

→ Recursion

Assignment

Backtracking - I

Foundation Course on Data Structures & Algorithm - Part I

→ Backtracking :-

↑
→
Person
Re
Nk nca

Note



Letter tile possibilities:-

1 min to read question

ip → "AAB"

o/p → 8

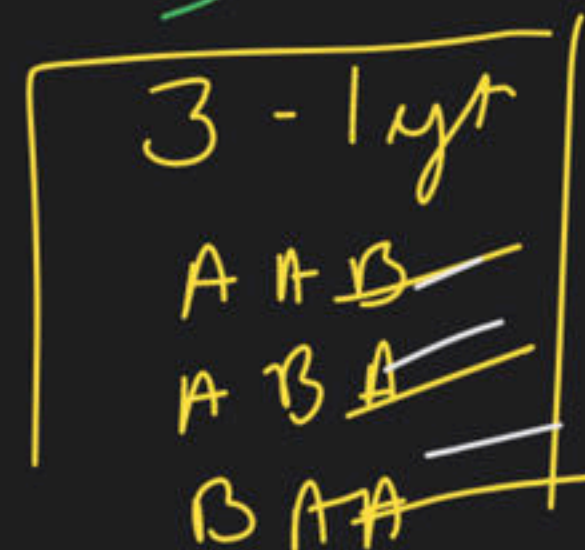
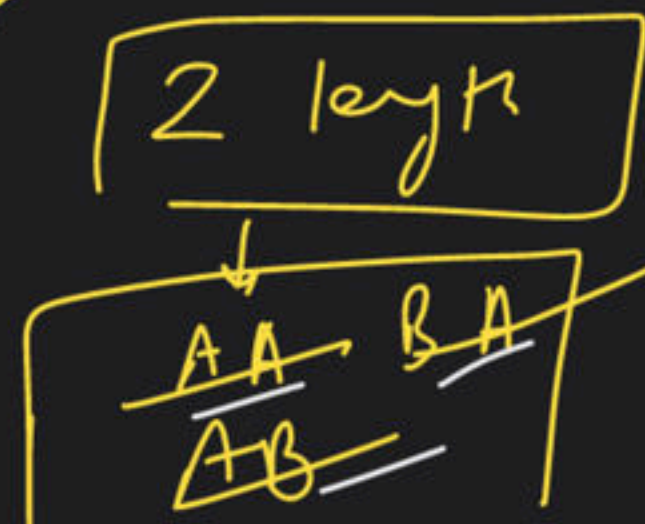
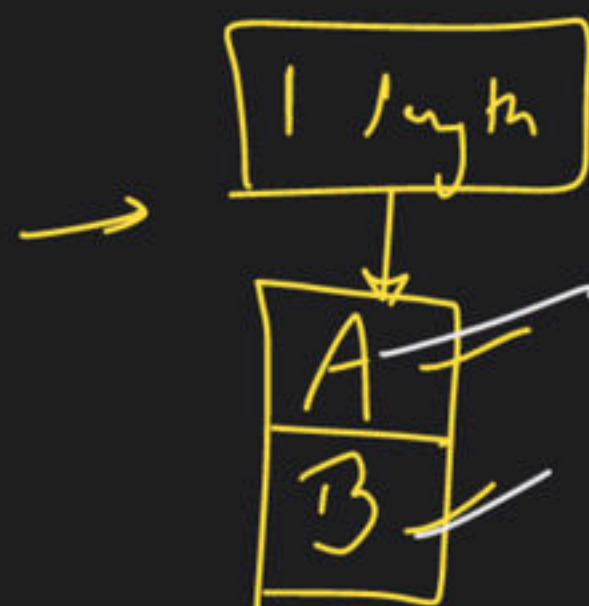
["A", "B", "AA", "AB", "BA", "AAB",
"ABA", "BAA"]

500E

AND
DATA

BVBVB

"AAB"



→ 8

i/p \rightarrow "V"

o/p \rightarrow 1

i/p \rightarrow "XY"

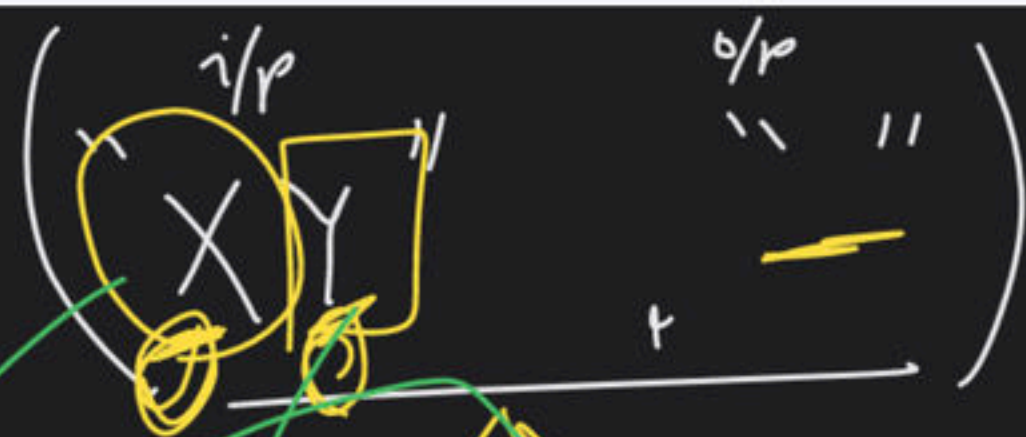
o/p \rightarrow ?

X	XY
Y	Y X

 \rightarrow (4)

