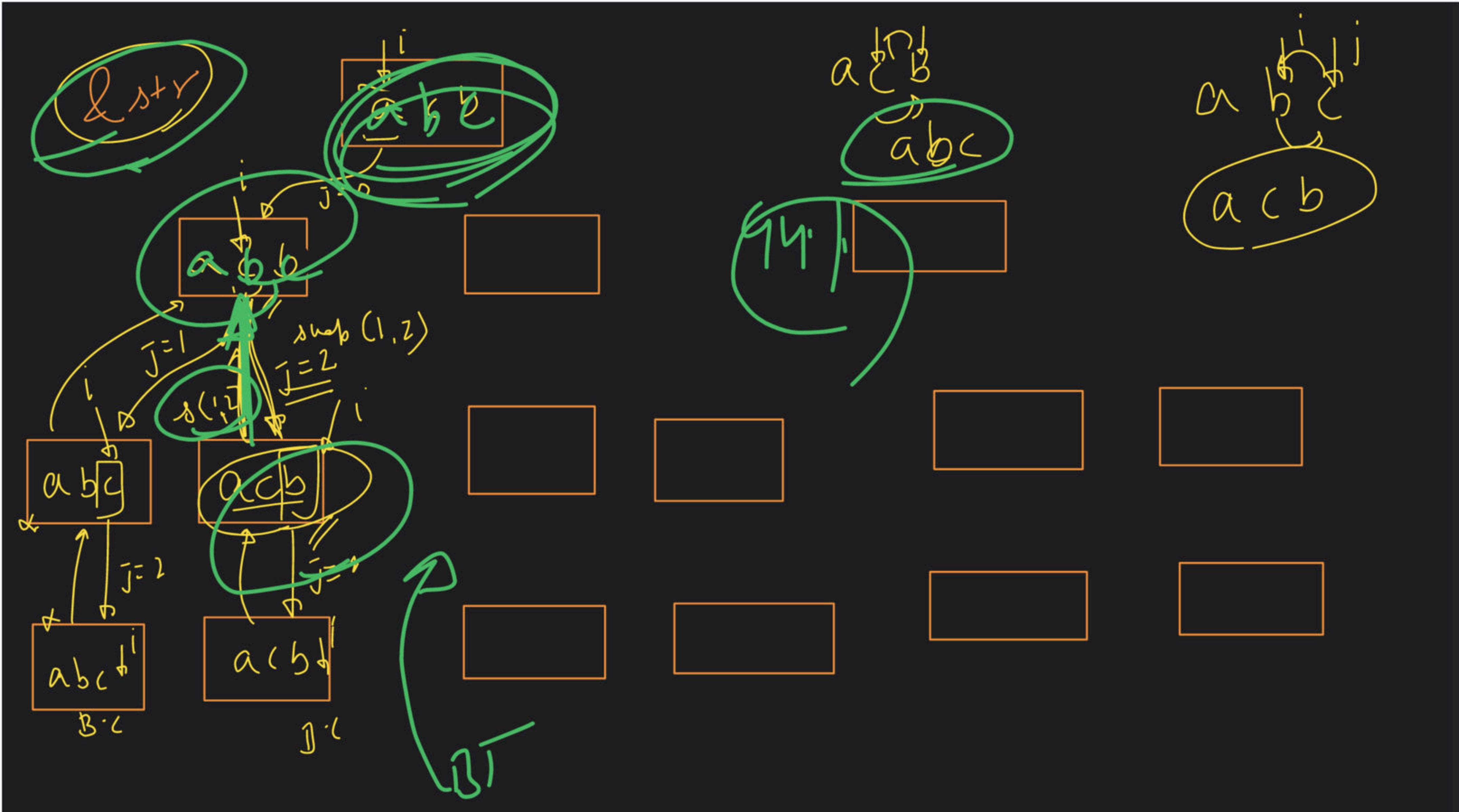




B A

A B





for (

i

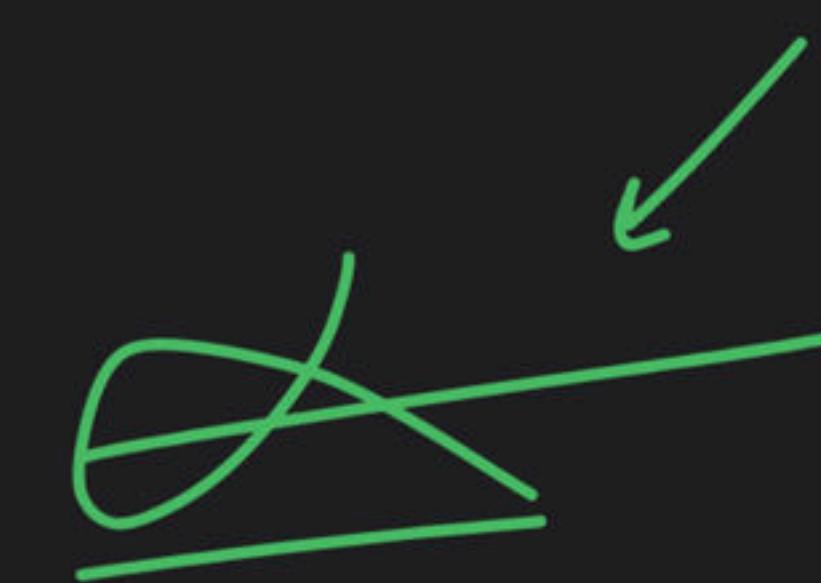
sup

bai

solve

for (

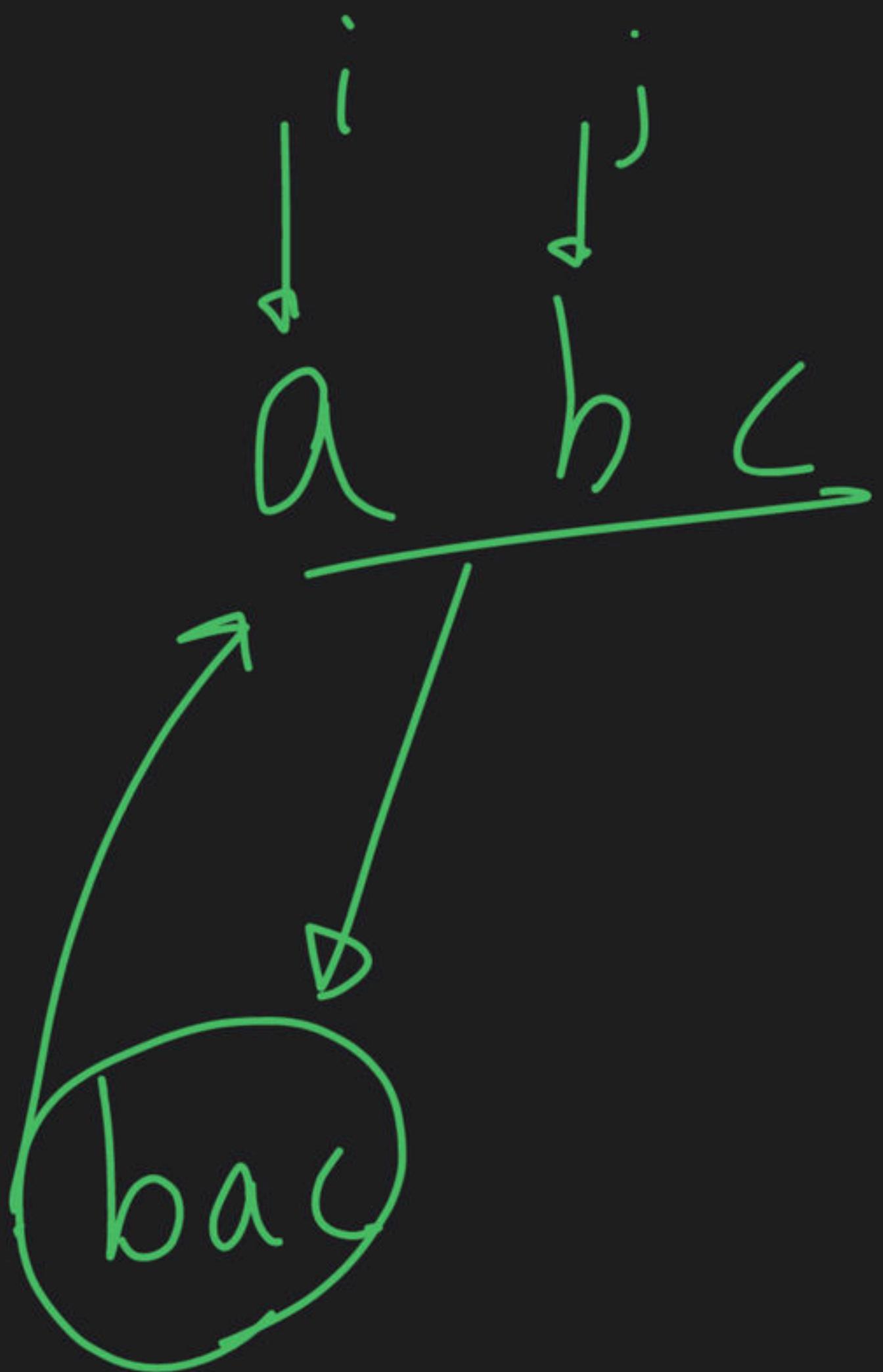
)



