DnC Level-5

Special class

Love Babbar • Mar 25, 2023

Jove

$\rightarrow$ Generate Parenthesis

$\textcircled{n} = \underline{\underline{1}} \longrightarrow$

($\rightarrow$ 1

) $\rightarrow$ 1

ans $\rightarrow$ $\boxed{(\,)}$ $\downarrow$ 1

) ($\downarrow$

$\underline{\underline{n=2}}$

$(\longrightarrow 2$ ✓

$) \longrightarrow 2$ ✓

$$\text{ans} \rightarrow \underbrace{\boxed{(()) \quad , \quad ()()}}_{2}$$

$\boxed{n = 3}$

Total bracket → $\boxed{n*2}$

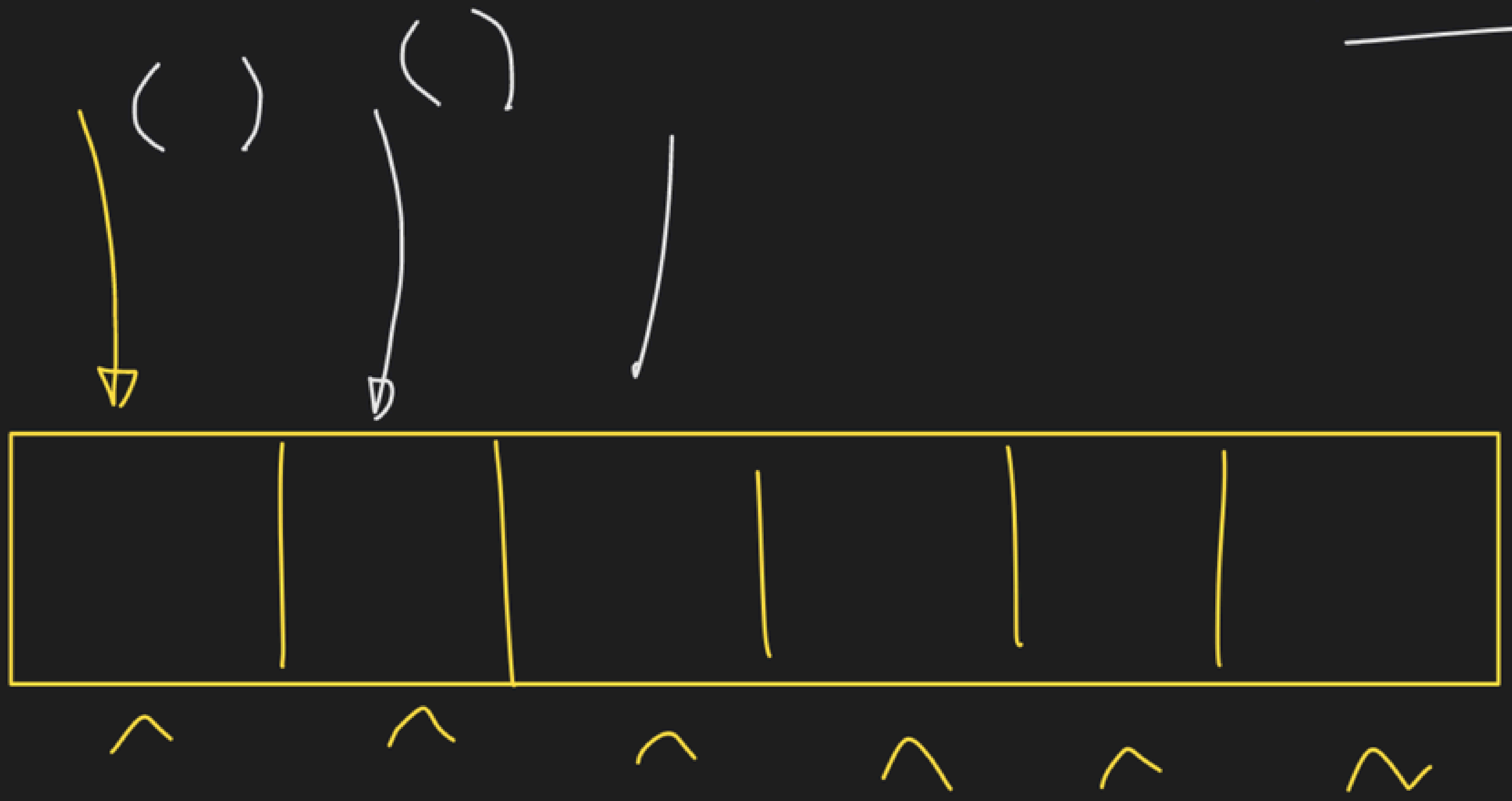
(→ 3

) → 3

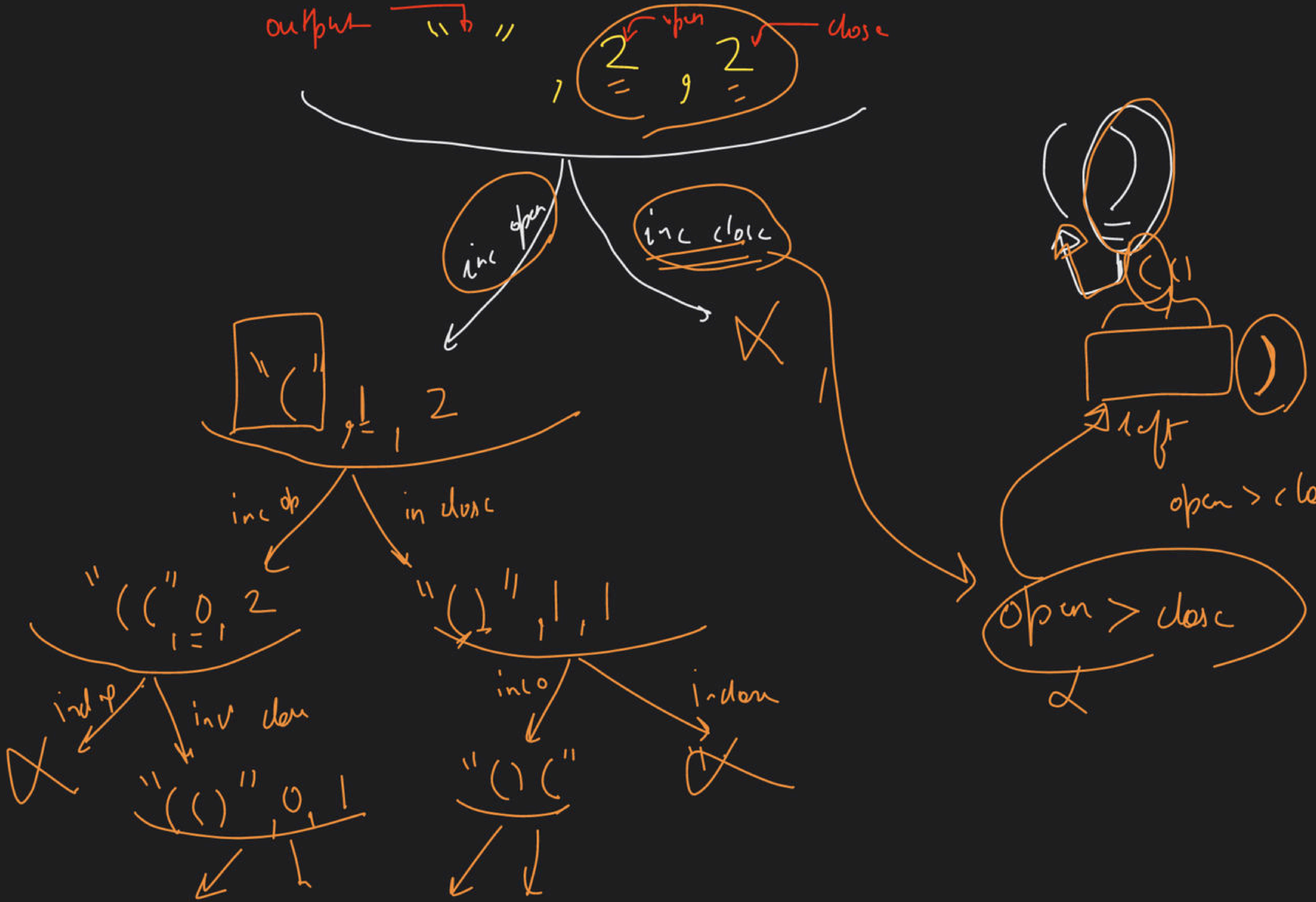
ans → ((())) , (()()) , ()()() , ()(()) , (())()