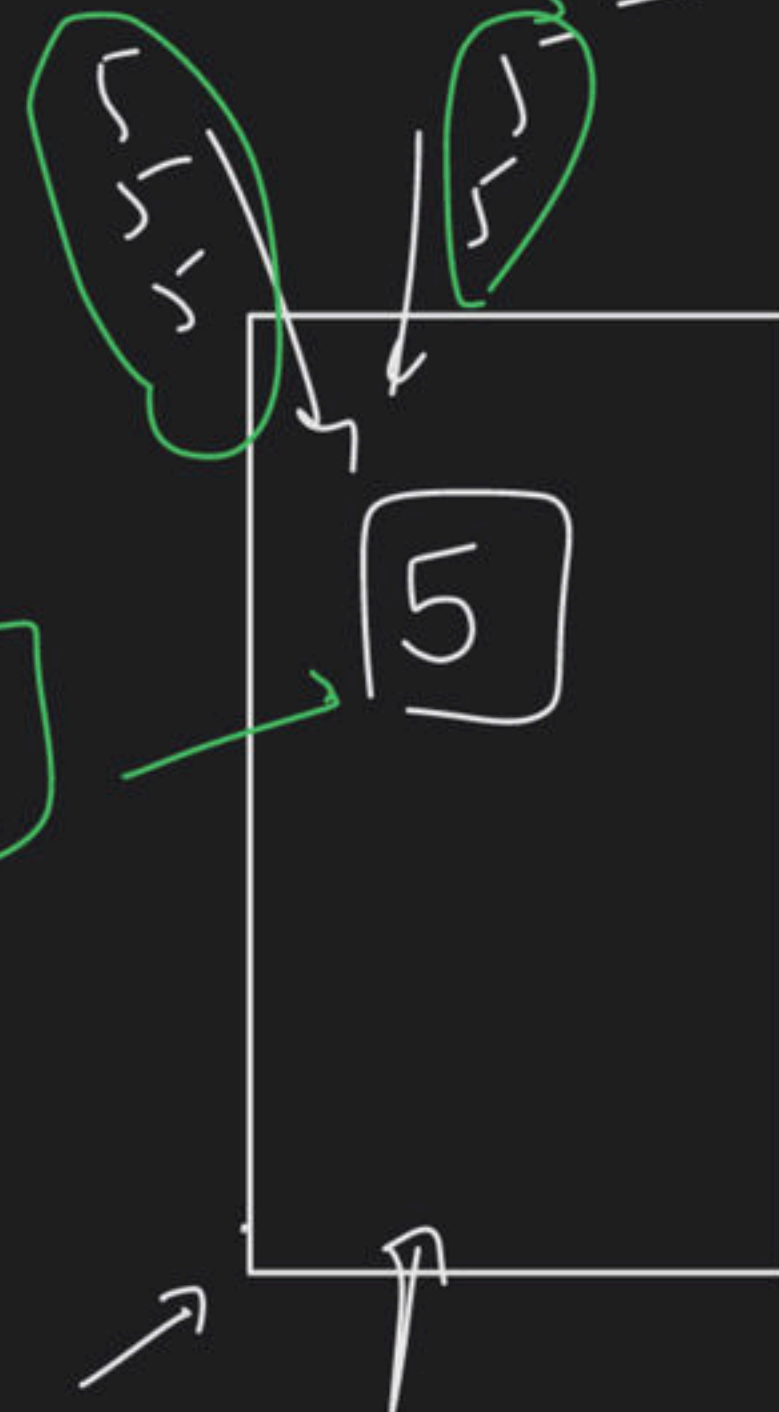
i/p \rightarrow AAAABC

o/p \rightarrow 138 \rightarrow



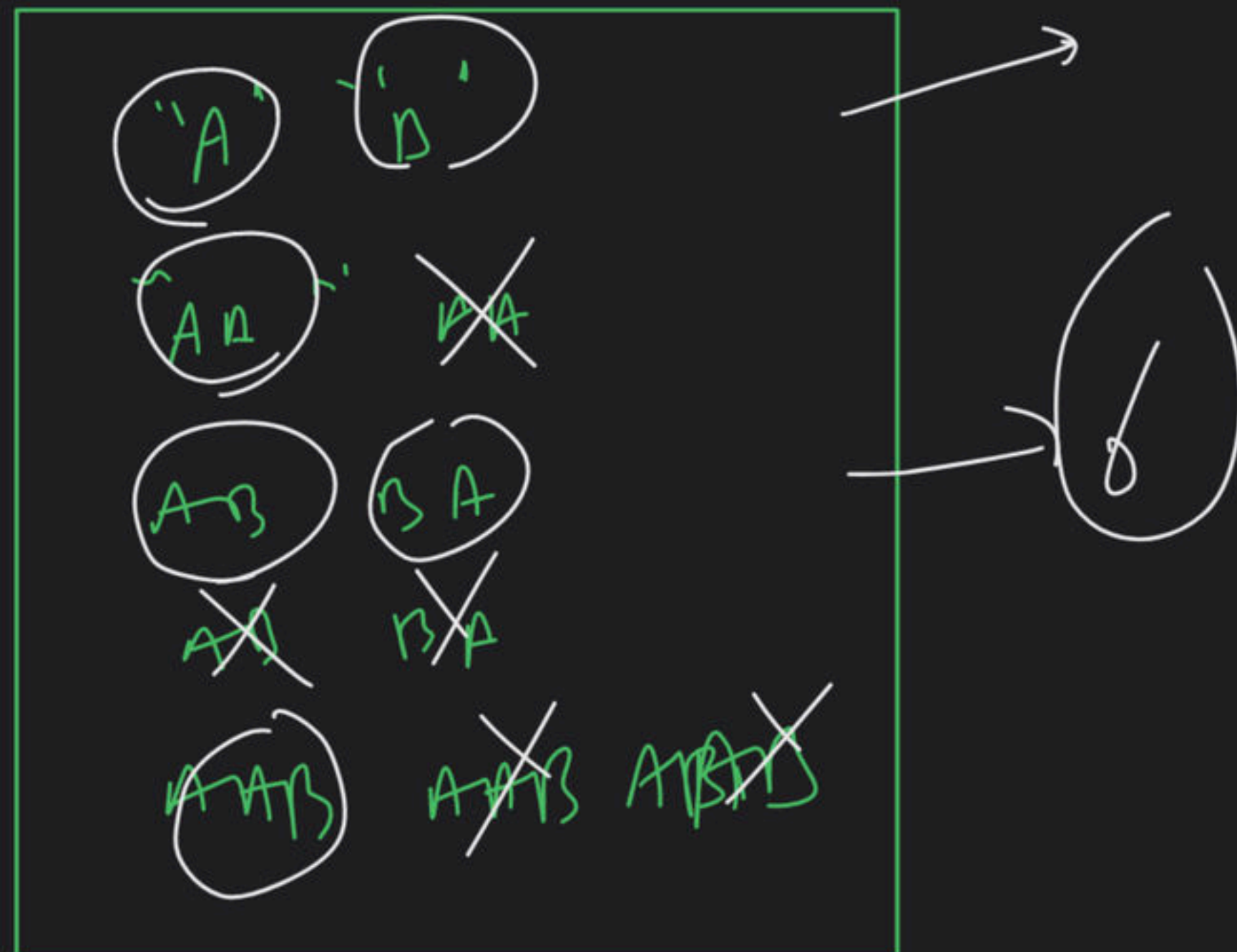
Data Str

unique



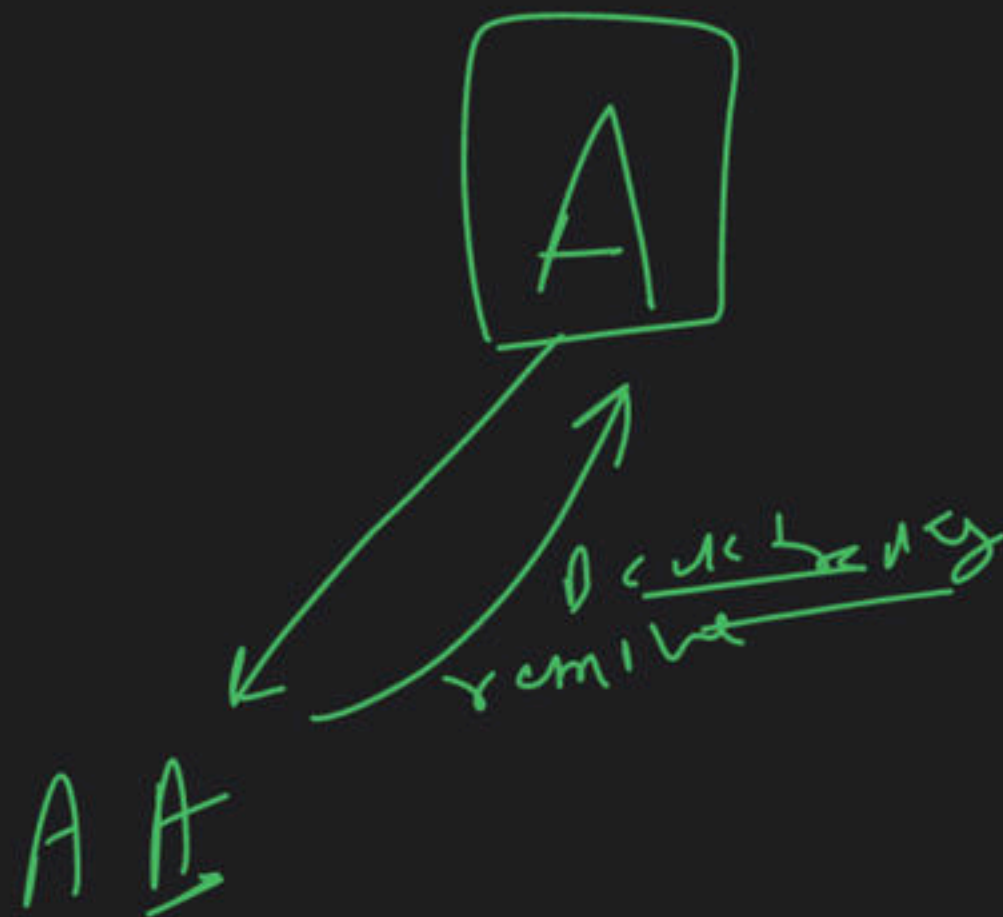
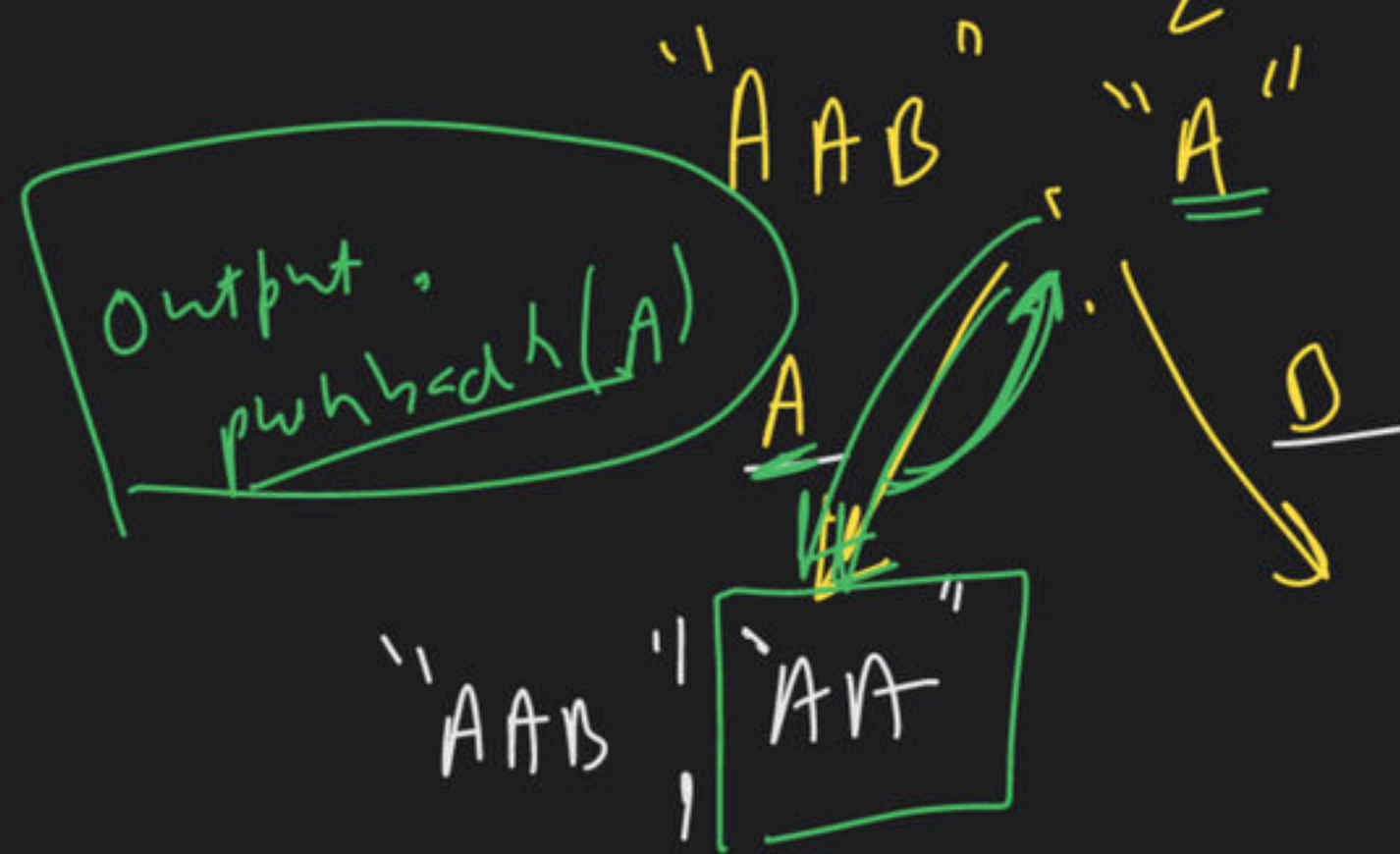
Set





A[A] → A

output.pop-back()