a b

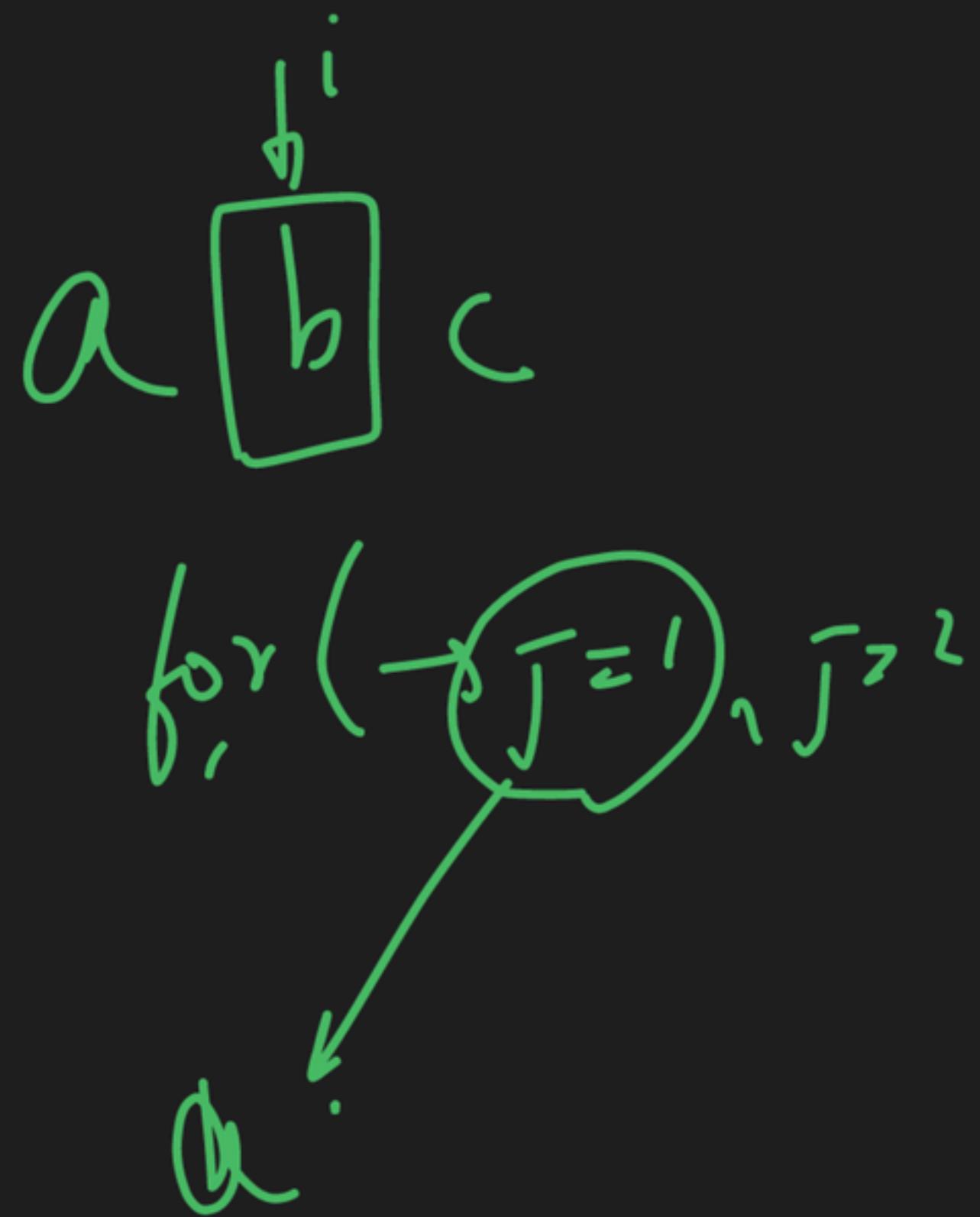
i j
bai

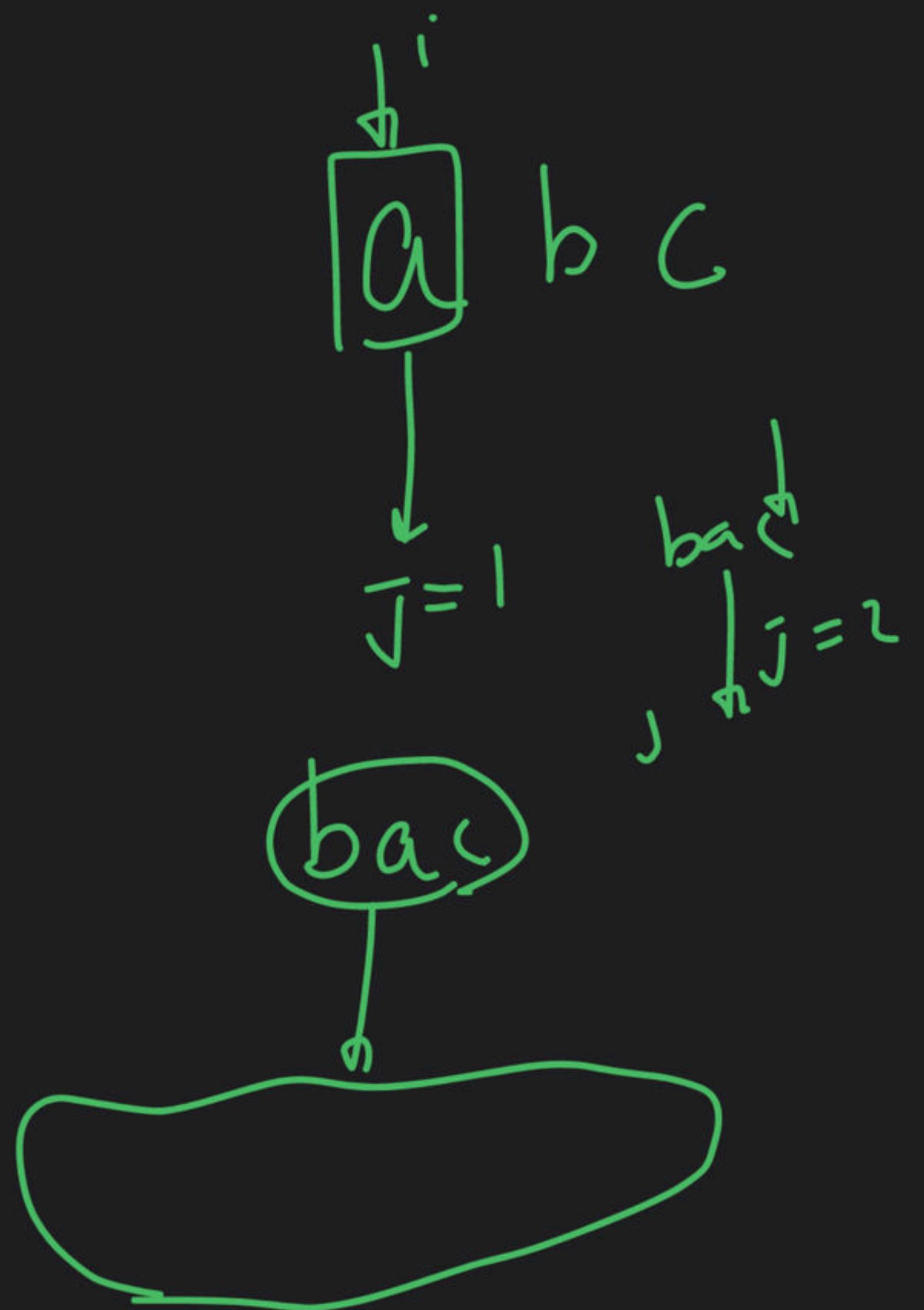
20