$\boxed{5}$

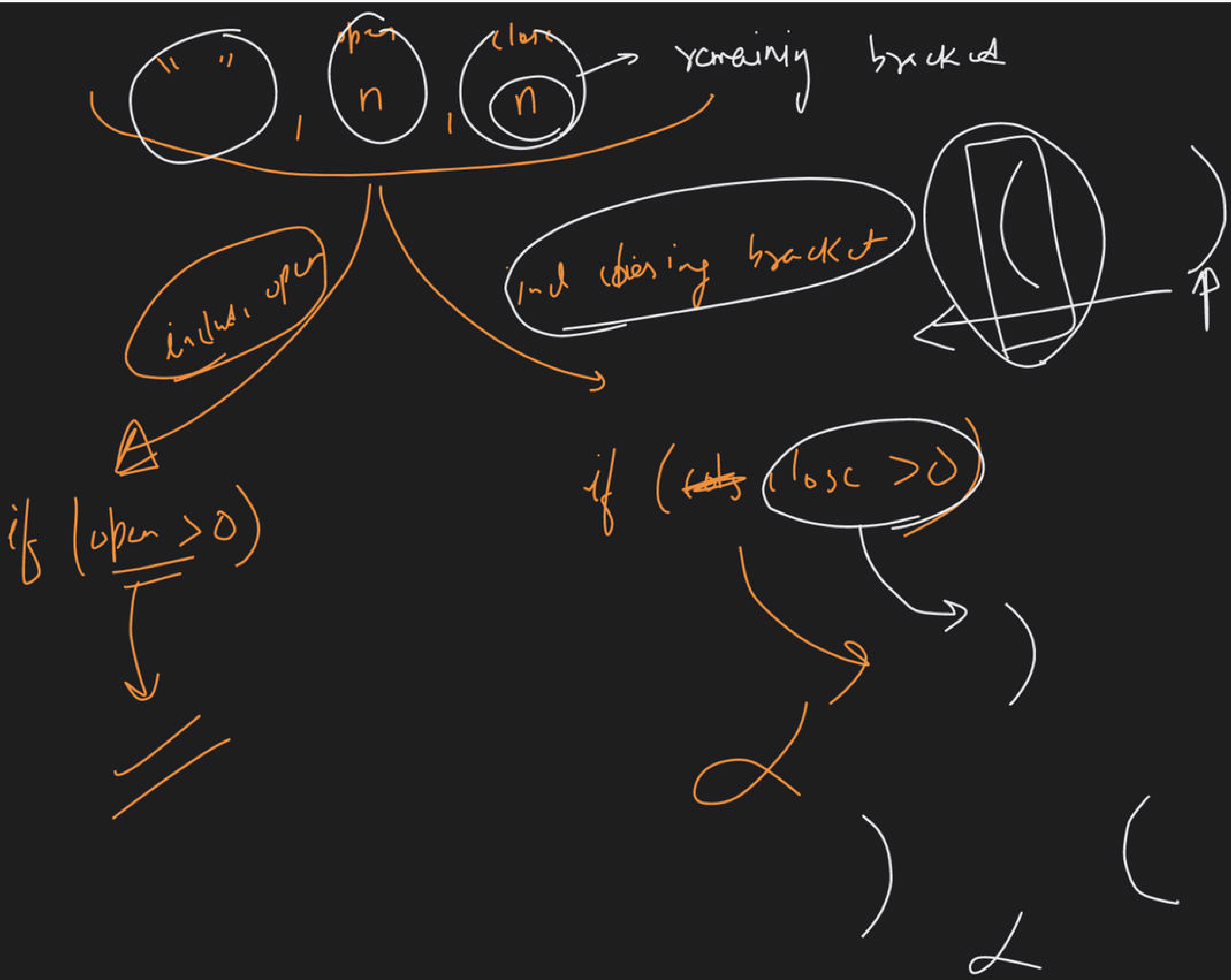
include / exclu



n=2



output "" , 2 , 2
open
close
inc open
inc close
"(" , 1 , 2
inc op
in close
"((" 0 , 2
"()" , 1 , 1
inc op
inv clou
inc o
i close
"(()" , 0 , 1
"()("
open > close
left
open > close
(( )