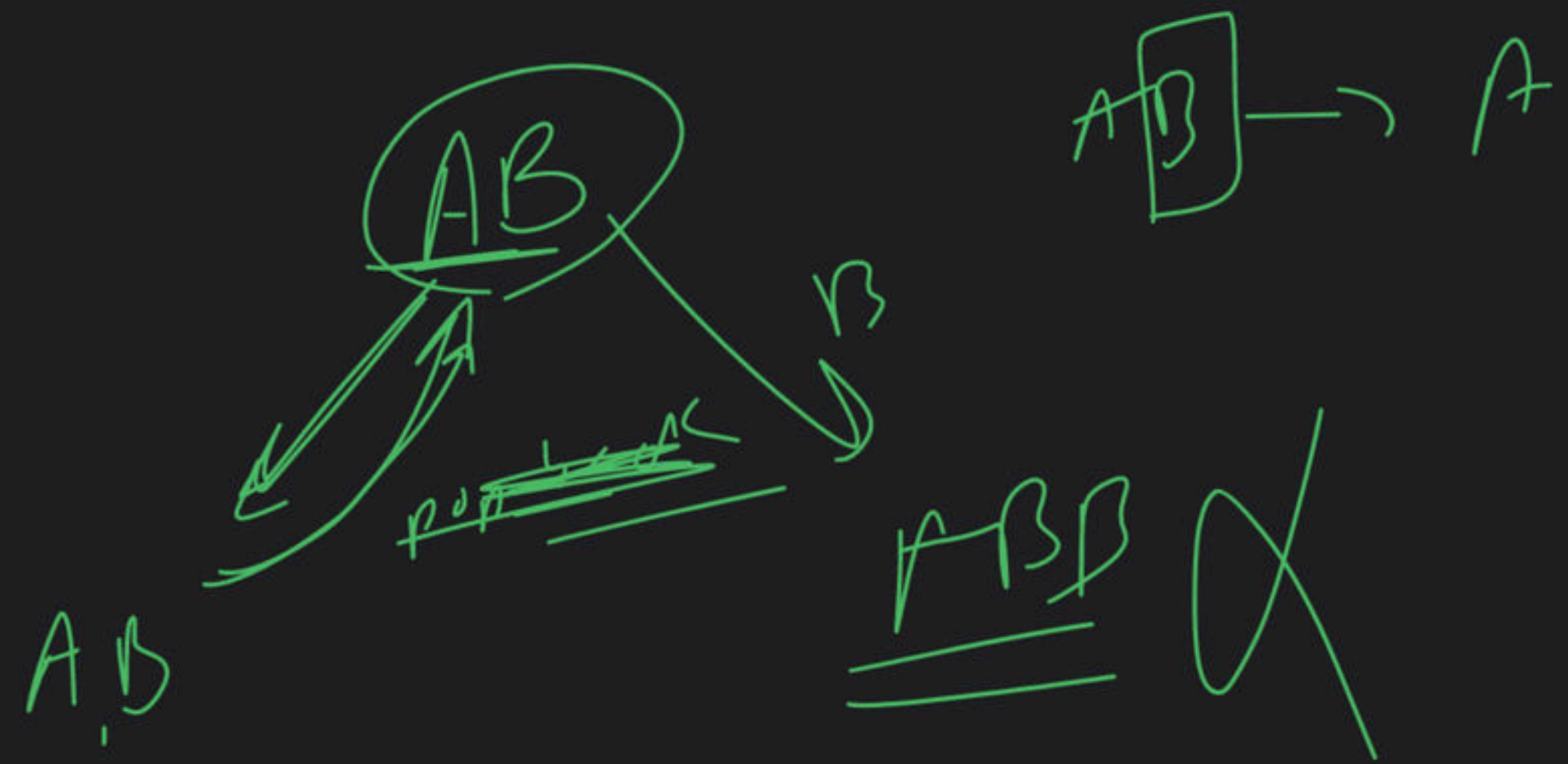


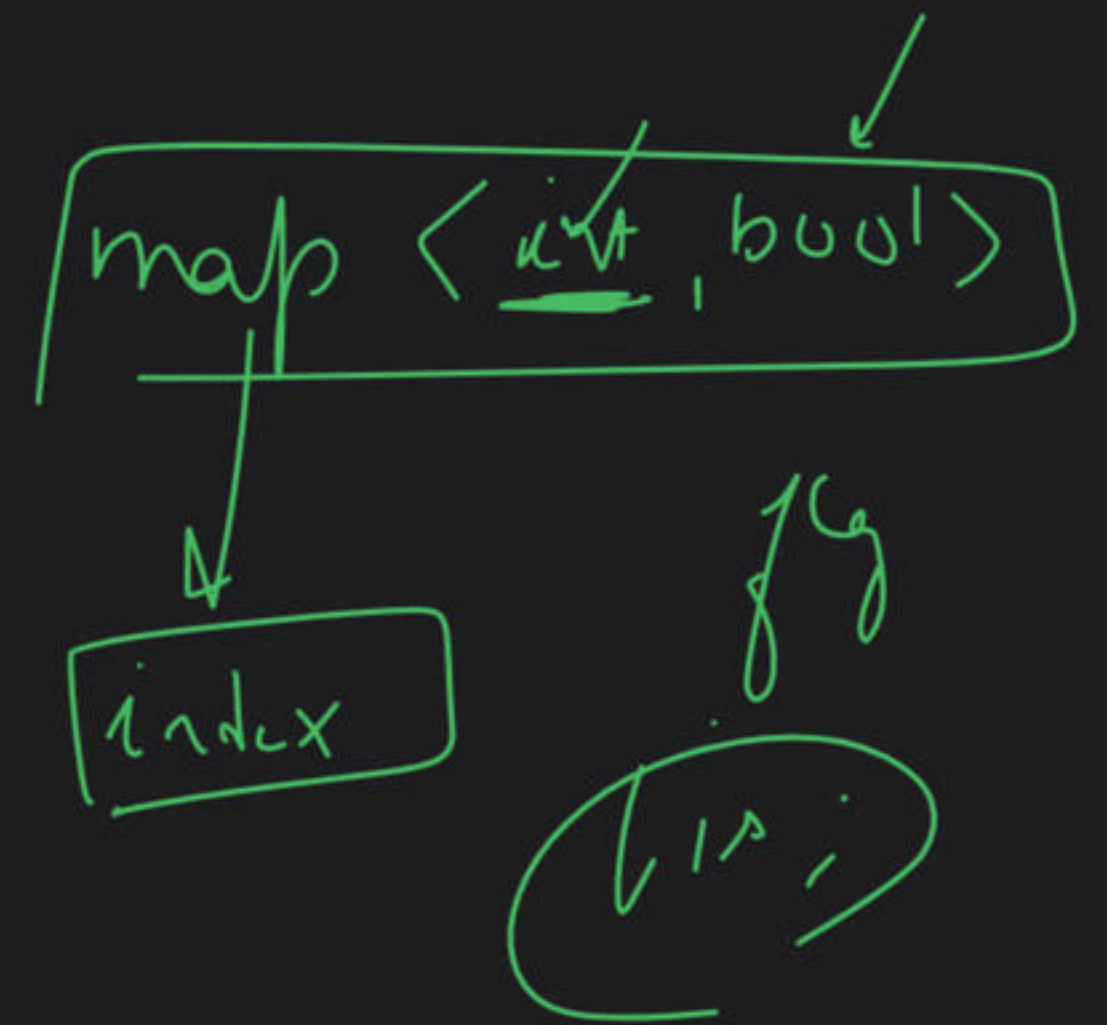
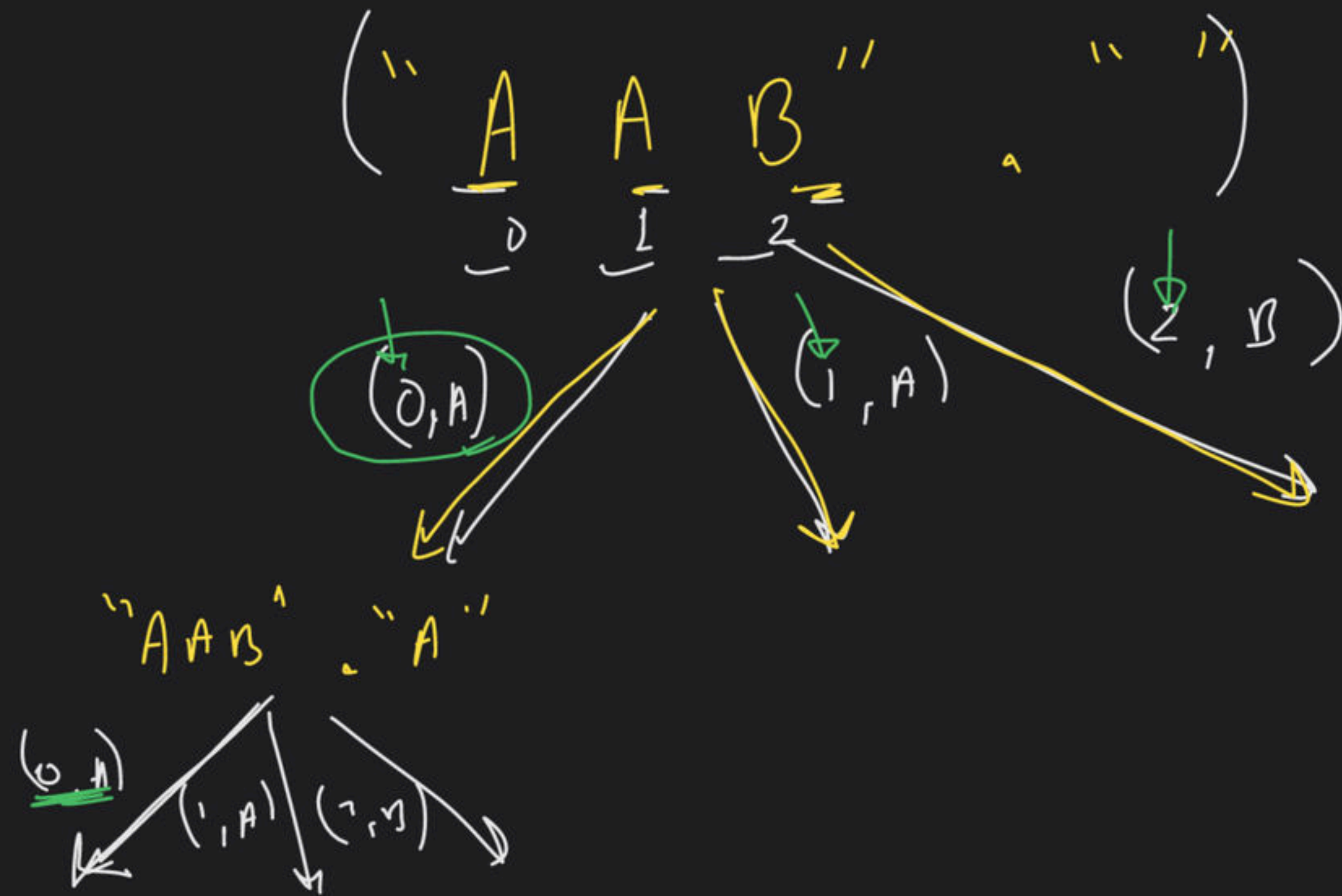
str.length

for ()

RC

Permutation





$\left(\begin{array}{cc} \text{"AAB"} & \text{" " " "} \\ \text{---} & \text{---} \end{array} \right)$