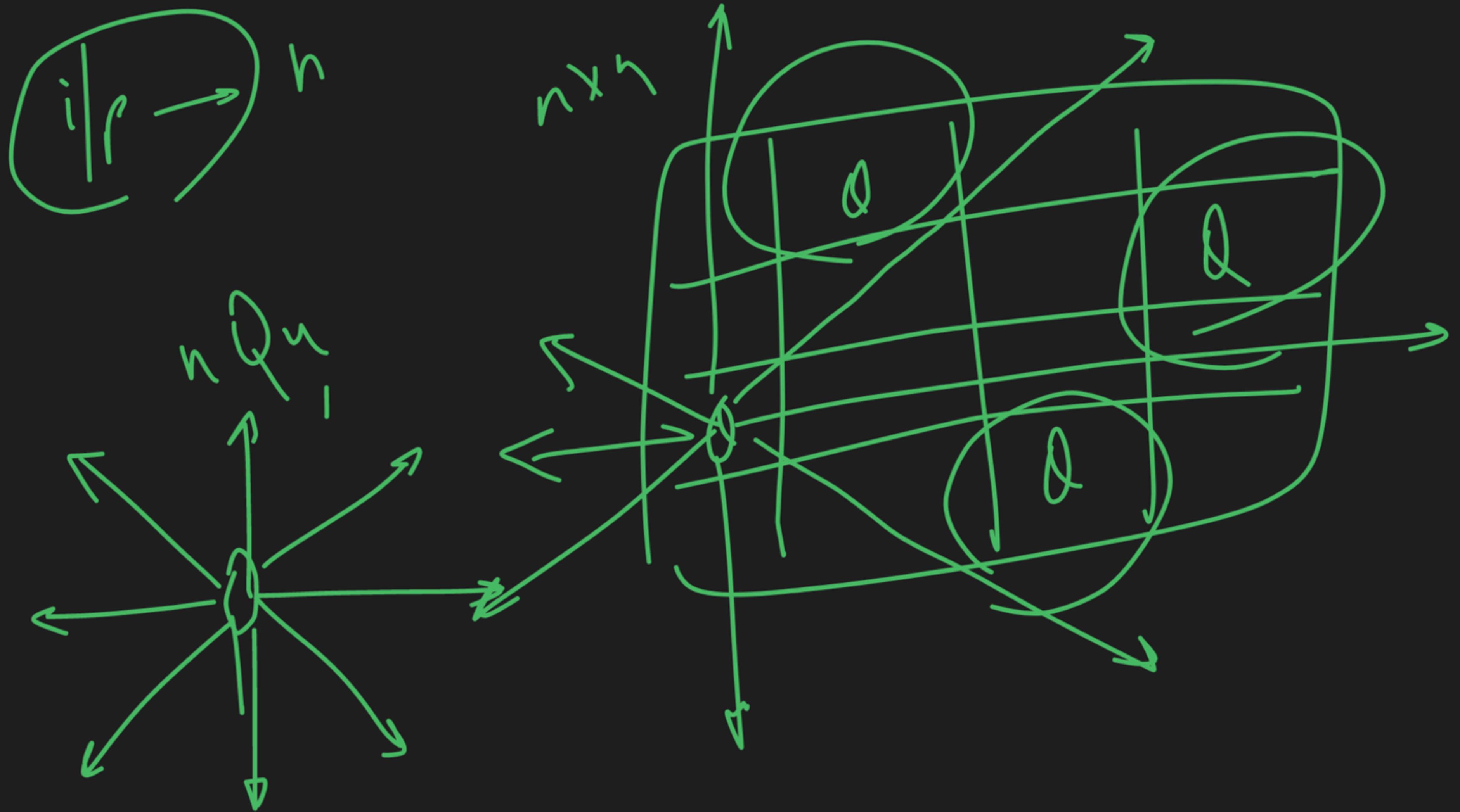




for



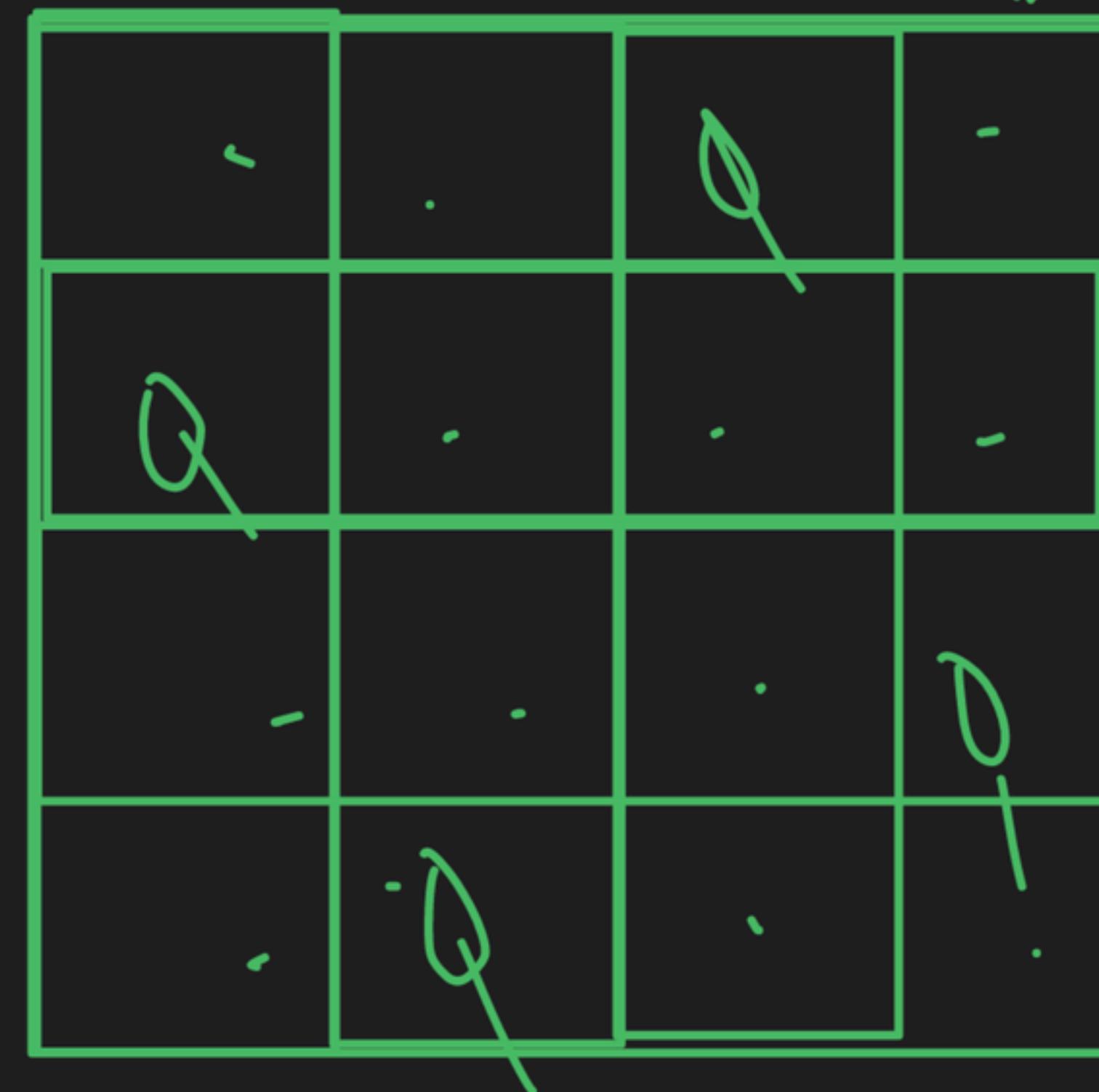




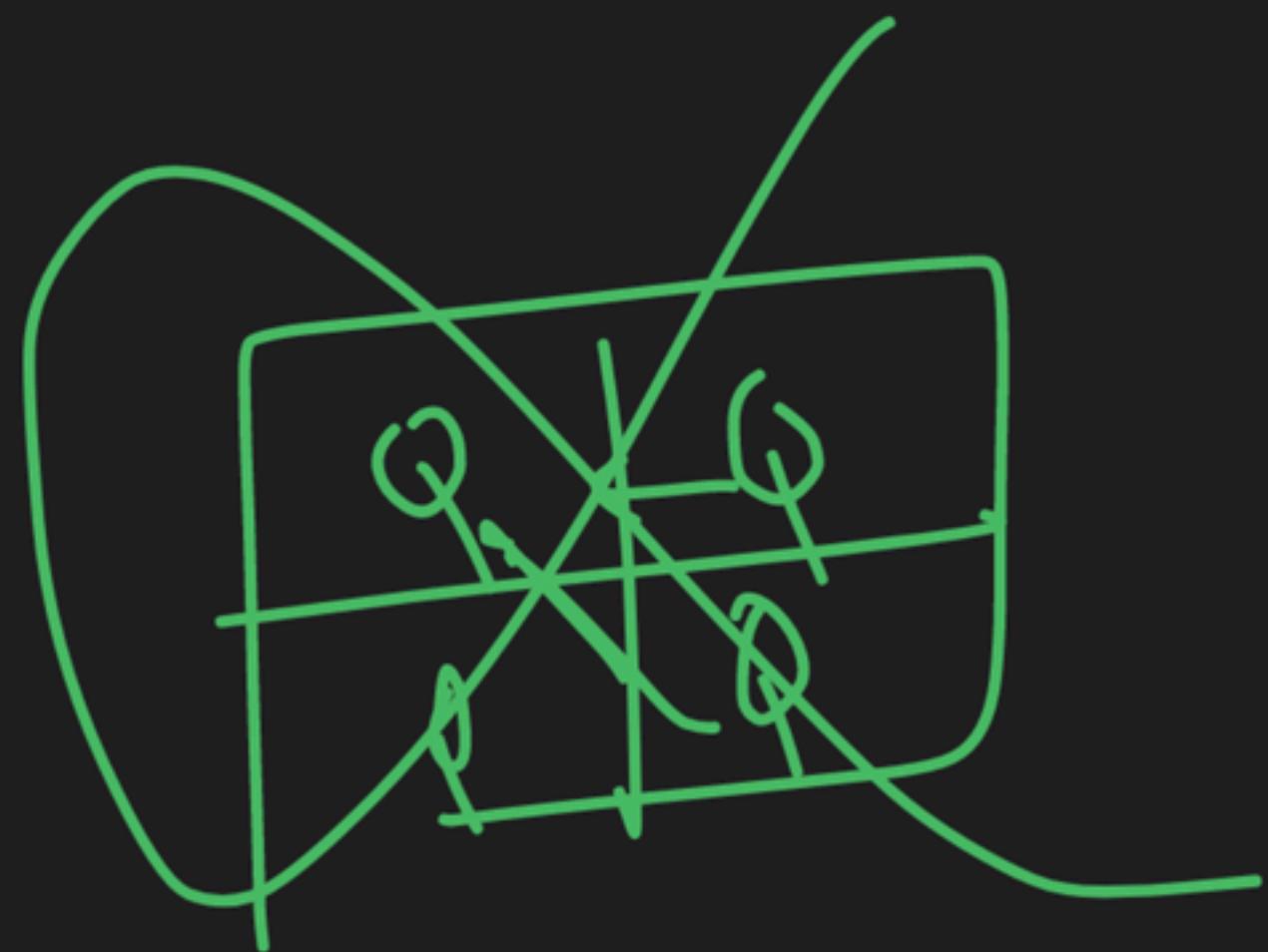