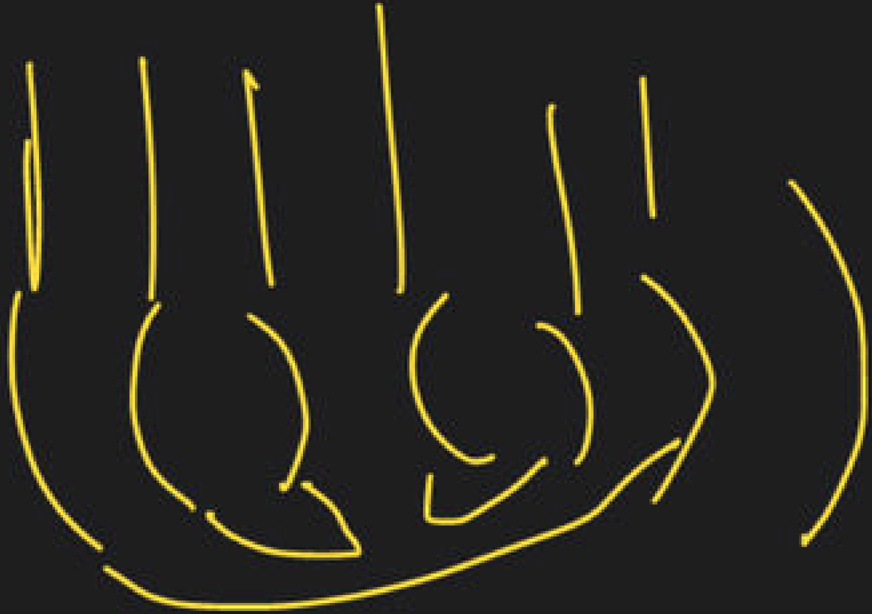


" "
open
n
close
n
remaining bracket
include open
incl closing bracket
if (open > 0)
if (close > 0)