0	→ T
2	→ T



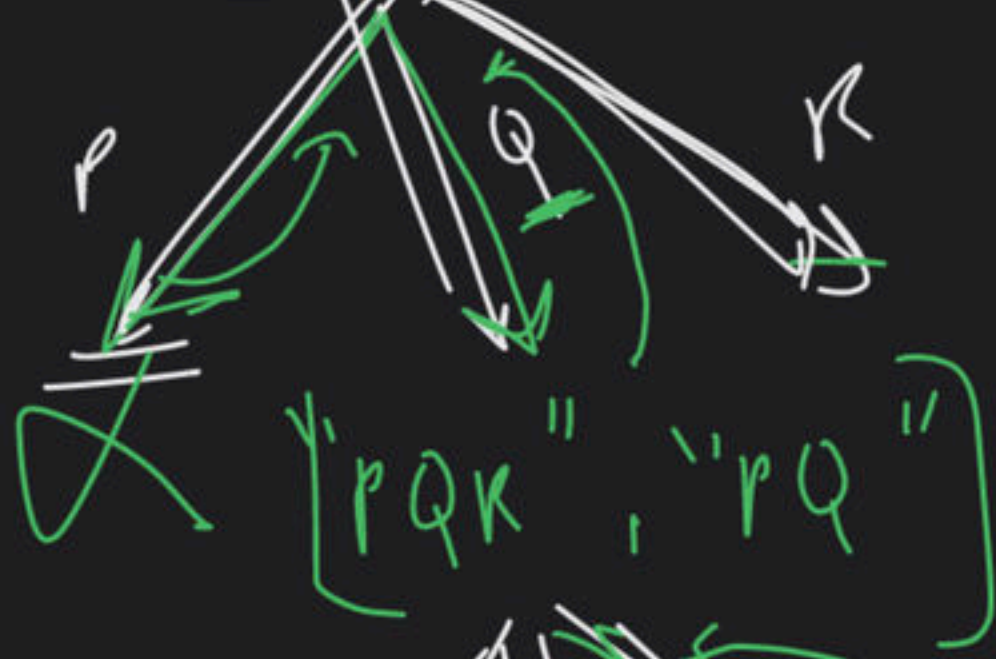
map <int, bool>

0	→ true
1	→ true

2 → true

0	→ T
2	→ T

$[PQR, P]$



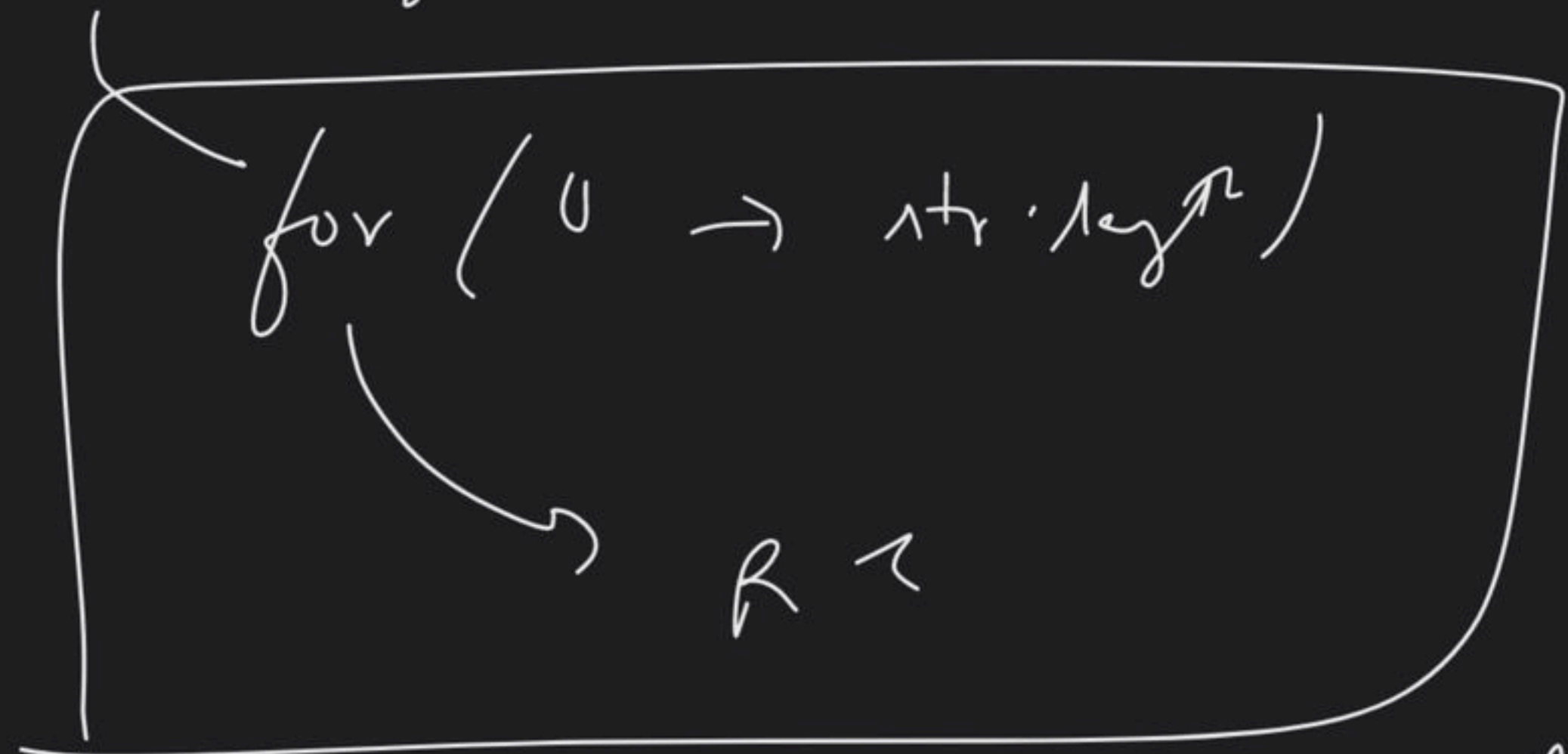
PQR

PQR

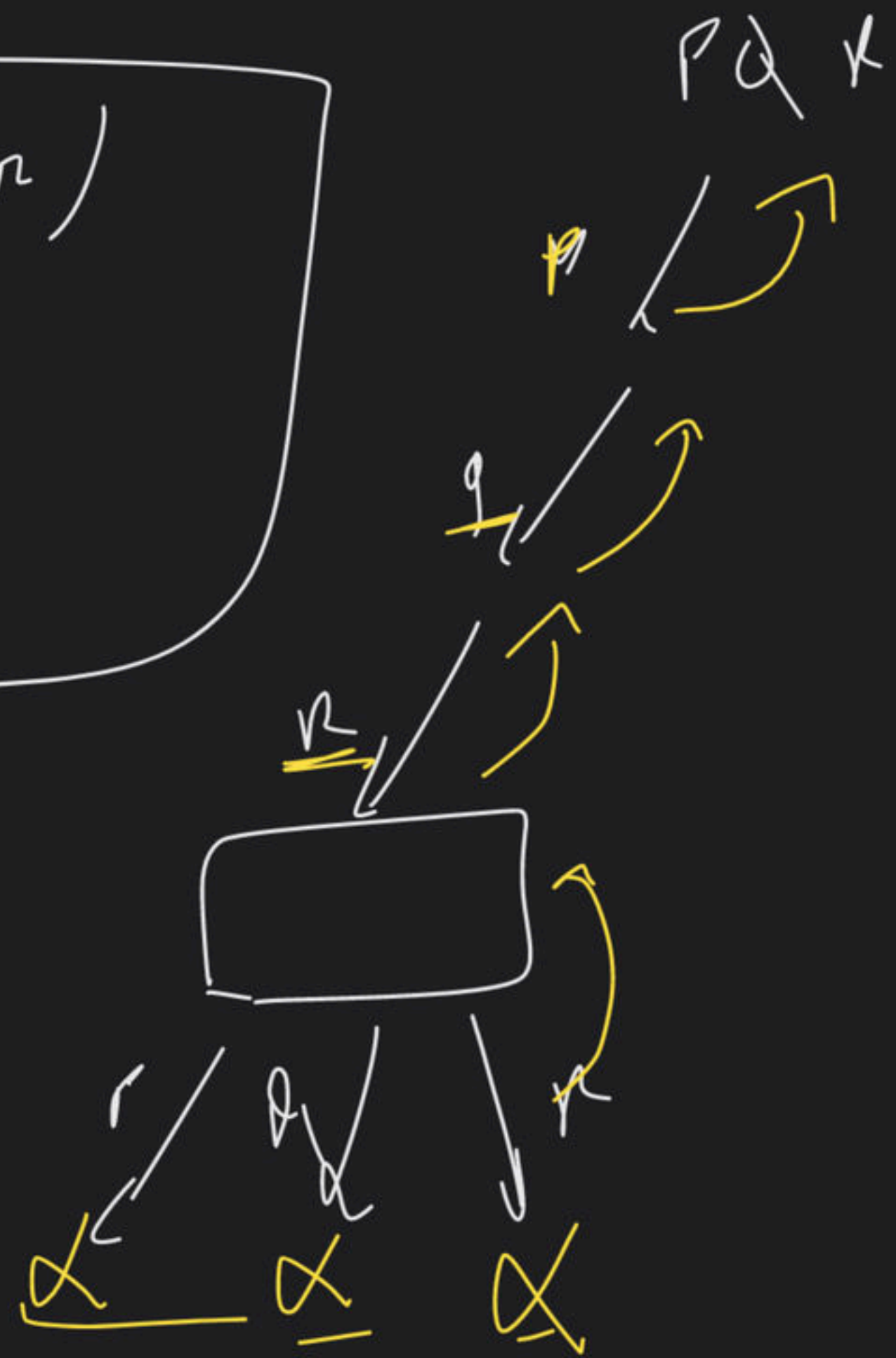
PQR, PRQ

PQR, PQR

str. left



\rightarrow map α



no duplicates

set <int> s; ← create

unique
element

<u>1</u>	A
2	AA
<u>3</u>	

s.insert(1)

s.insert(2)

s.insert(3)

s.insert(1)

s.insert(3)

অনু (২১)

'AAB'

for (—)

int arr[20] = {0}

1/1/4

STL Class

5 ↓
Mazal^c

R K G I I

0	2	a
1	1	b
<hr/>		
25		z



i/p \rightarrow $n = \boxed{3}$ $\begin{matrix} 3 \rightarrow 00 \\ 3 \rightarrow 10 \end{matrix}$

Generate Parenthesis 22
LC

o/p \rightarrow

()

- ((())) \rightarrow
- (() ()) \rightarrow
- (()) () \rightarrow
- () (()) \rightarrow
- () () () \rightarrow

} Brackets

Catalan
no.

55.55%

1 min

Output = "("

Output pop-back
 $vis[i] = 0$

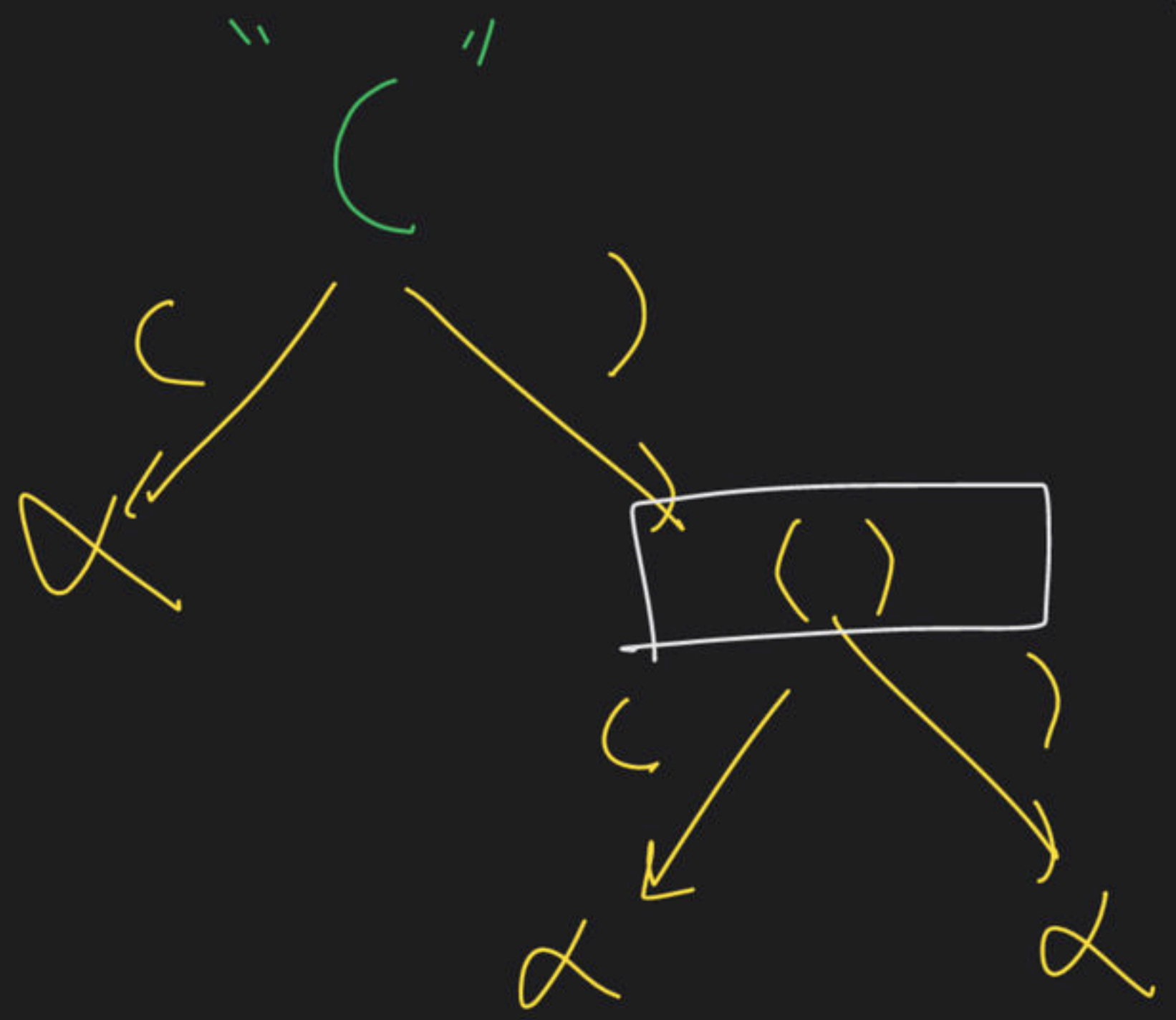
$$n=3$$
$$\begin{aligned} 0 &= 0 \\ 1 &= 0 \end{aligned}$$