$$n=4$$



𠂔 𠂔 𠂔 𠂔 𠂔



$n=4$



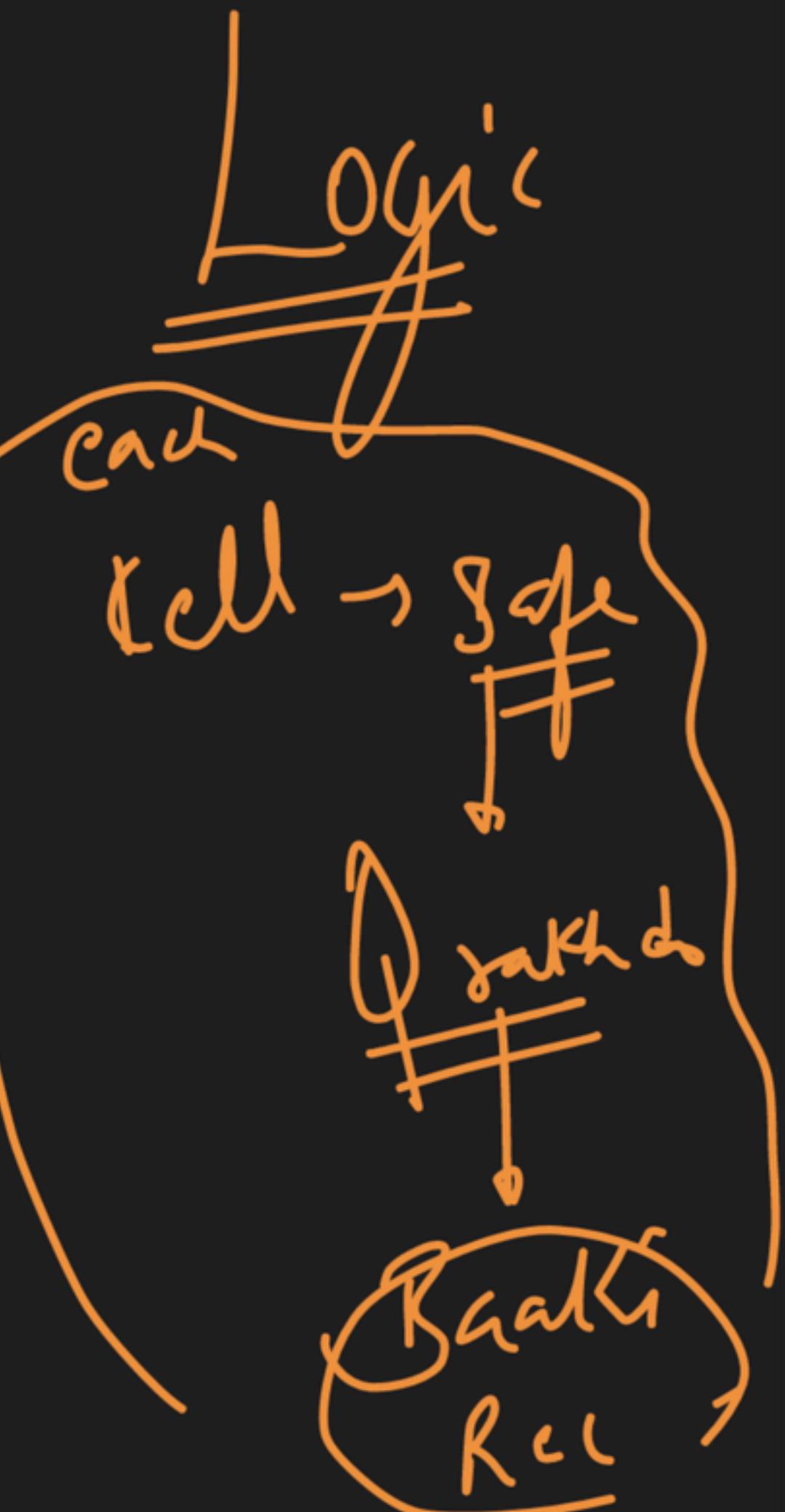
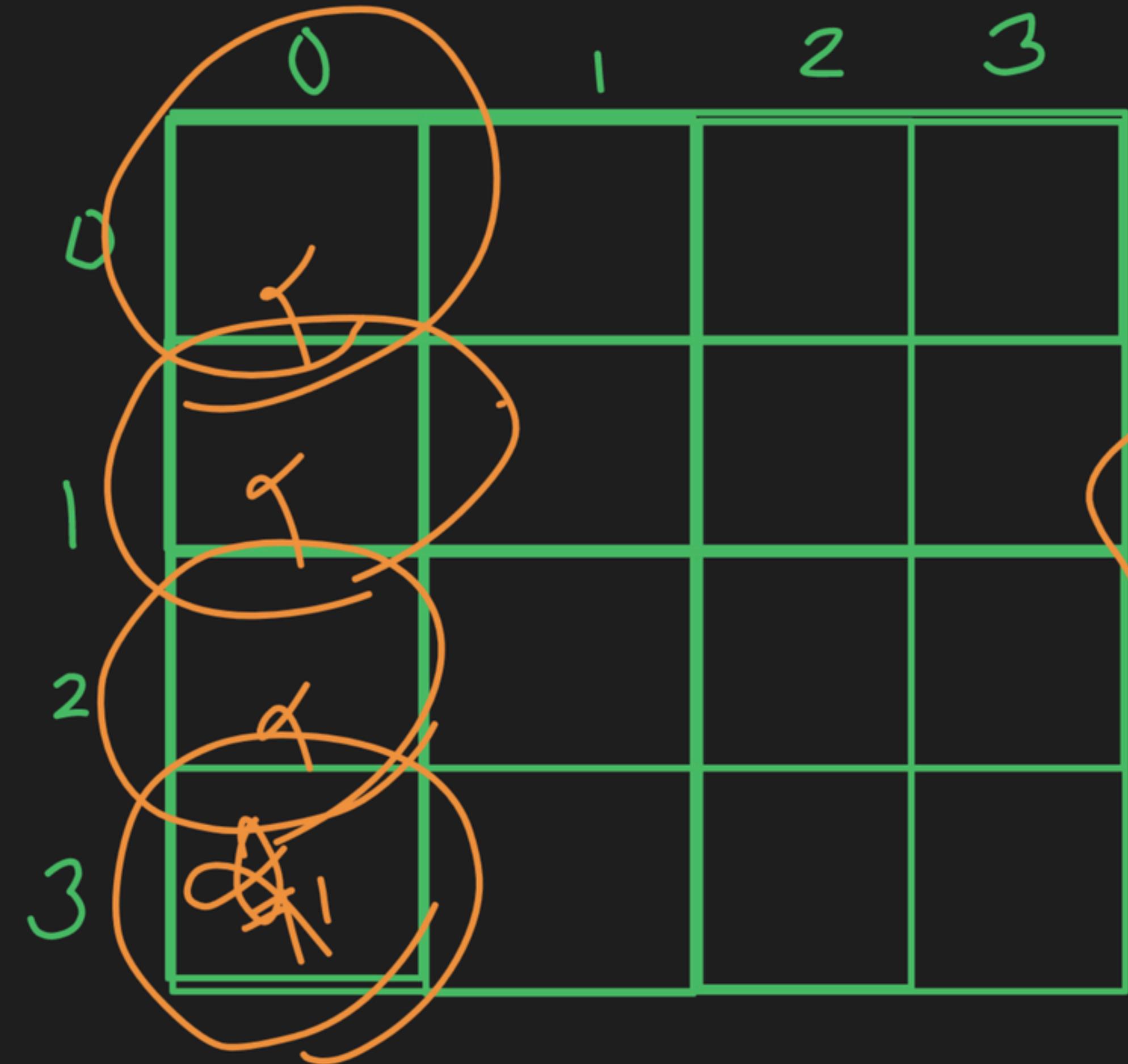
-