$(a+b)$

$((x+2y+9))$

$) \qquad ($

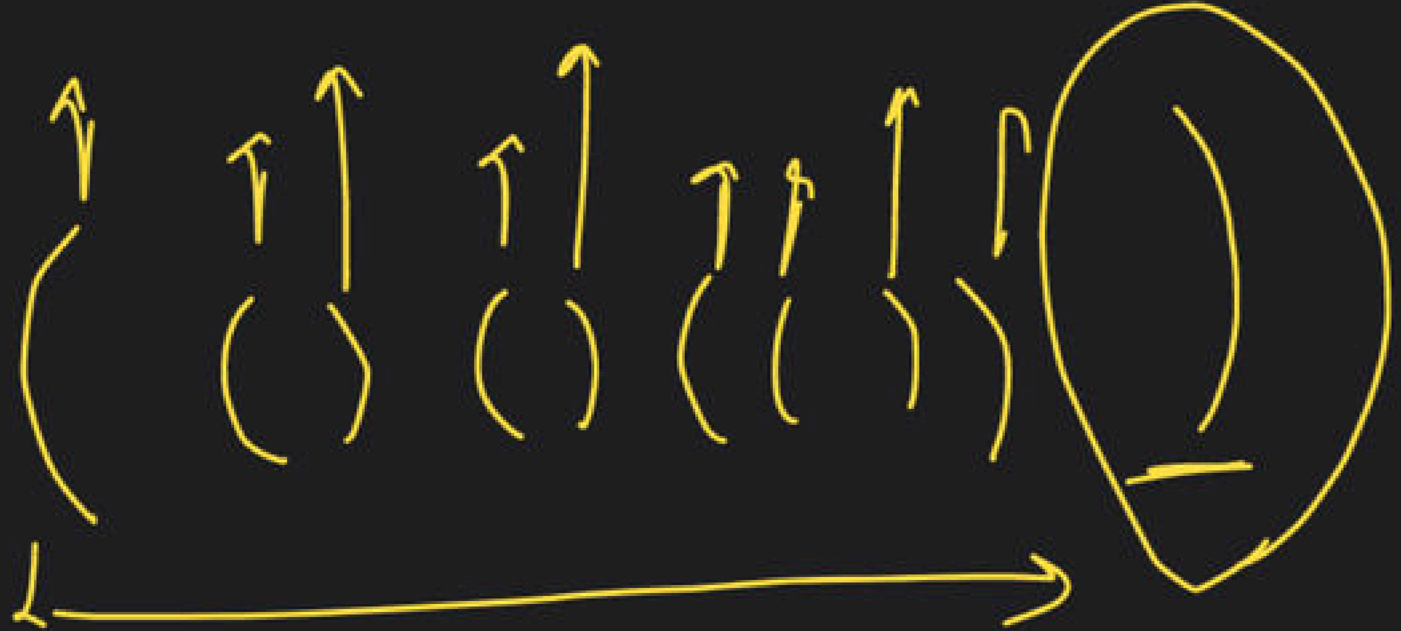


open = 3

close = 3

$\downarrow$

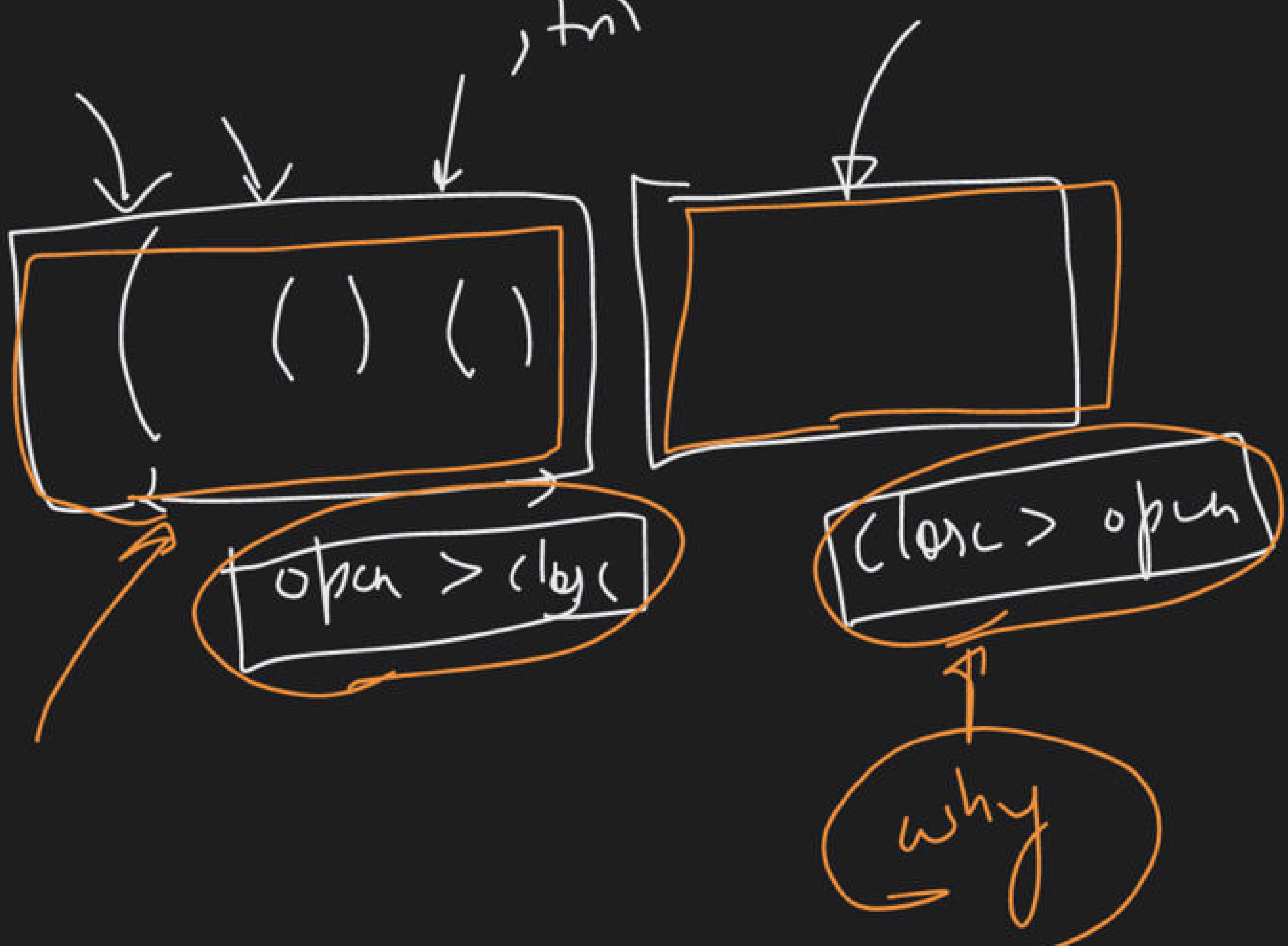