7. L

$n=1$

$(\rightarrow \cancel{1} / \cancel{0}$

$) \rightarrow \cancel{1} / 0$



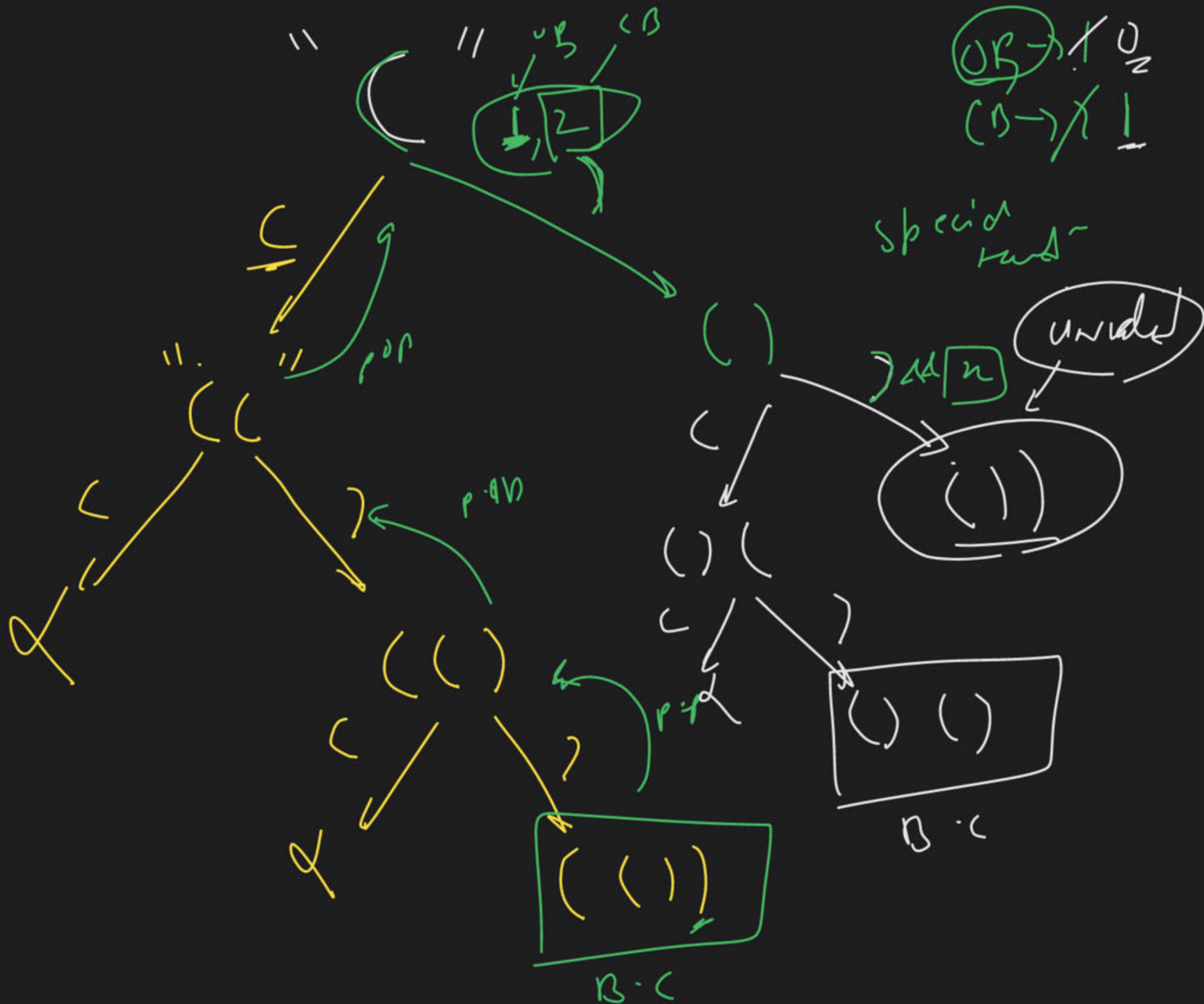
$n=1$
 $\boxed{()}$

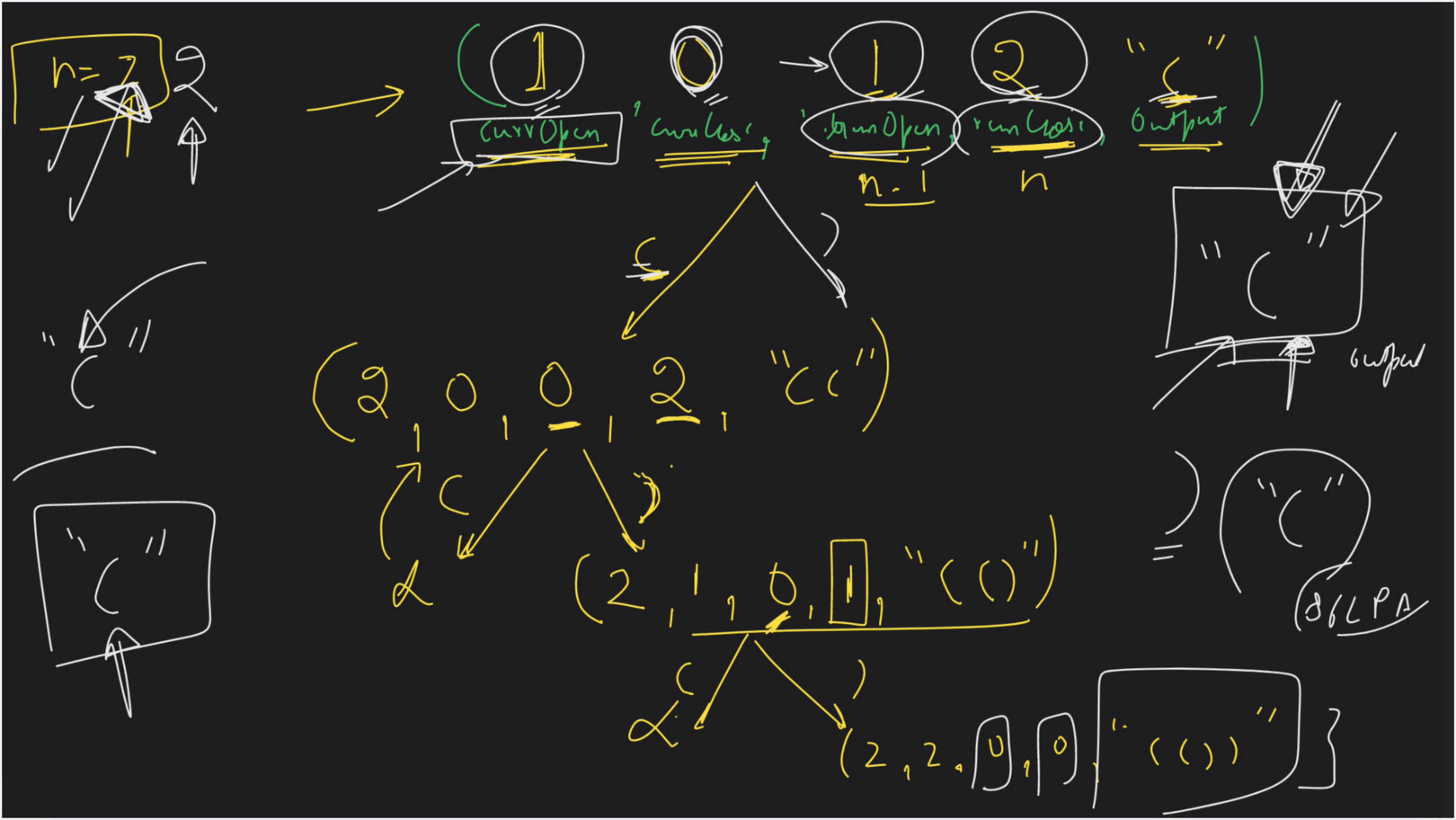
$n = 2$

OB \rightarrow ~~2~~ ~~1~~ 0

CB \rightarrow ~~2~~ ~~1~~ 0

OB = 0
CB = 0





Valid
Bracket

Output = ")"