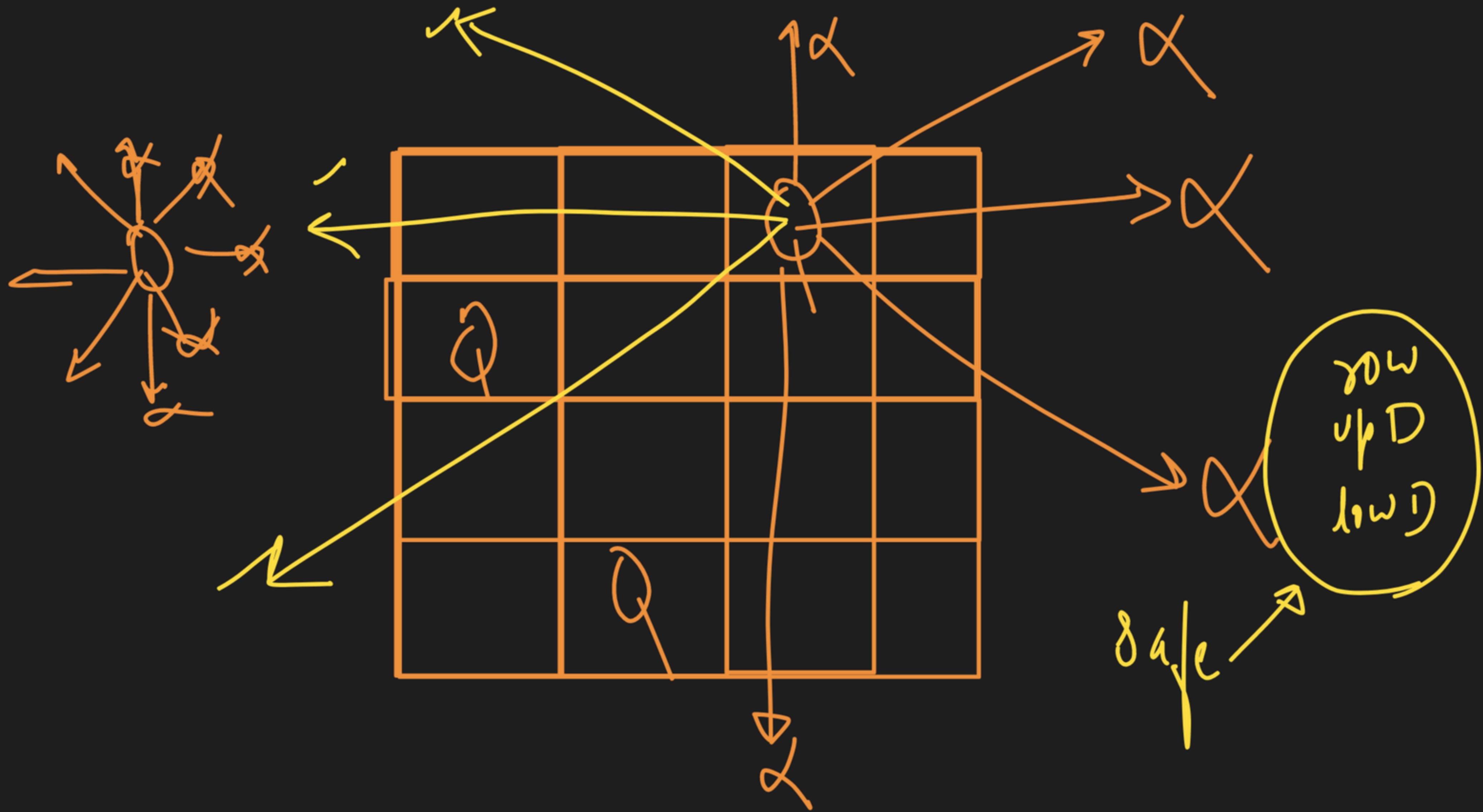


$n=4$

Q_1
 Q_2
 Q_3
 Q_4

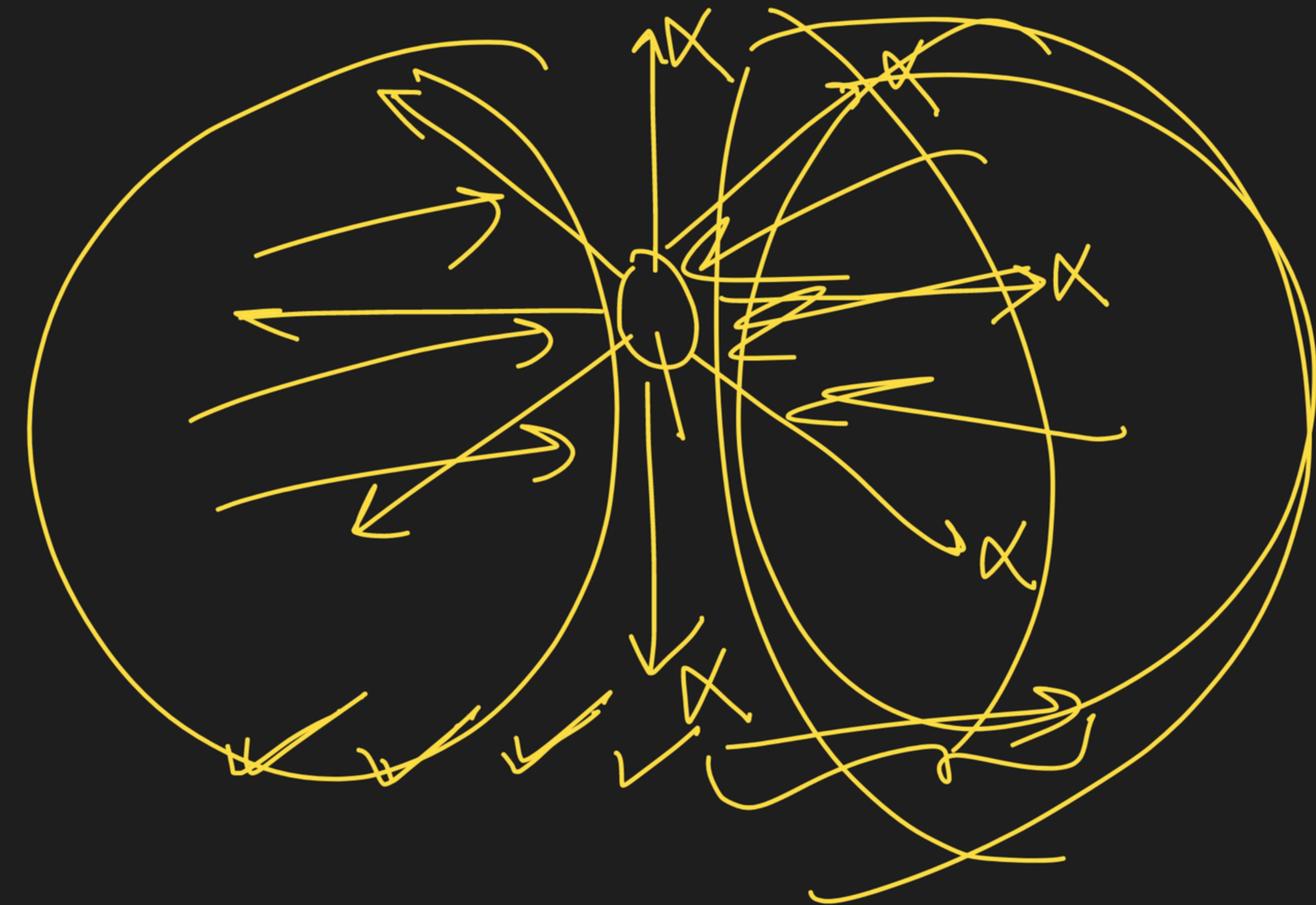
5

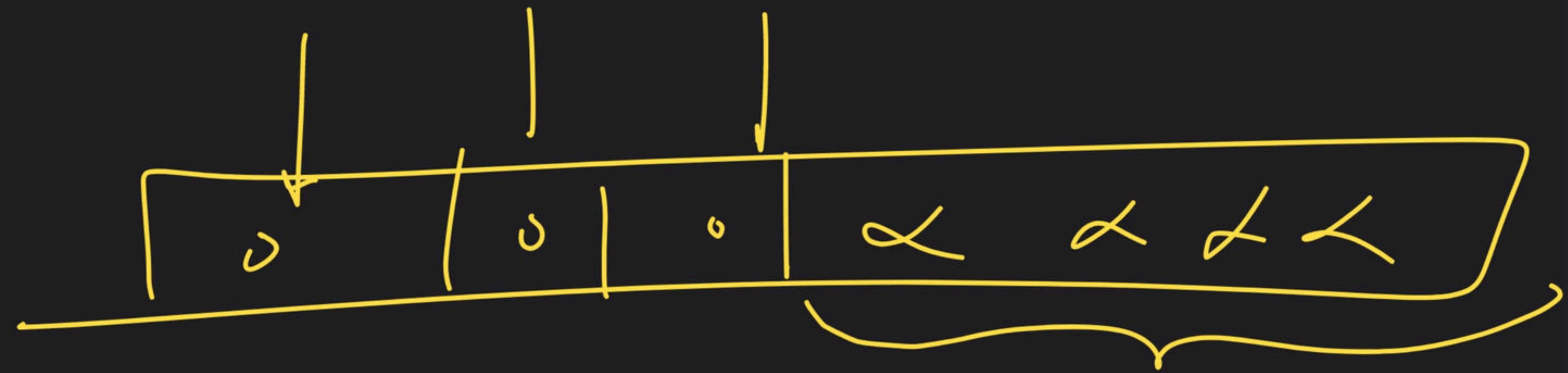


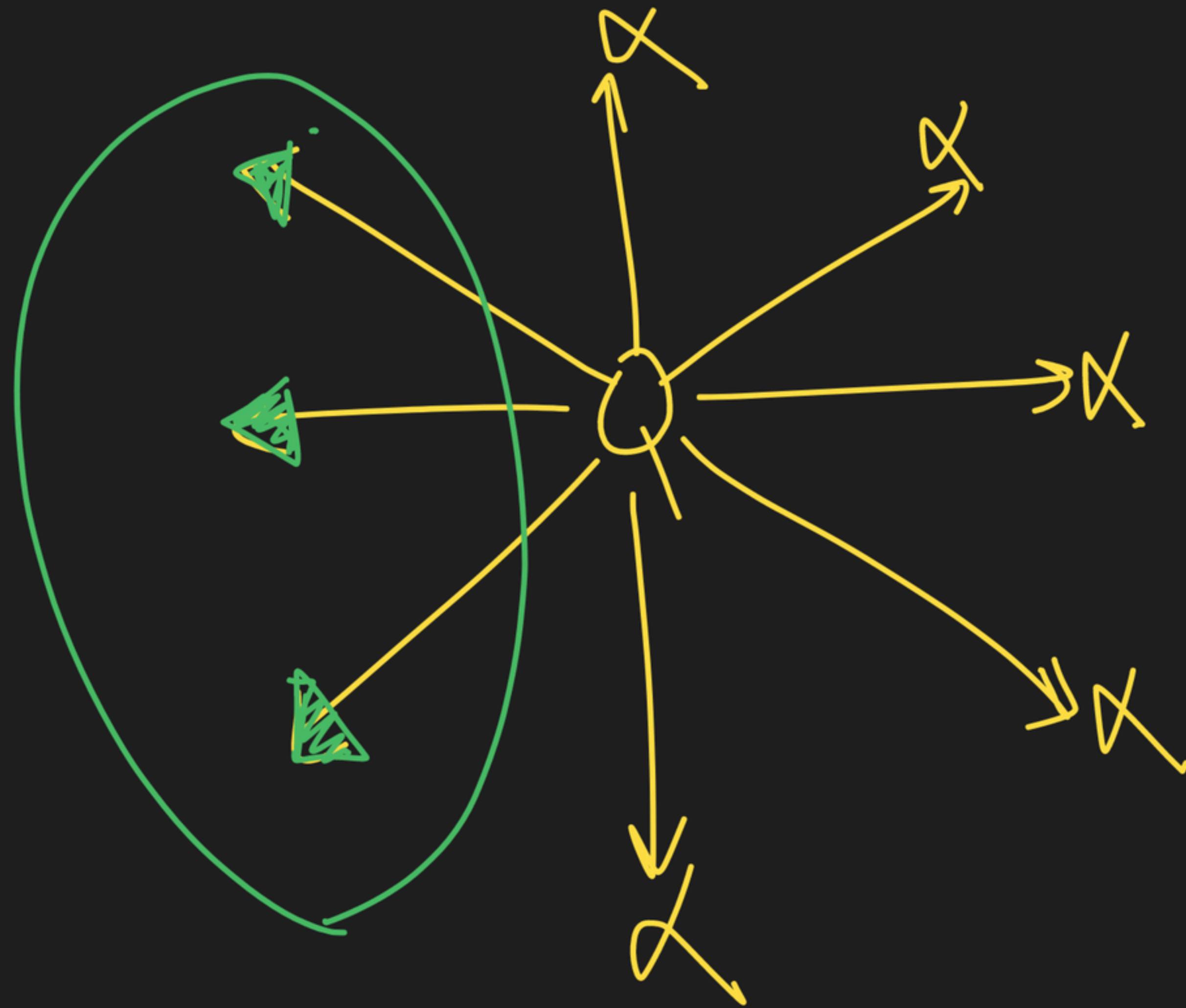


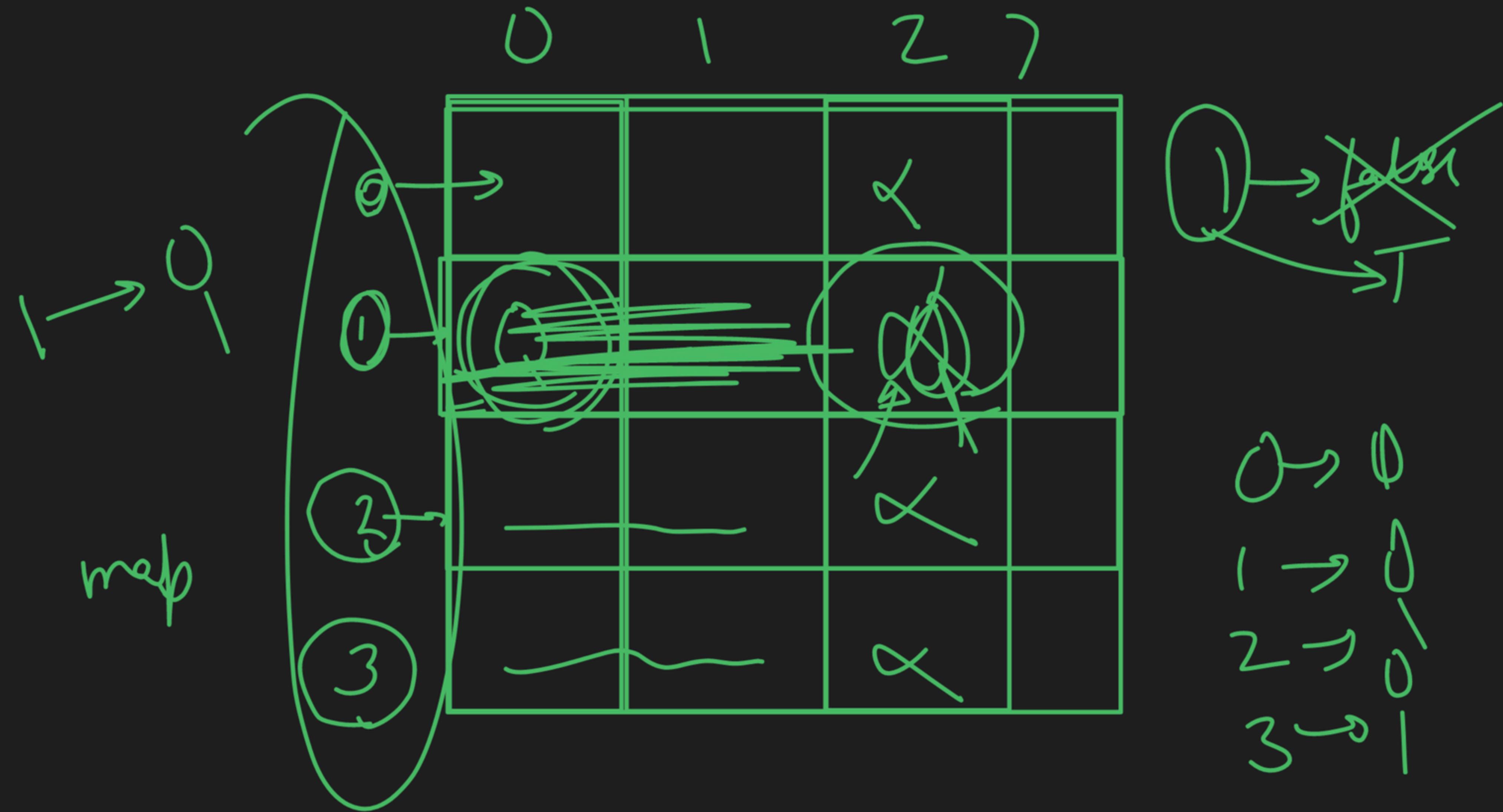
~~Safe~~

	0	1	2	3
0				
1				
2				
3				









row check

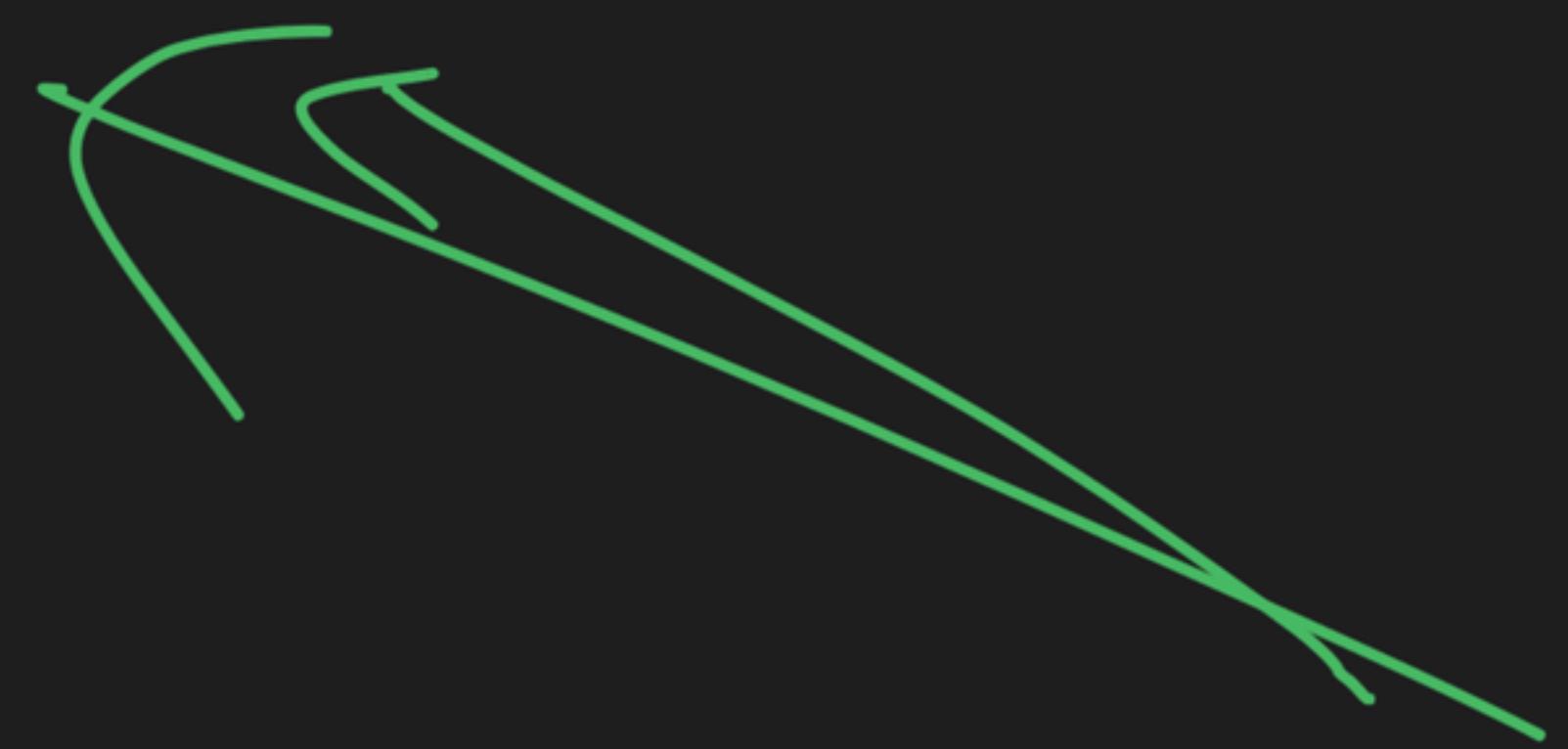
map < rowIndex -> boolean >