~~closing~~



open = 1

close = 4

open > close $\rightarrow$ clear



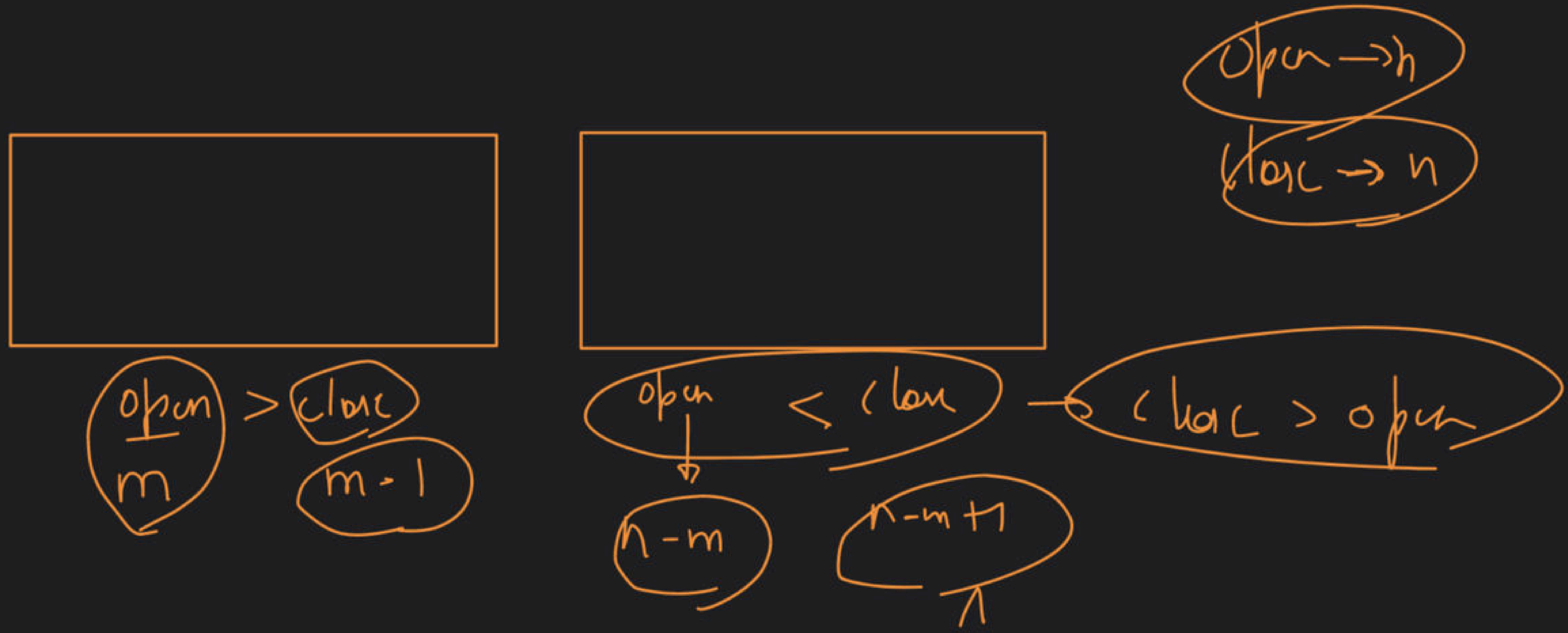
Rahu

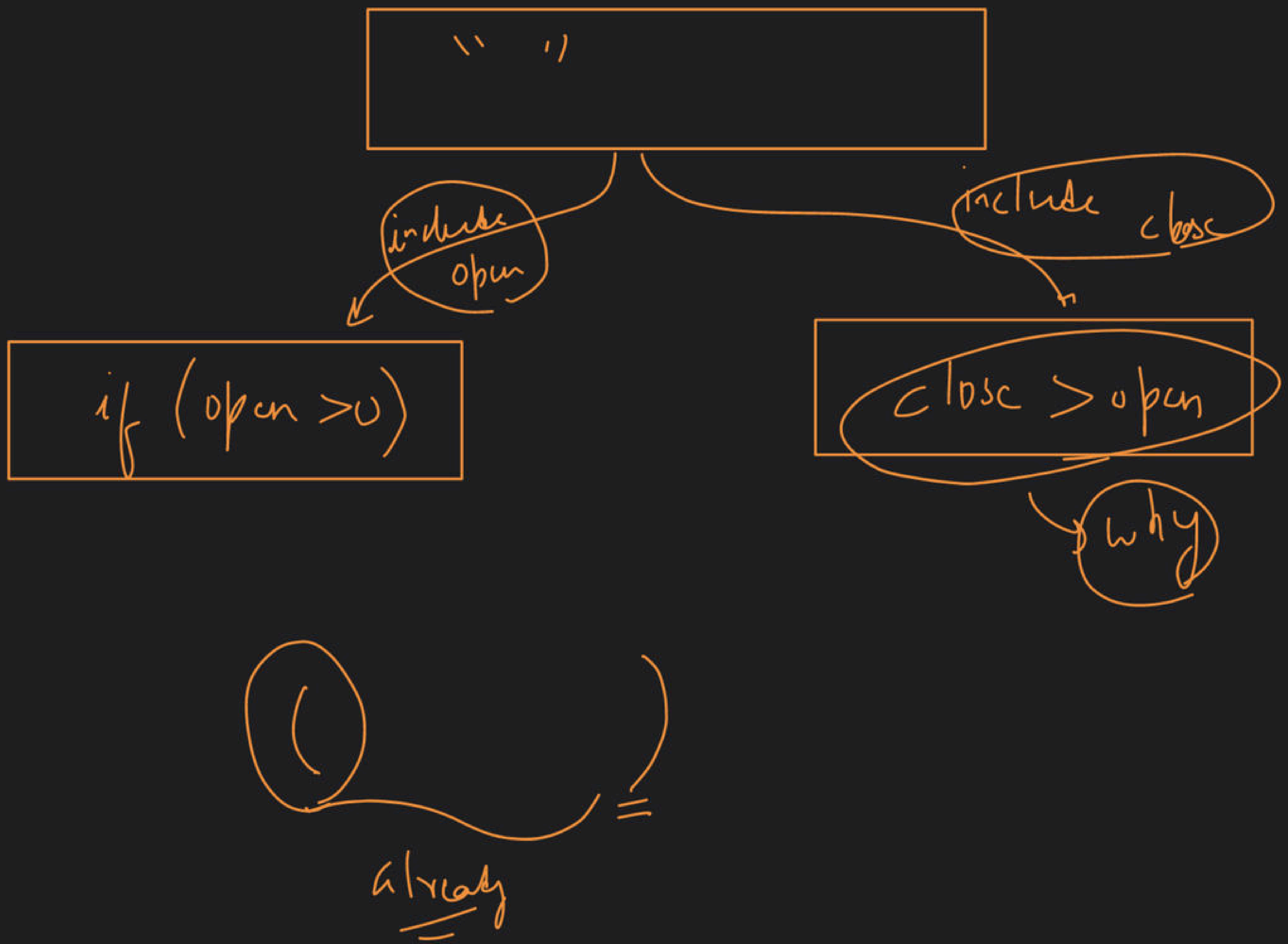