↓
Possible

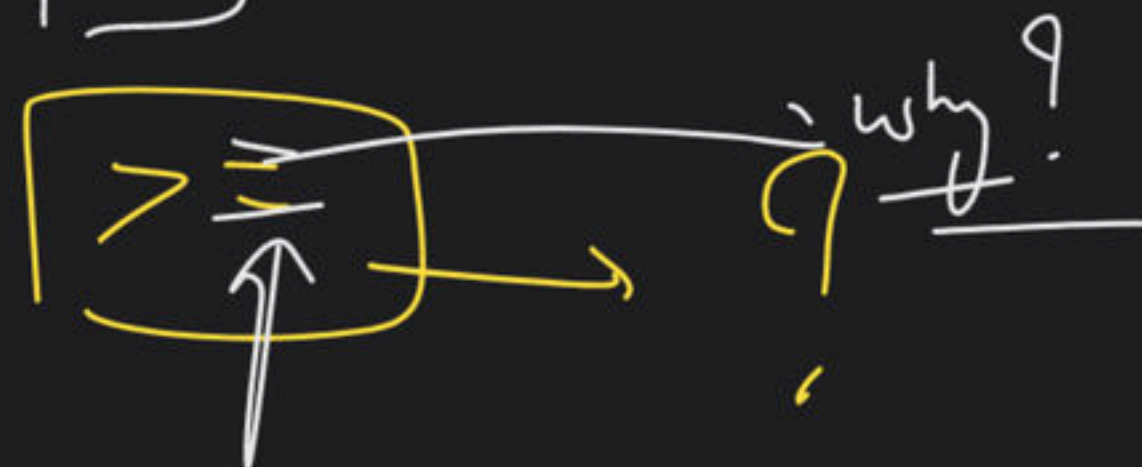


$(\underbrace{() }_{=} \underline{() })$

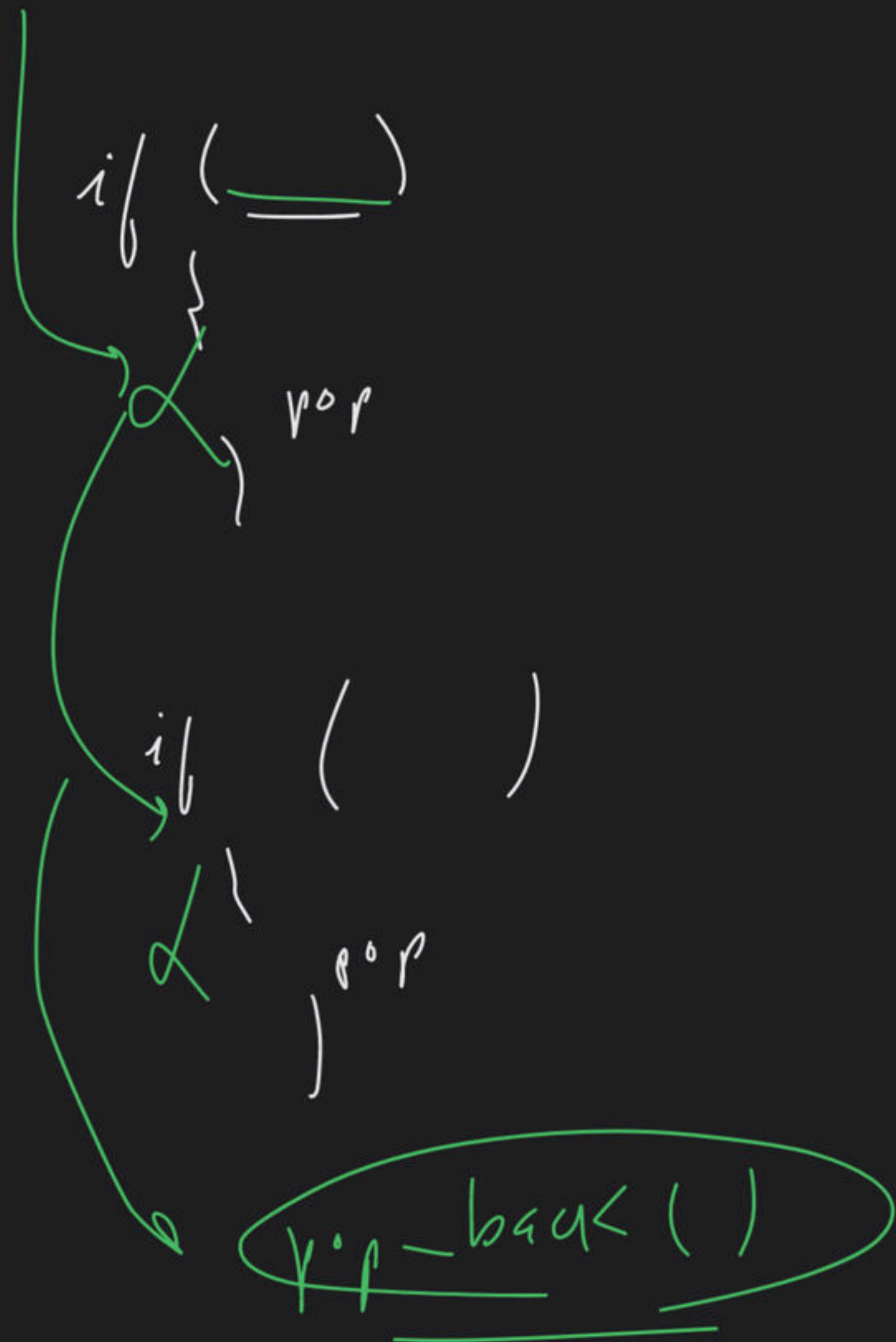
< numOpen > numClose

$(\underline{() })$

CursorOpen > cursor loc → Why



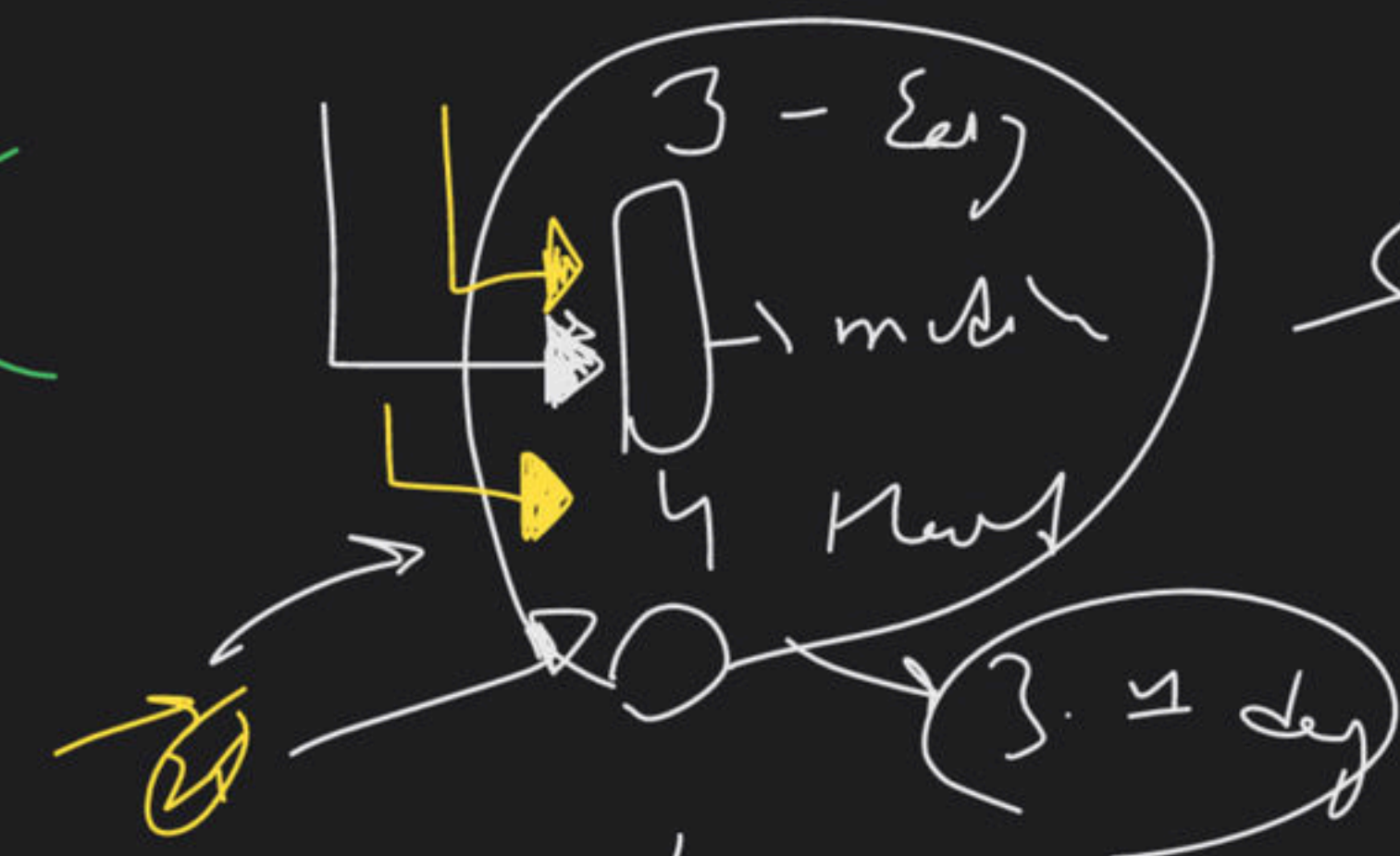
No+ allo



→ Dipansh

Permutation —, How?

Rec tree



Reunion

add

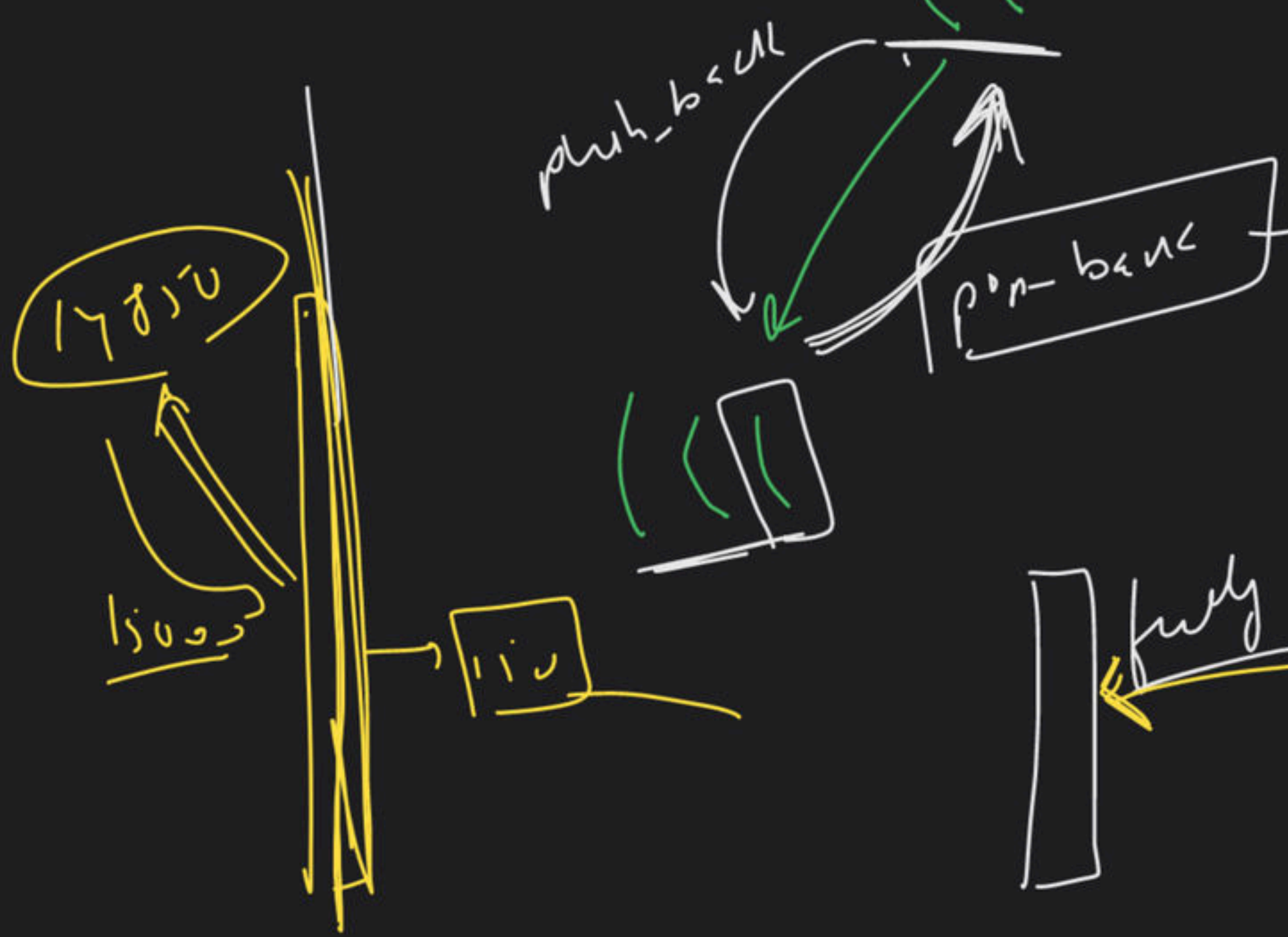
Free

(11)

swap

Test

→ 1 Question



Backtrack

fully prepared

AKD

Youtuber

14

This is the Day

Kripa

Ramesh

Uthage -)