place -> index -> T

safe -> check index -> T/F

you check

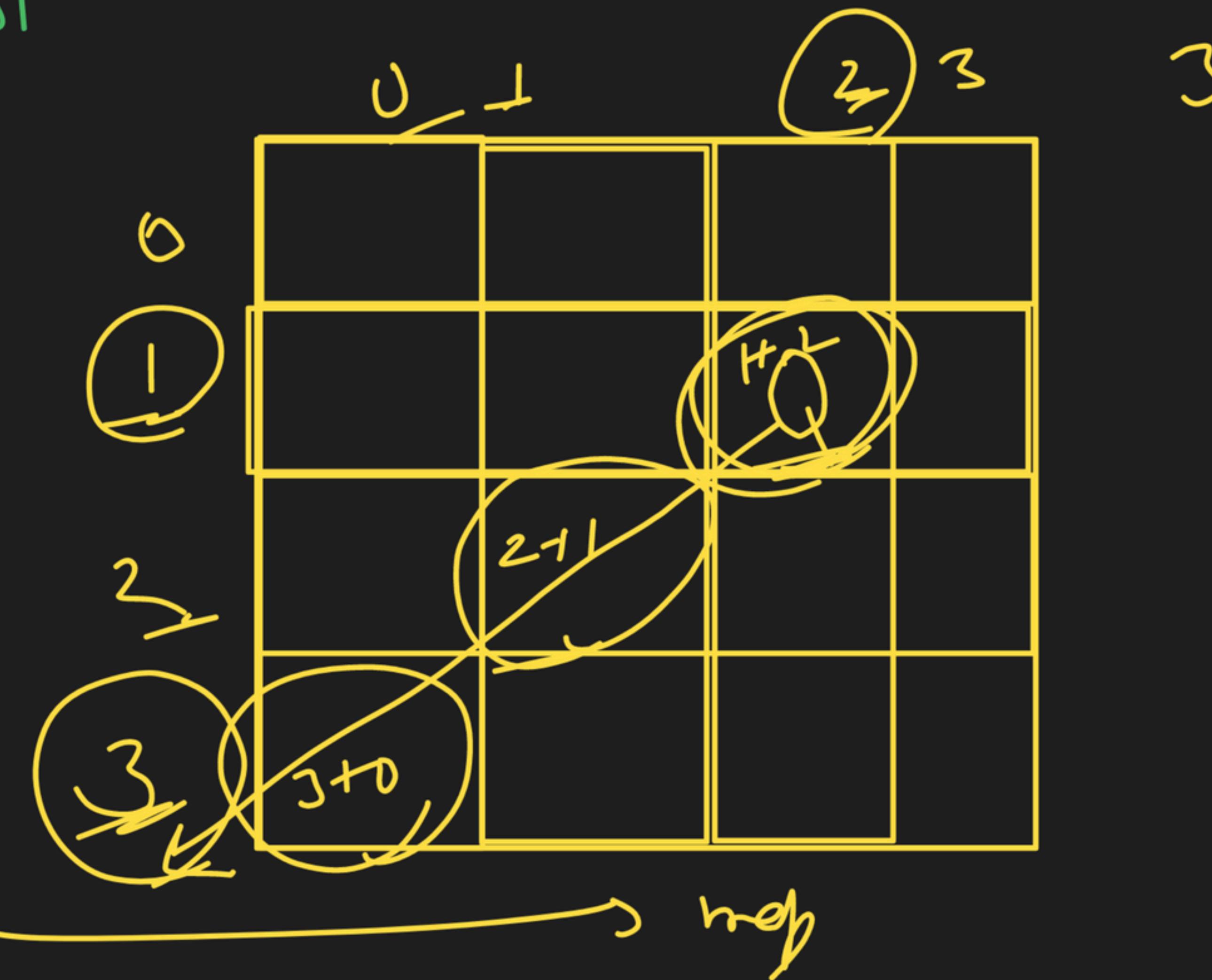
$O(1)$ → safe/not
safe



Lower Diagonal

Rowinden +
Colinden

$$\gamma + \zeta$$



Lower Diag hd

map < row+col → boolean >

Upper Diagonal

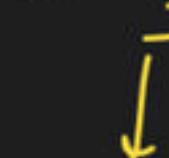
$i - 0$



0	1	2	3
0	1		
1		1	
2			1
3			

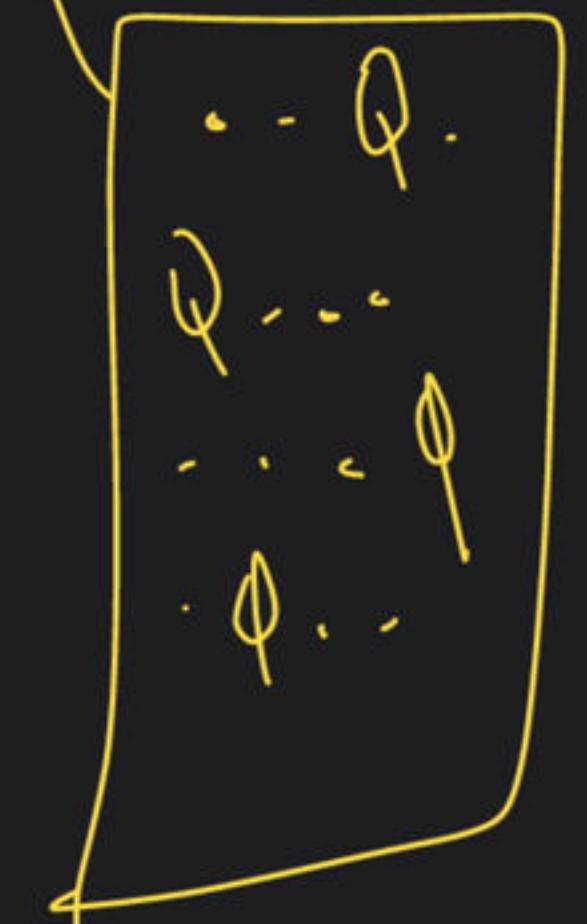