Nitin → 4

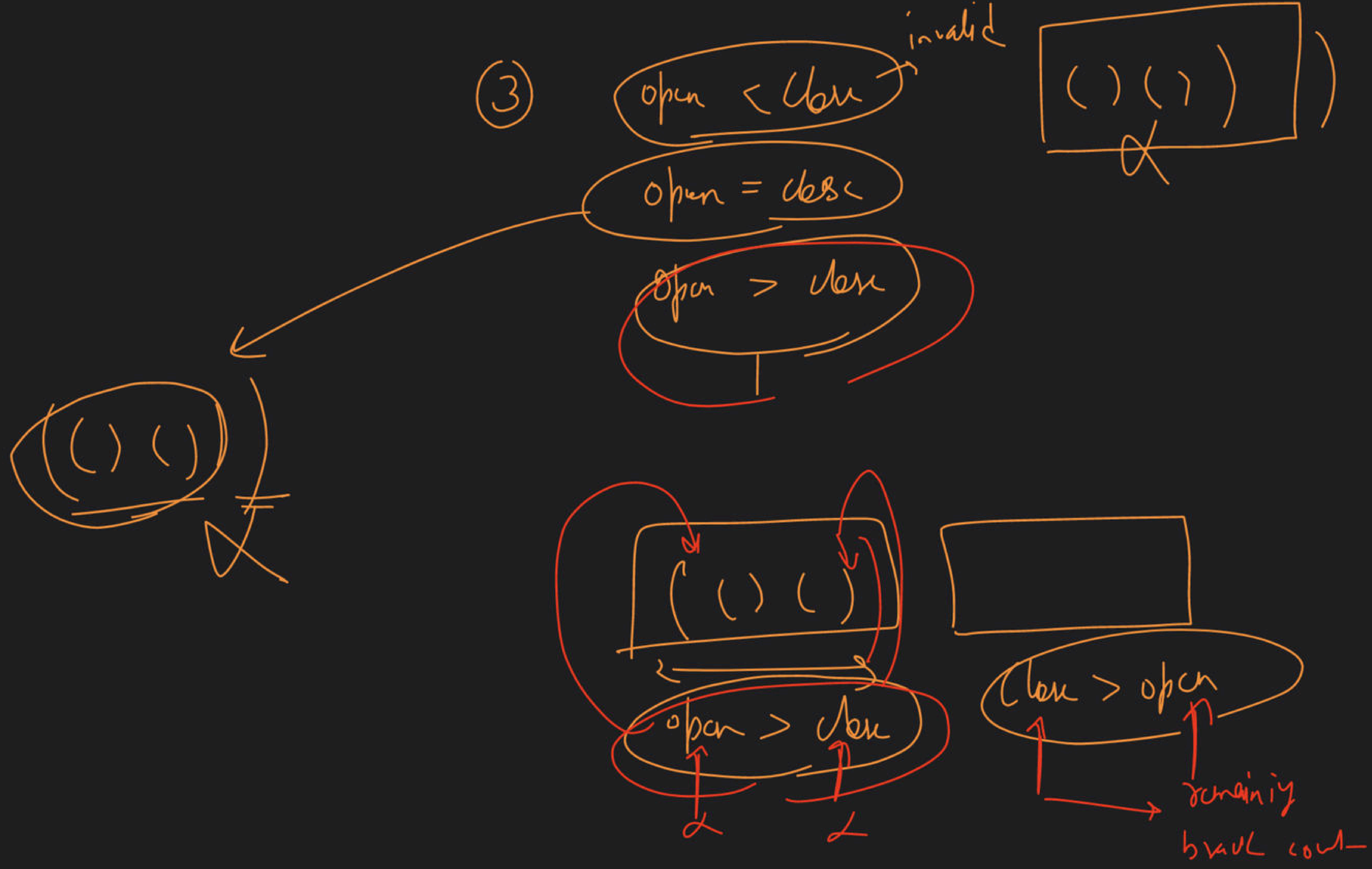
Love → 4

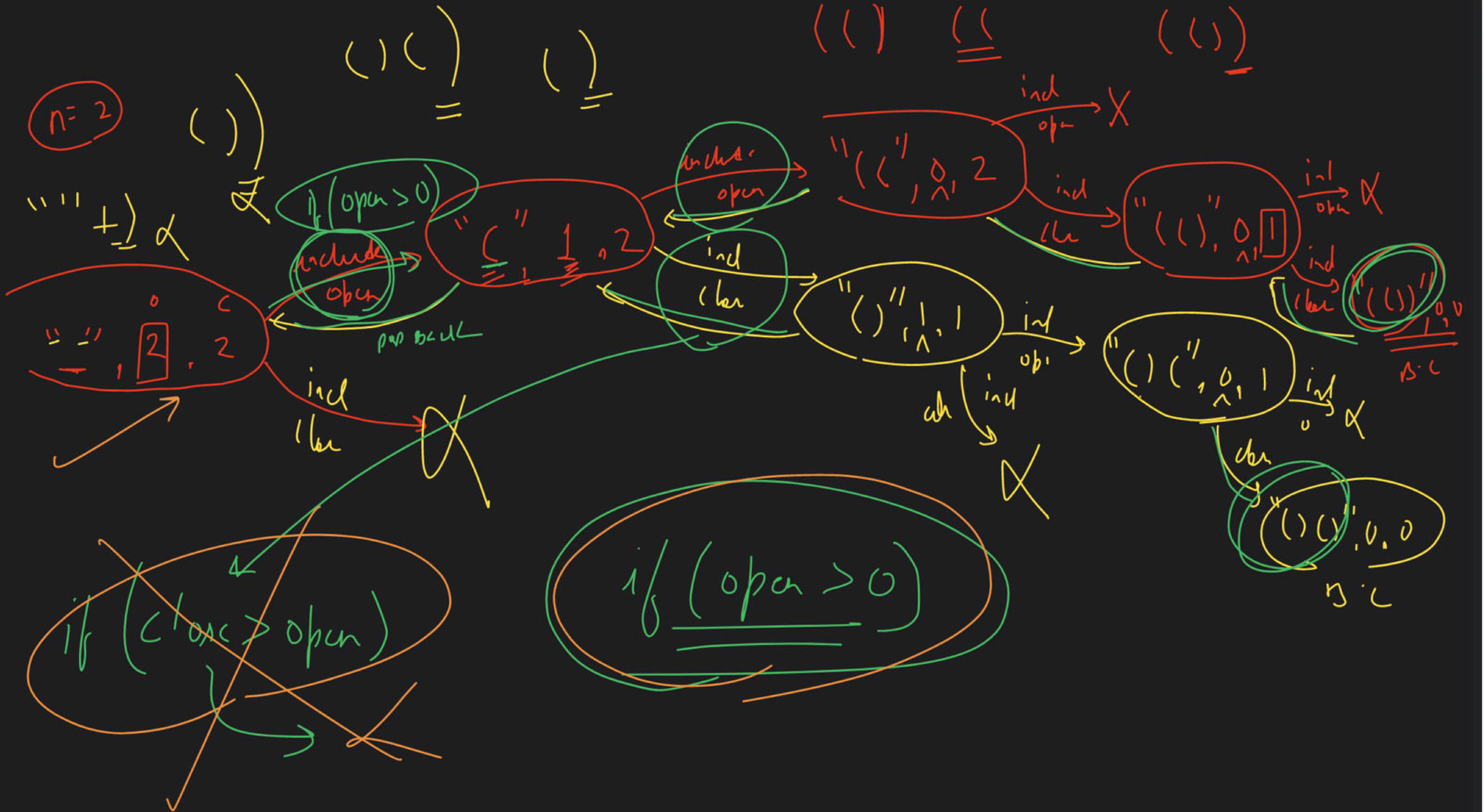
$$\begin{array}{cc} 2 & 1 \\ \end{array}$$
$$\boxed{N > L}$$