Sothage

hornel

Love

Uthage

Don

Uda

mhm

Kn

sen

Pik

Sh

Love Oberoi

View Sameer Malhotra

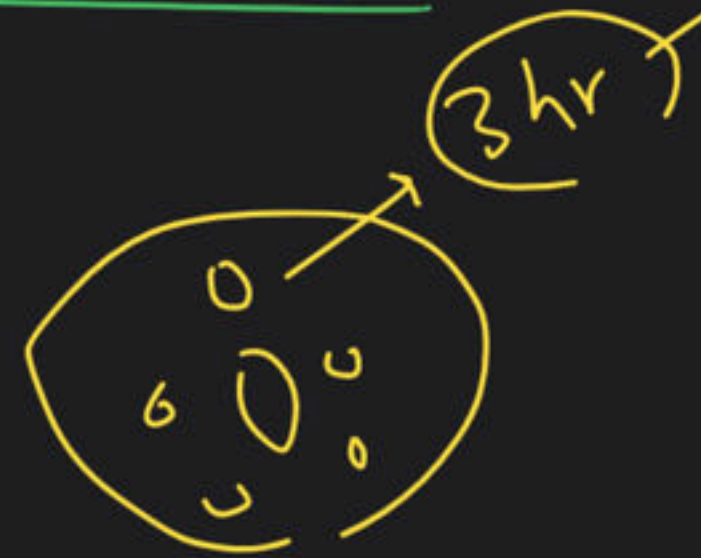


→ Beautiful Arrangement

$n = 3$

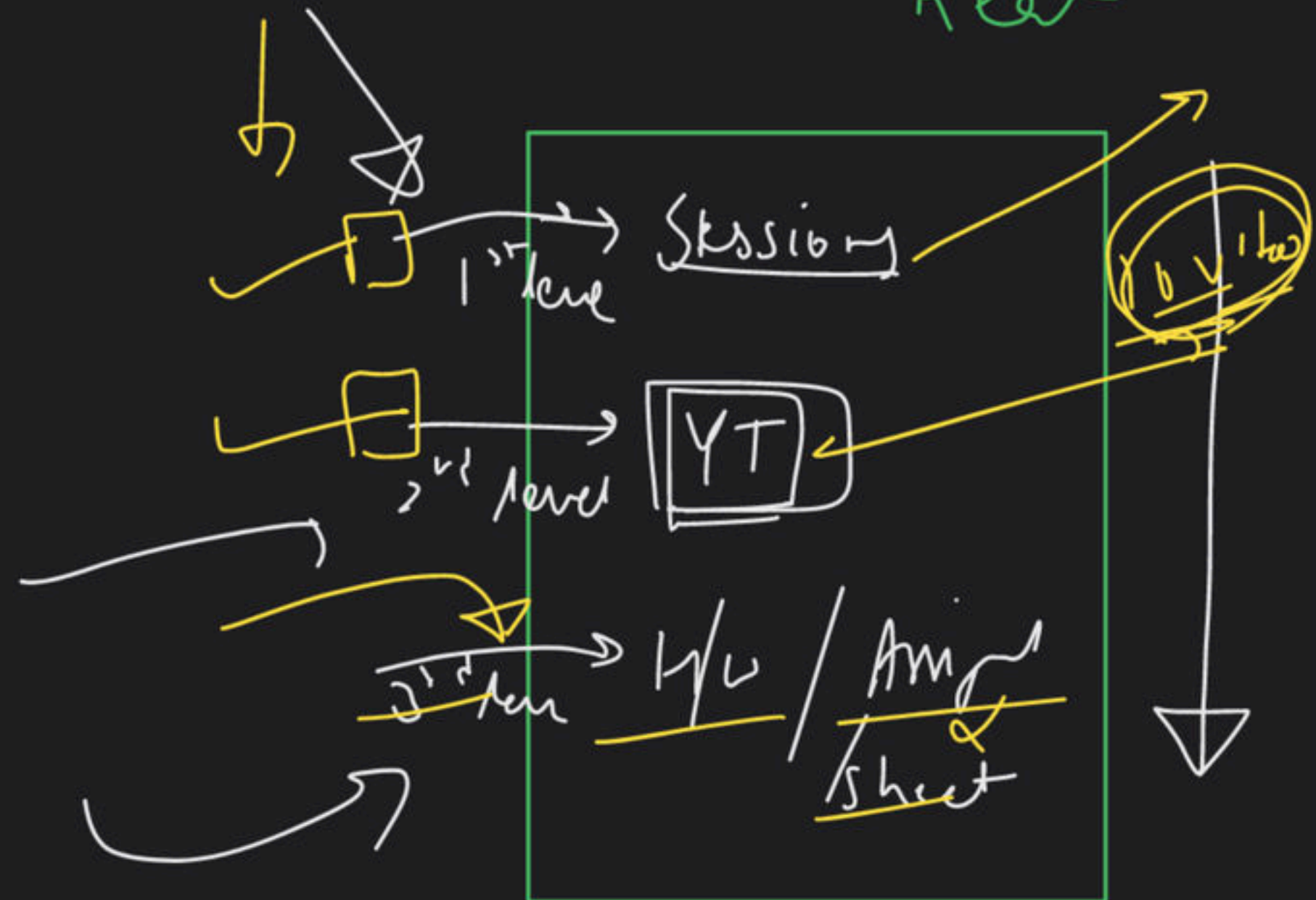
1, 2, 3

- ① 1 2 3
- ② 1 3 2
- ③ 2 1 3
- ④ 2 3 1
- ⑤ 3 1 2
- ⑥ 3 2 1



1 min

Qucy Read



(2 min)

create

act < int > s;

(Google)

low

s: input

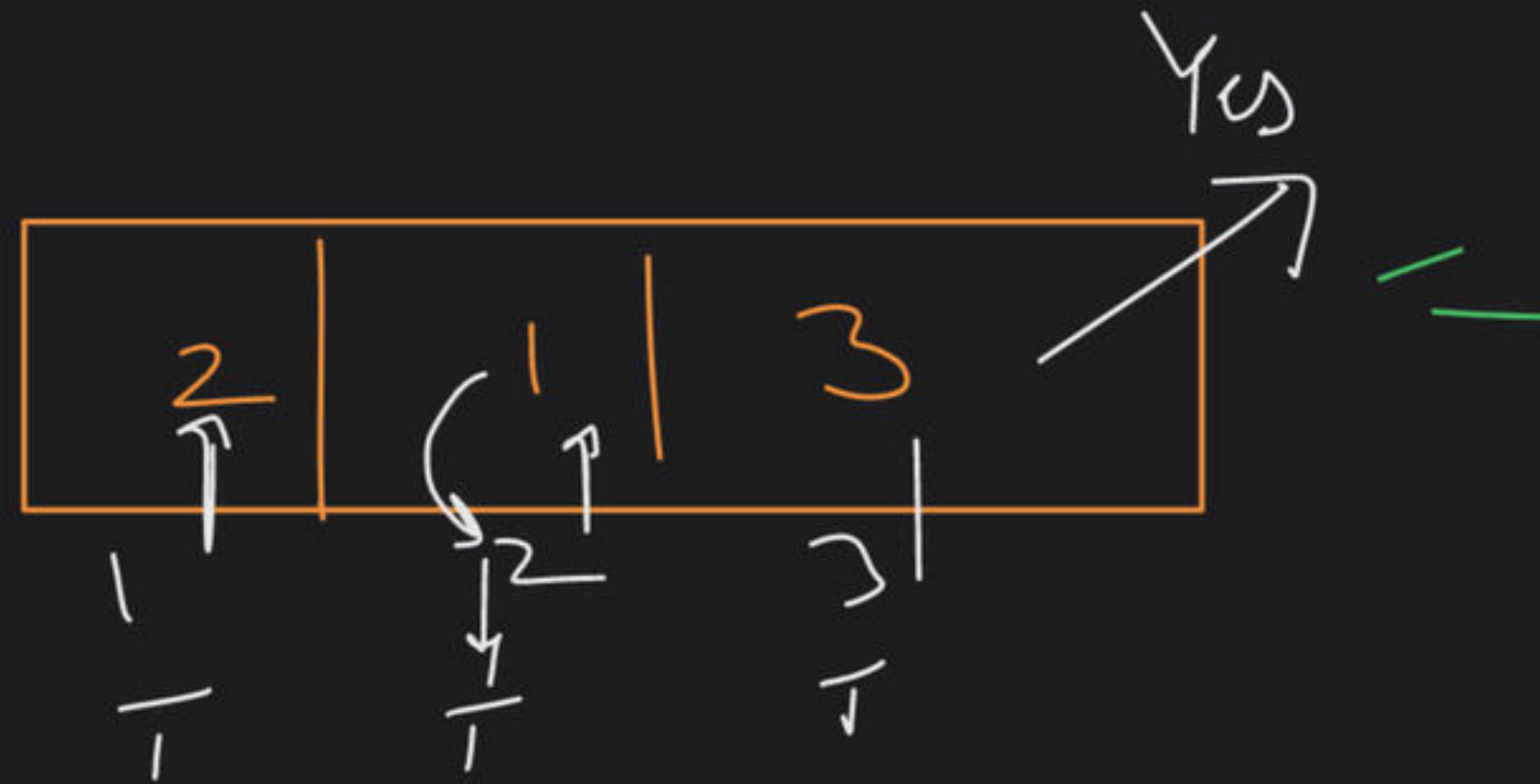
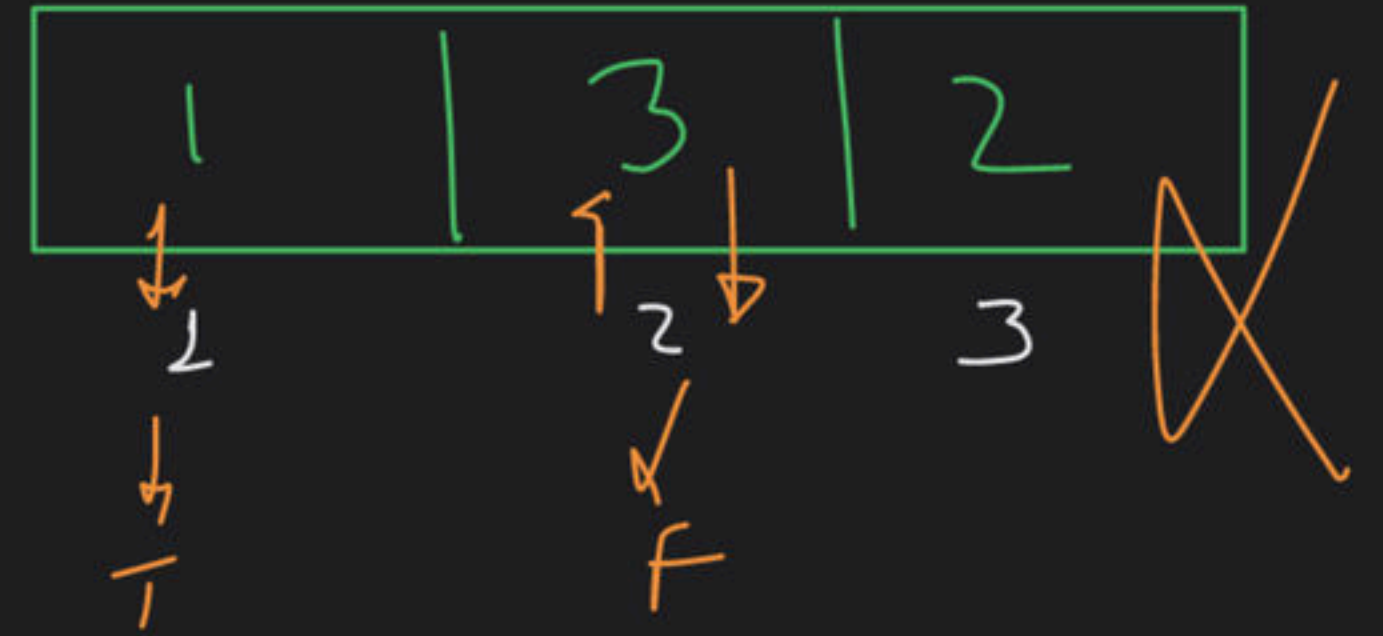
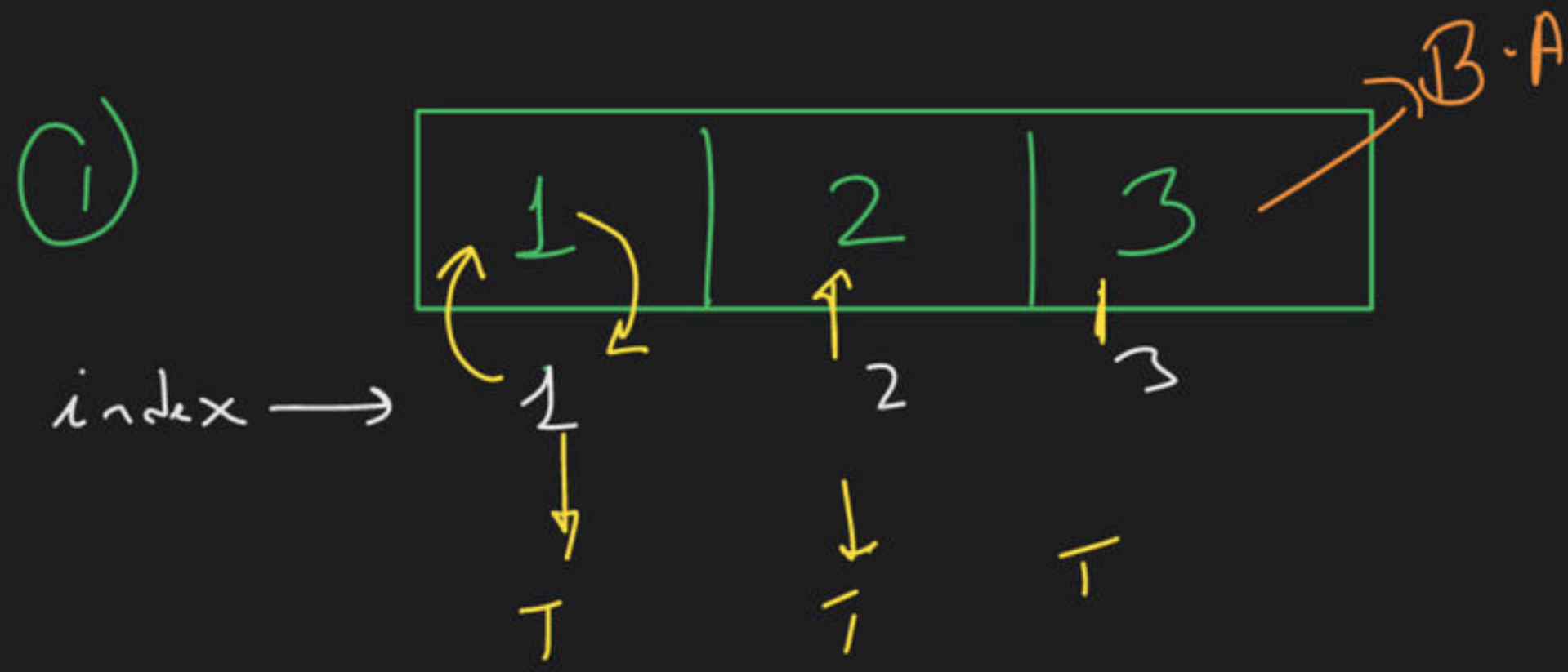
s: eval