$\text{map} < \gamma_{\text{low}}(\delta) \rightarrow \text{bool}_{\alpha}$

return



vector <vector<string>> ans;

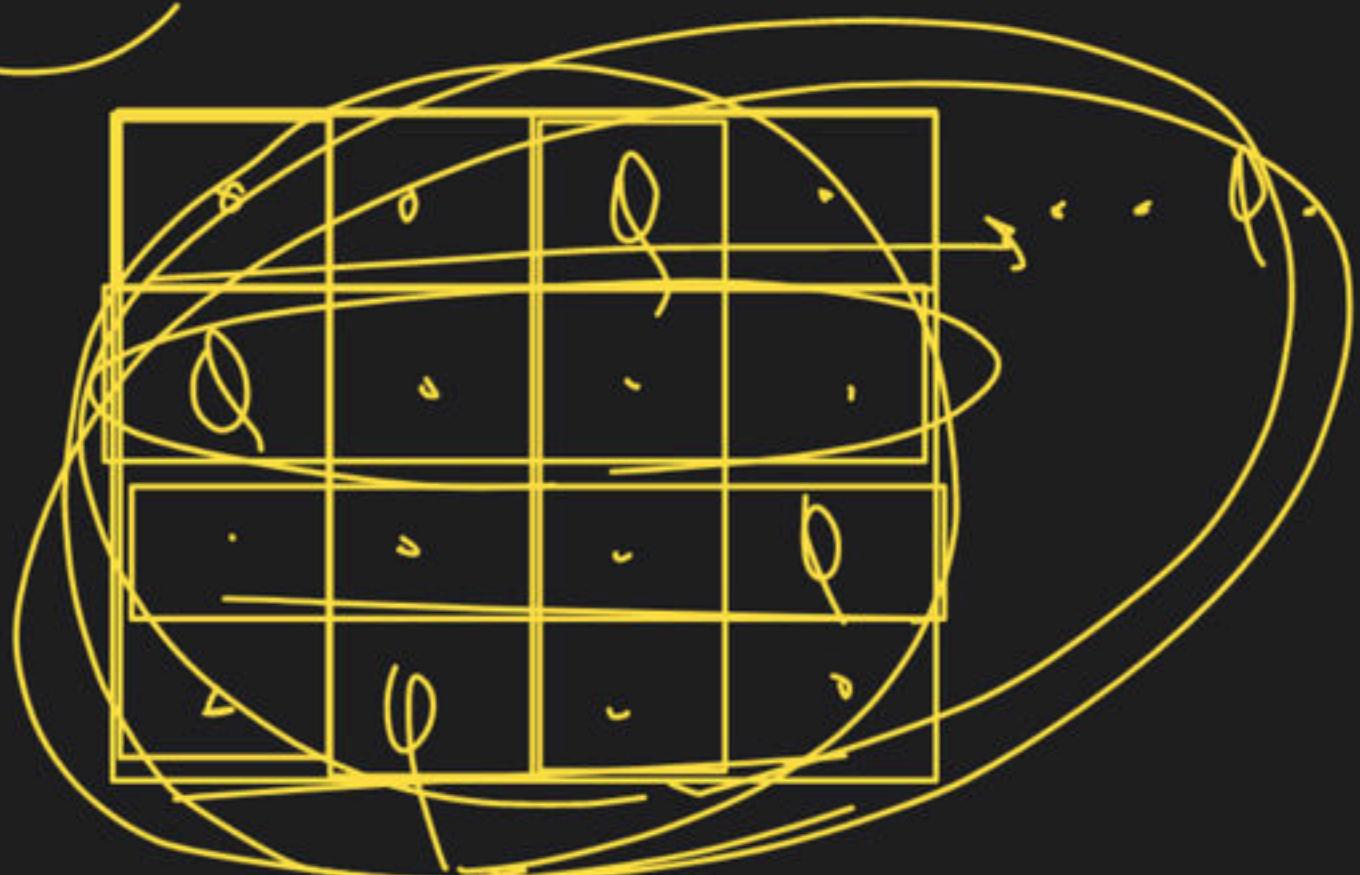
ans.pushback(temp)



board



vector <vector<(char)>>



for (row →

for (col → atv = ``

atv

}

Q..Q..``

ans

vector <vector<string>> ans

1 answer

row →

low)

→ row + col

up)