$$\begin{array}{cc} 2 & 3 \\ \end{array}$$
$$\boxed{N < L}$$


open > close
m
m - 1
open < close
n - m
n-m+1
close > open
open → n
close → n

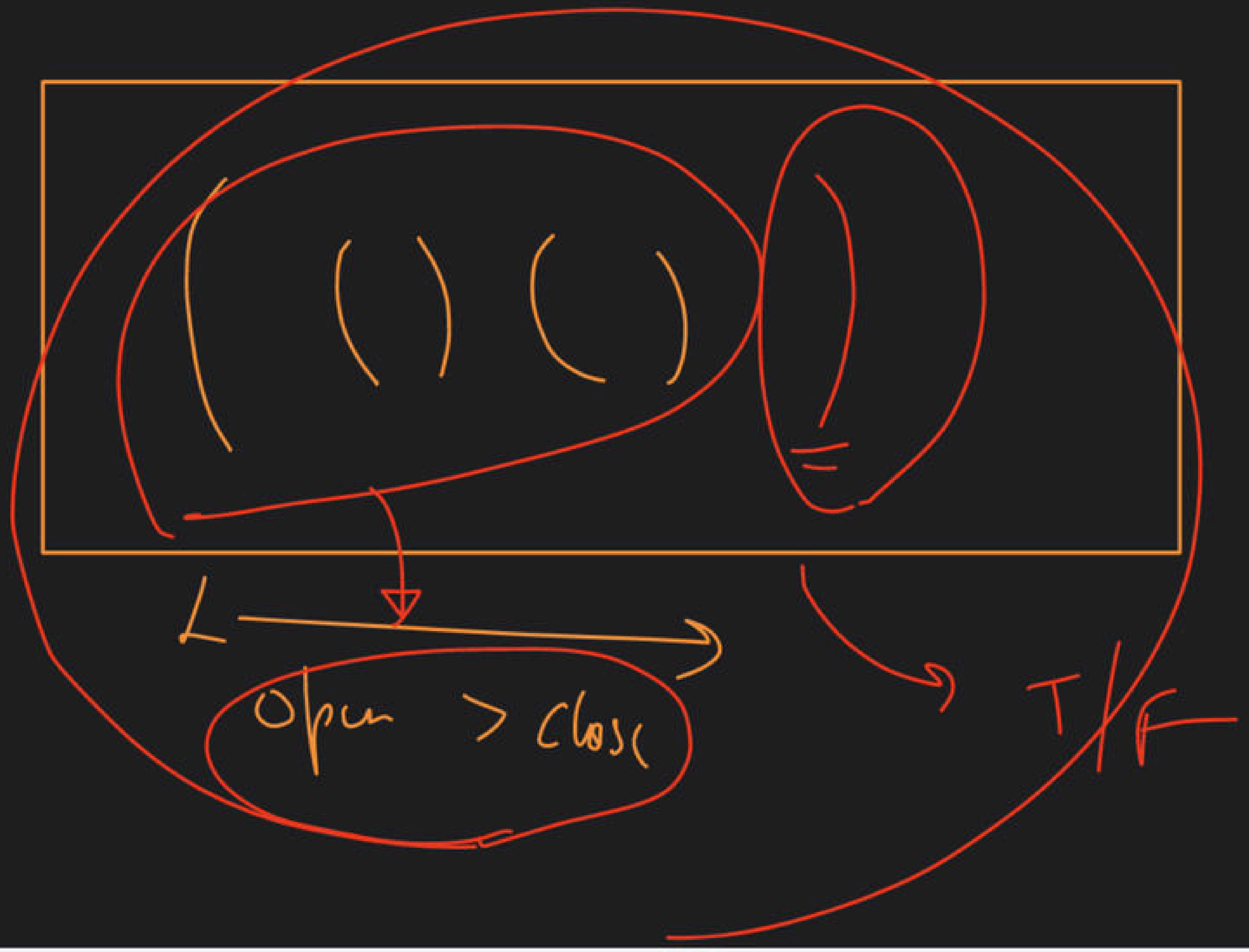

" "
include open
if (open > 0)
include close
close > open
why
(
)
already
=

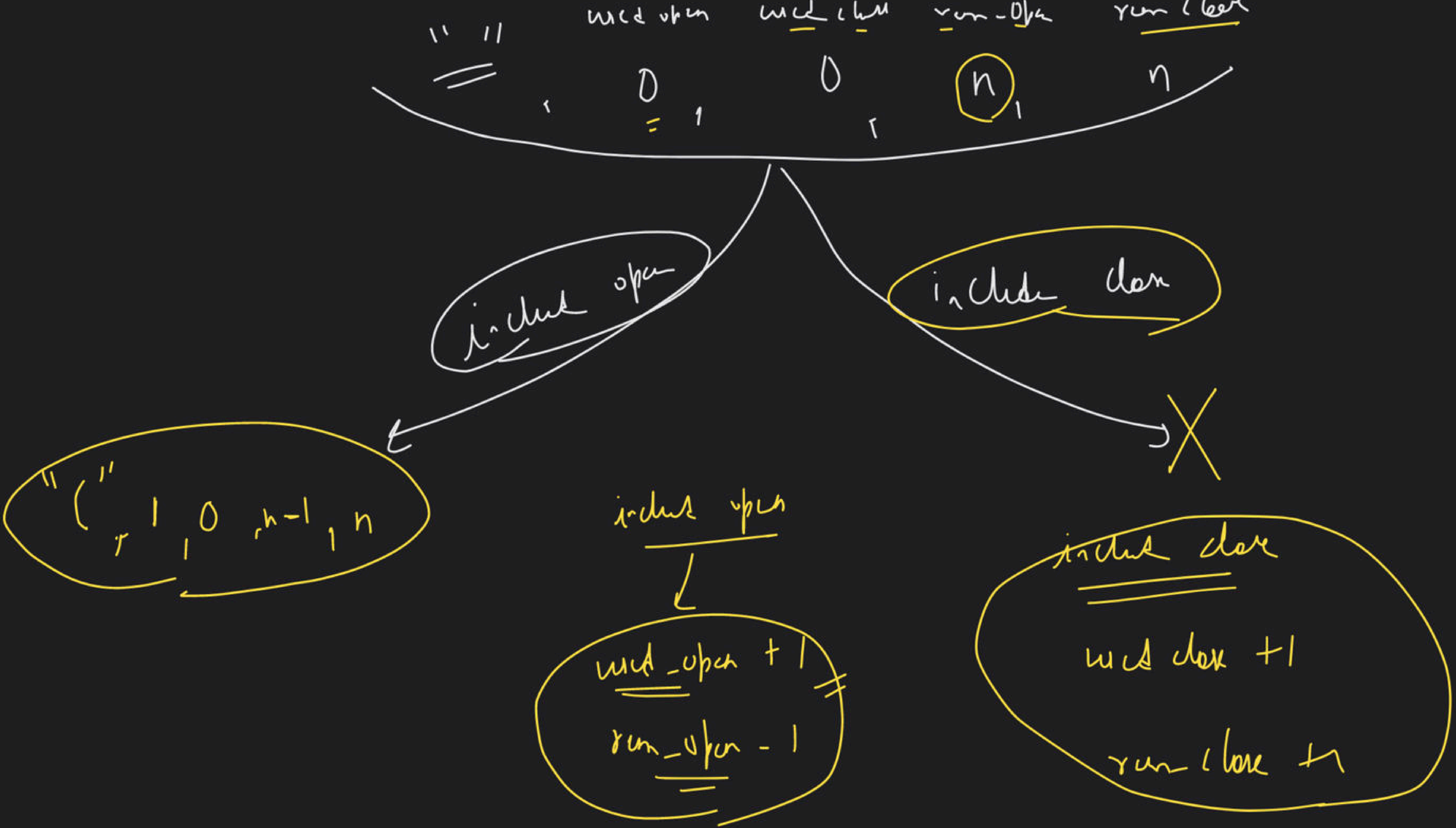

3
open < close
invalid
open = close
open > close
open > close
close > open
remaining
brack count


n= 2
( ))
( ) ( )
( )
(( )
((
(( ))
"" +) X
"" , 2 , 2
if (open > 0)
include open
pop back
"(" , 1 , 2
incl close
"((" , 0 , 2
incl open
incl close
"(()" , 0 , 1
"(())" , 0 , 0
B.C
"()" , 1 , 1
"()(" , 0 , 1
"()()" , 0 , 0
B.C
if (close > open)
if (open > 0)