(iterator)

low

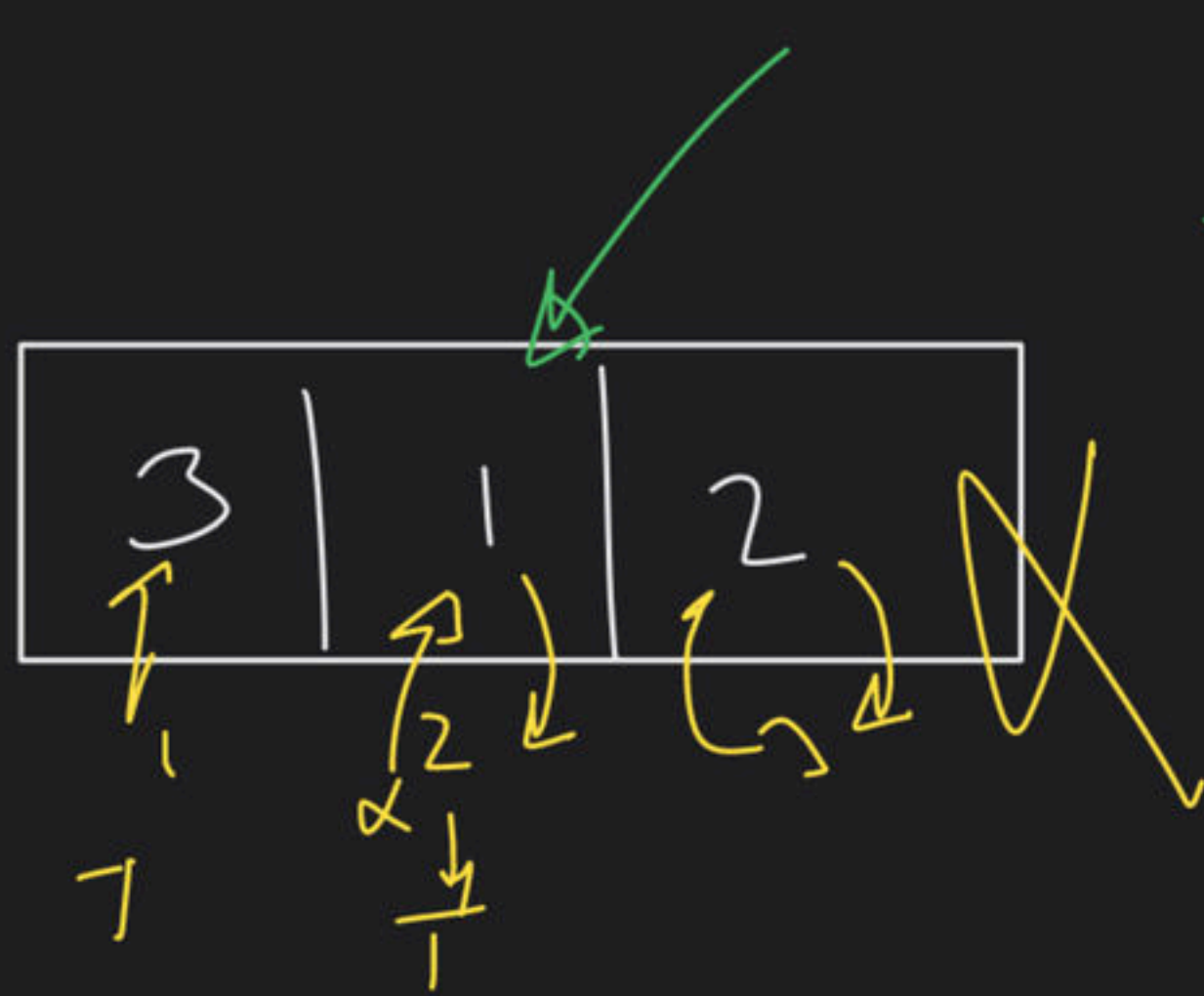
for (auto n: s)
{
}
}

(10 min)

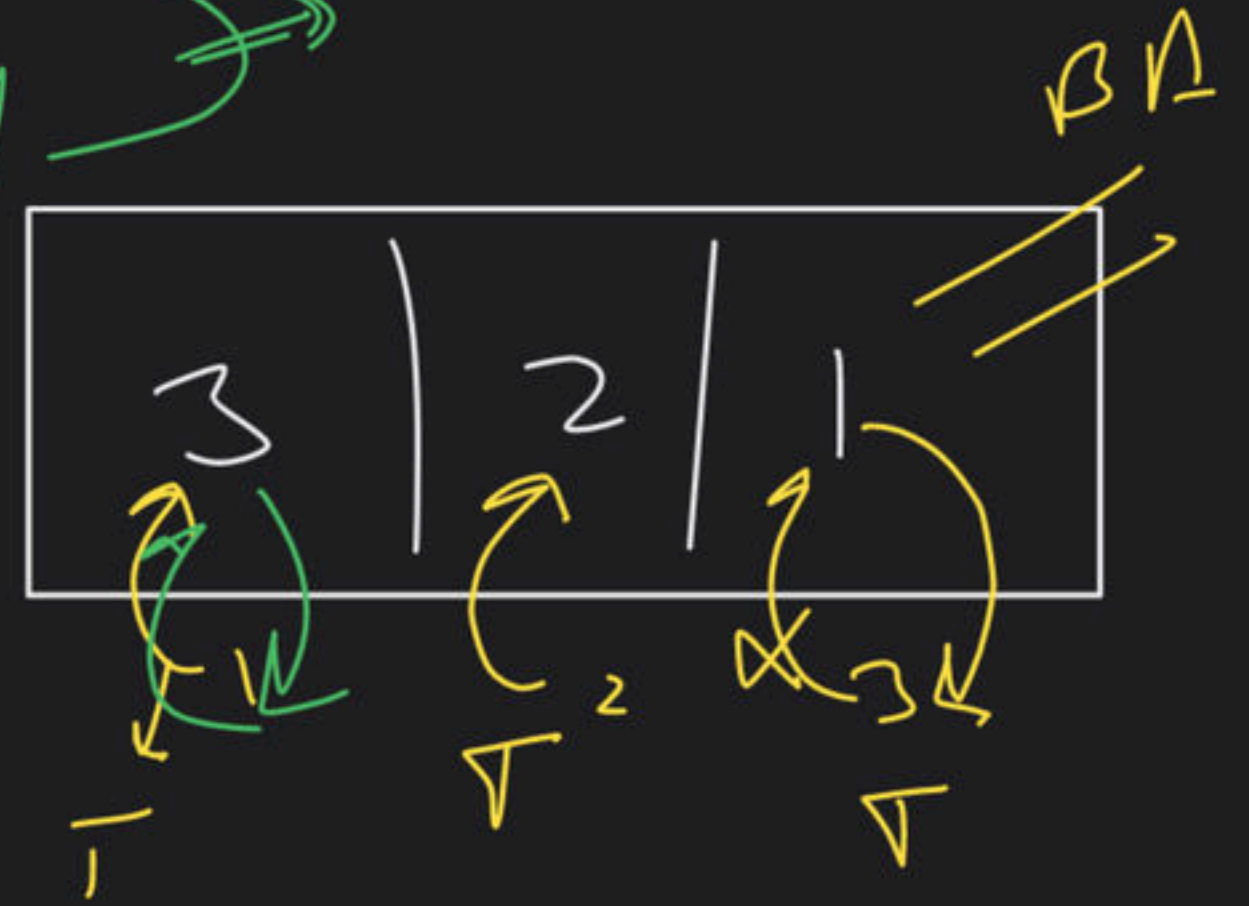


① $\% = 0$





IPB → 5 day →
 30%
 10% → 100%
 20%



$n=3 \rightarrow \text{v/p} \rightarrow$

1 2 3
 2 1 3
 3 2 1
 BA

Permut.
 4

Approach → saari pon → veru ← 1.5 hr

ST →

flex + - ~~reality~~ + -

Legit
flex

1.5 hr

2 month

flex

1 class

1.5 hr

mehrere bei

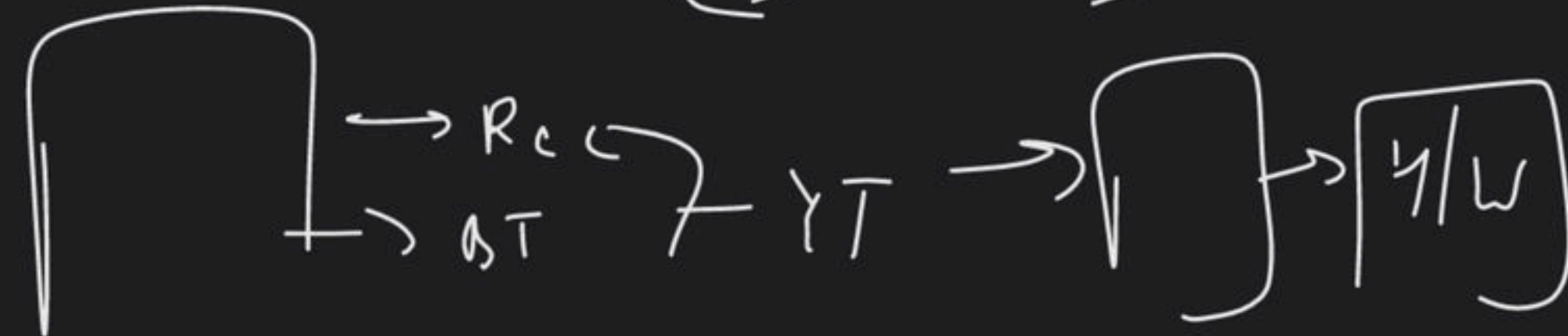
2 hr +

reality

1.5 hr +

Mon/Used

Rec → 4 u →



ST L