→ row - col

board

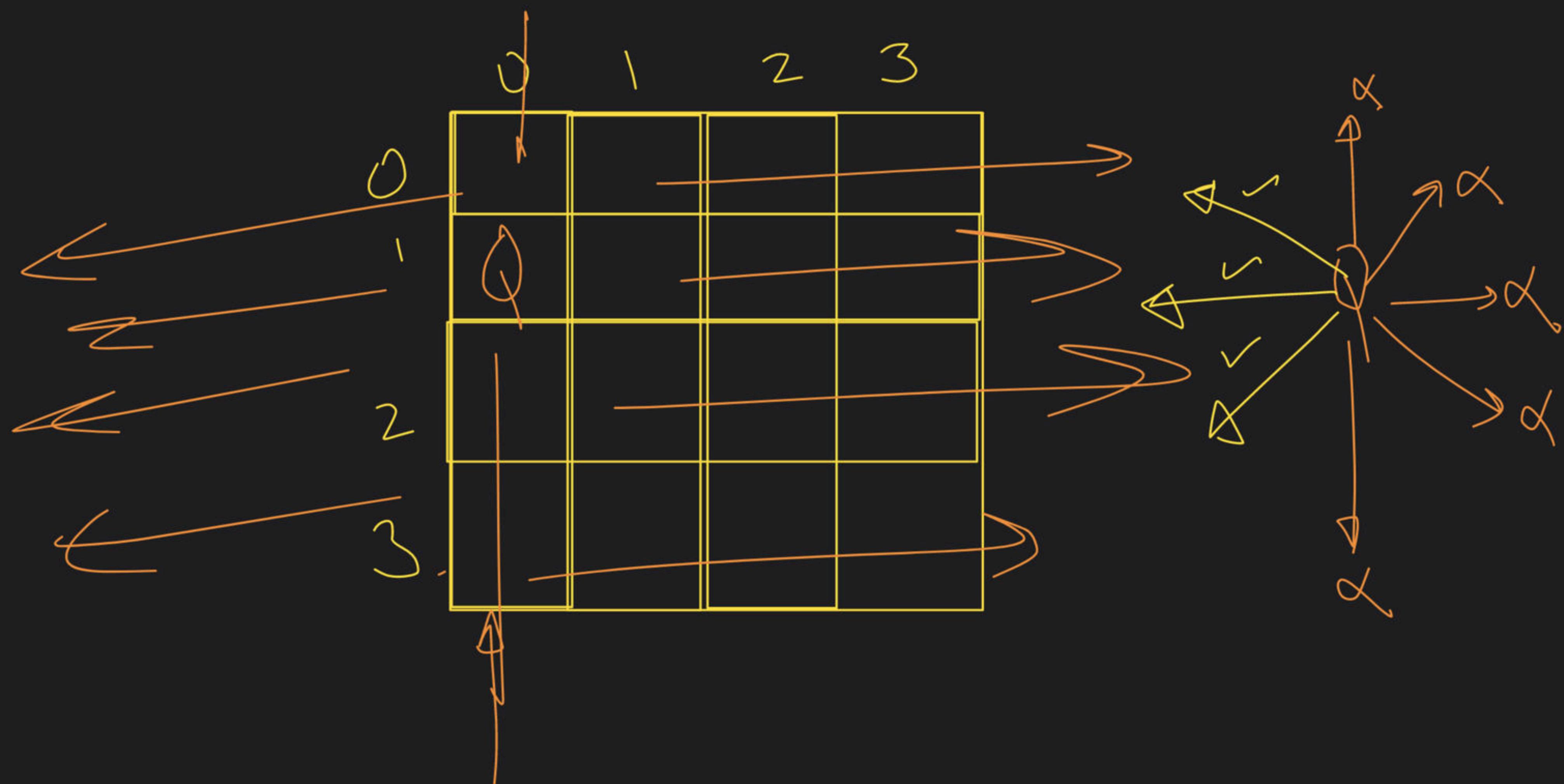
.	.	.	.
Q	.	.	.
.	.	.	Q
.	Q	.	.

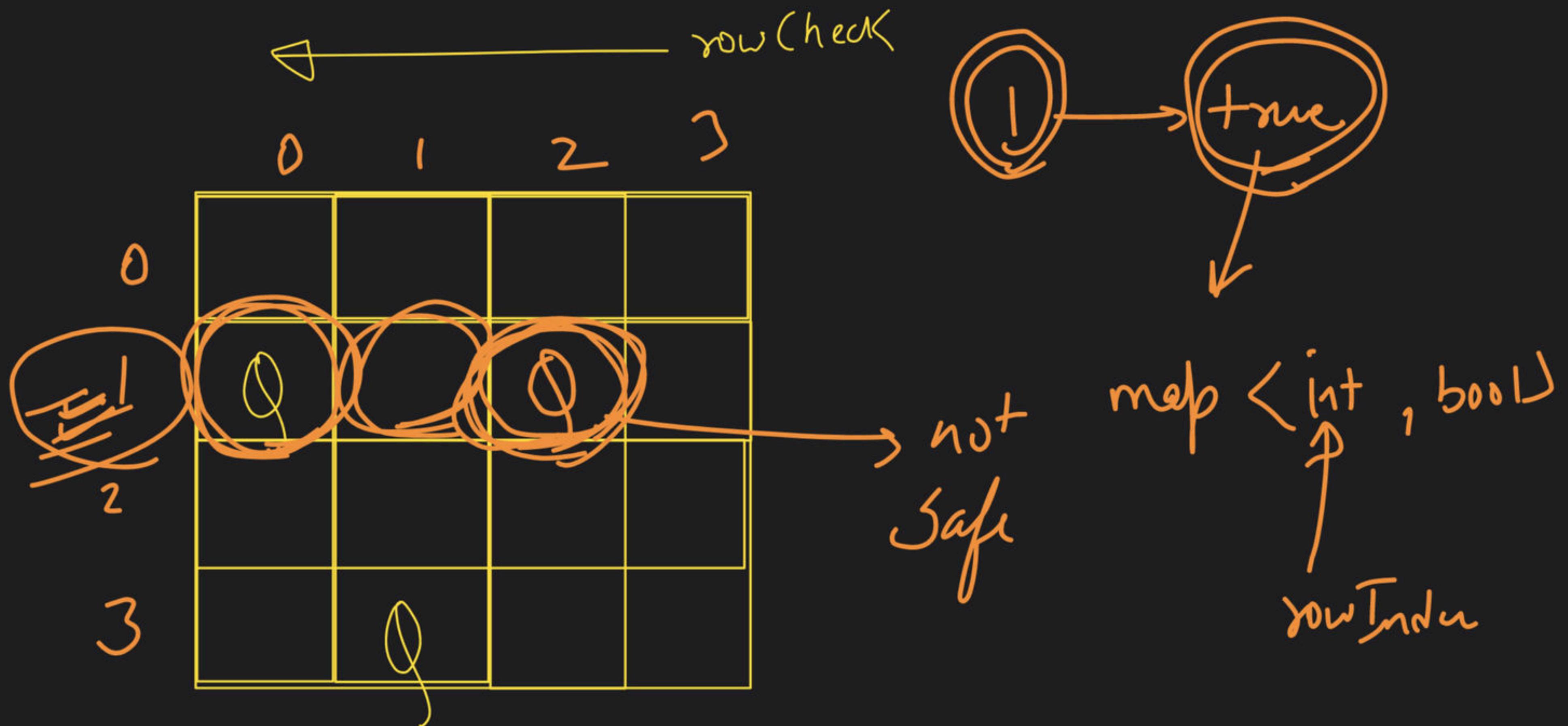
vector<string>
temp

"...Q..."

"Q..."

"..."



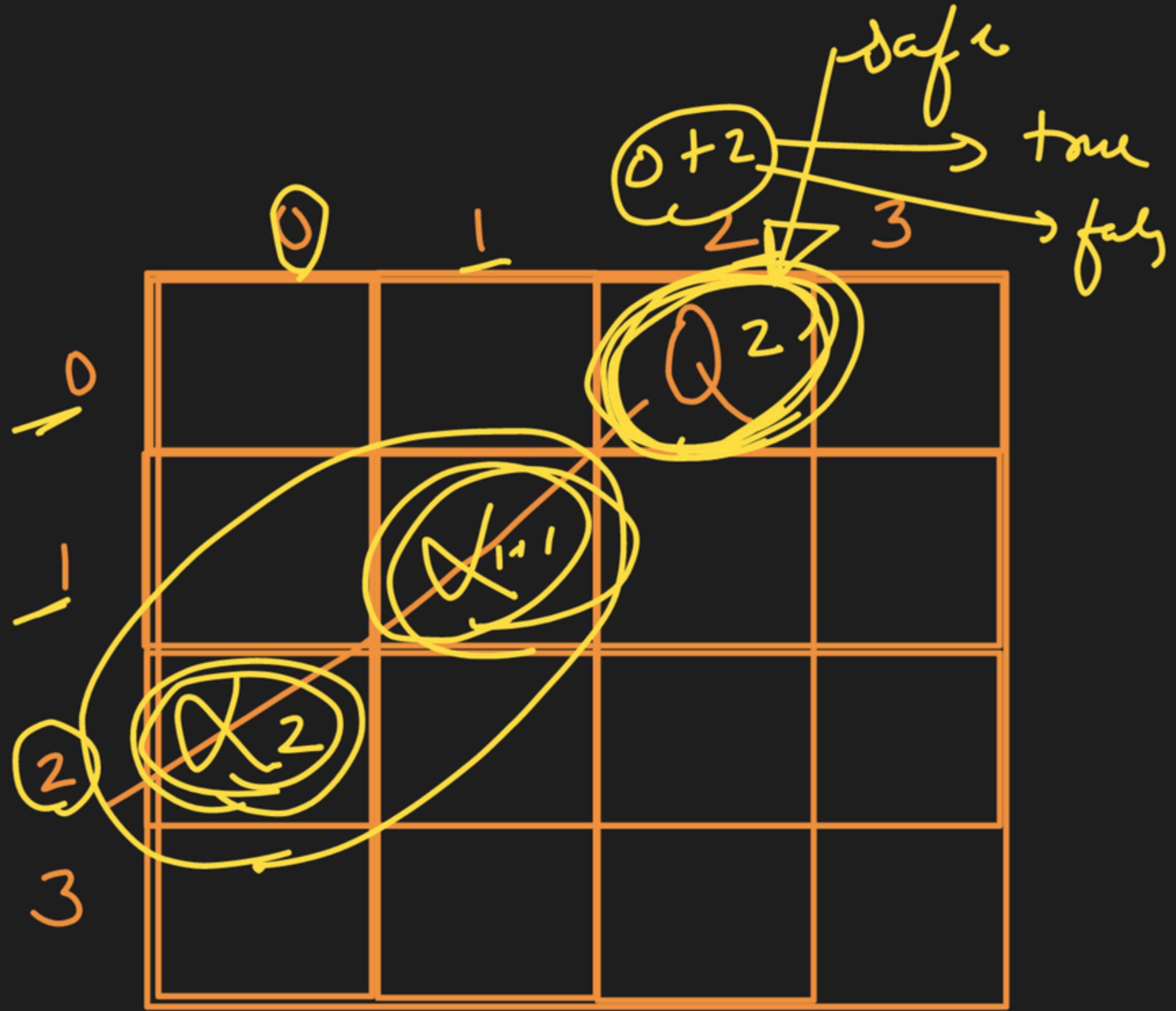


lower Diagonal

rowIndex + width

$i+j \longrightarrow \text{bool}$

map < y^t bool >



Upper Diagonal

row Index - (0) Index

i-j

map < int, bool >