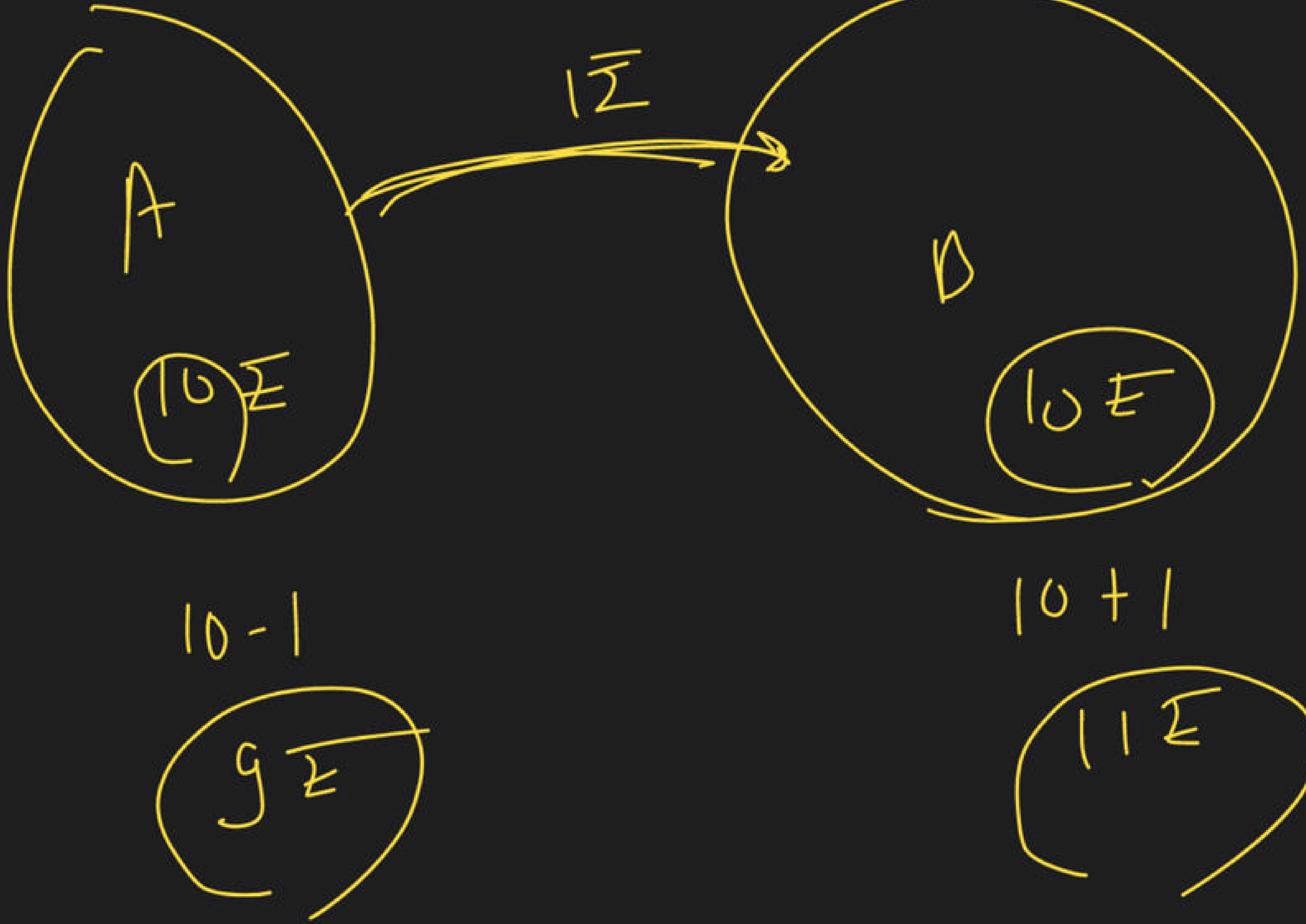
remaining opening bracket = n

remaining closing bracket = n





""
used open
used close
rem_open
rem close
"" , 0 , 0 , n , n
include open
include close
"(" , 1 , 0 , n-1 , n
include open
used_open +1
rem_open - 1
include close
used close +1
rem close +1

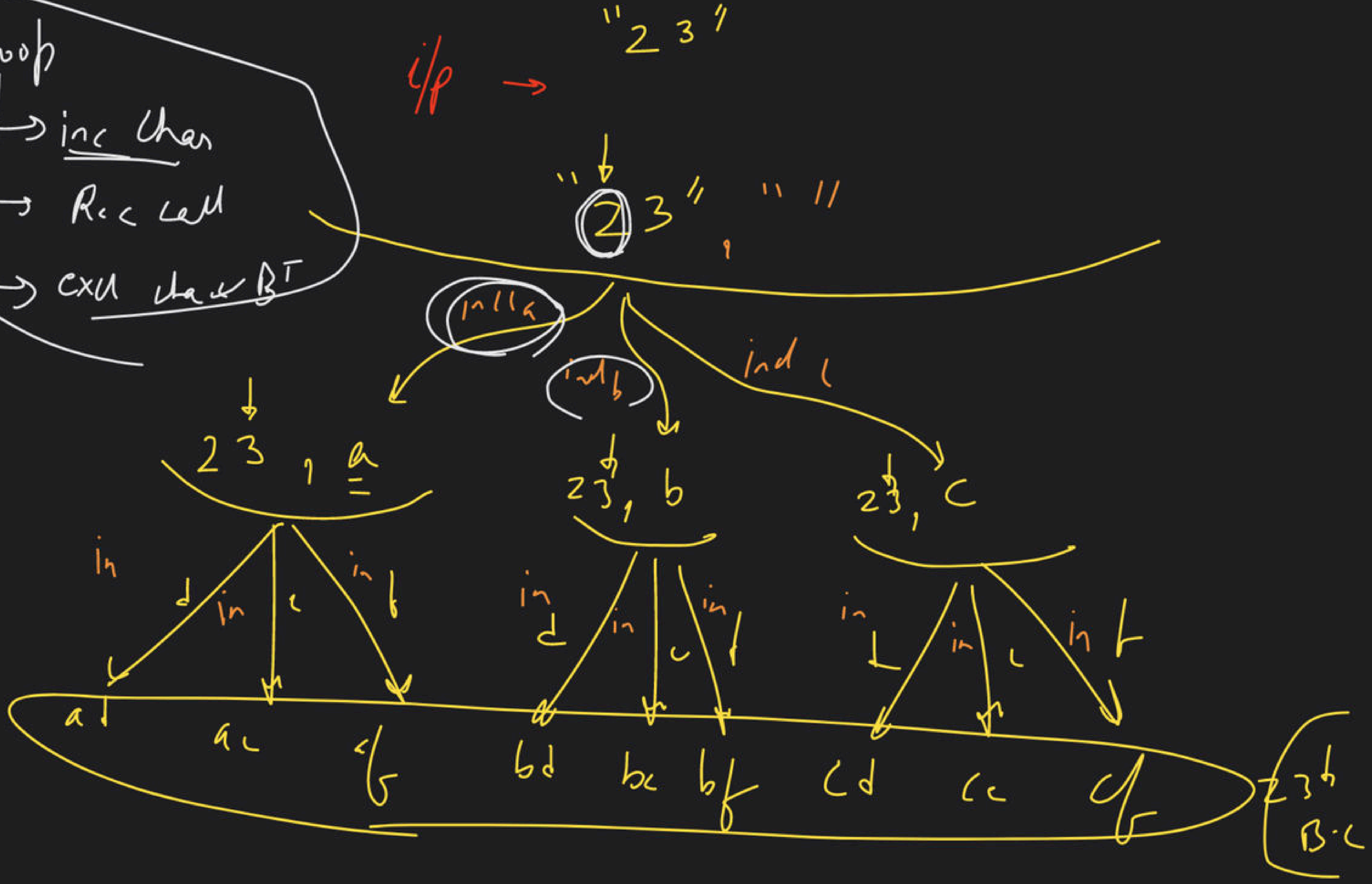

1E
A
10E
B
10E
10-1
9E
10+1
11E

mapping

2 →	"abc"
3 →	"def"
4 →	"ghi"
5 →	"jkl"
6 →	"mno"
7 →	"pqrs"
8 →	"tuv"
9 →	"wxyz"

array map

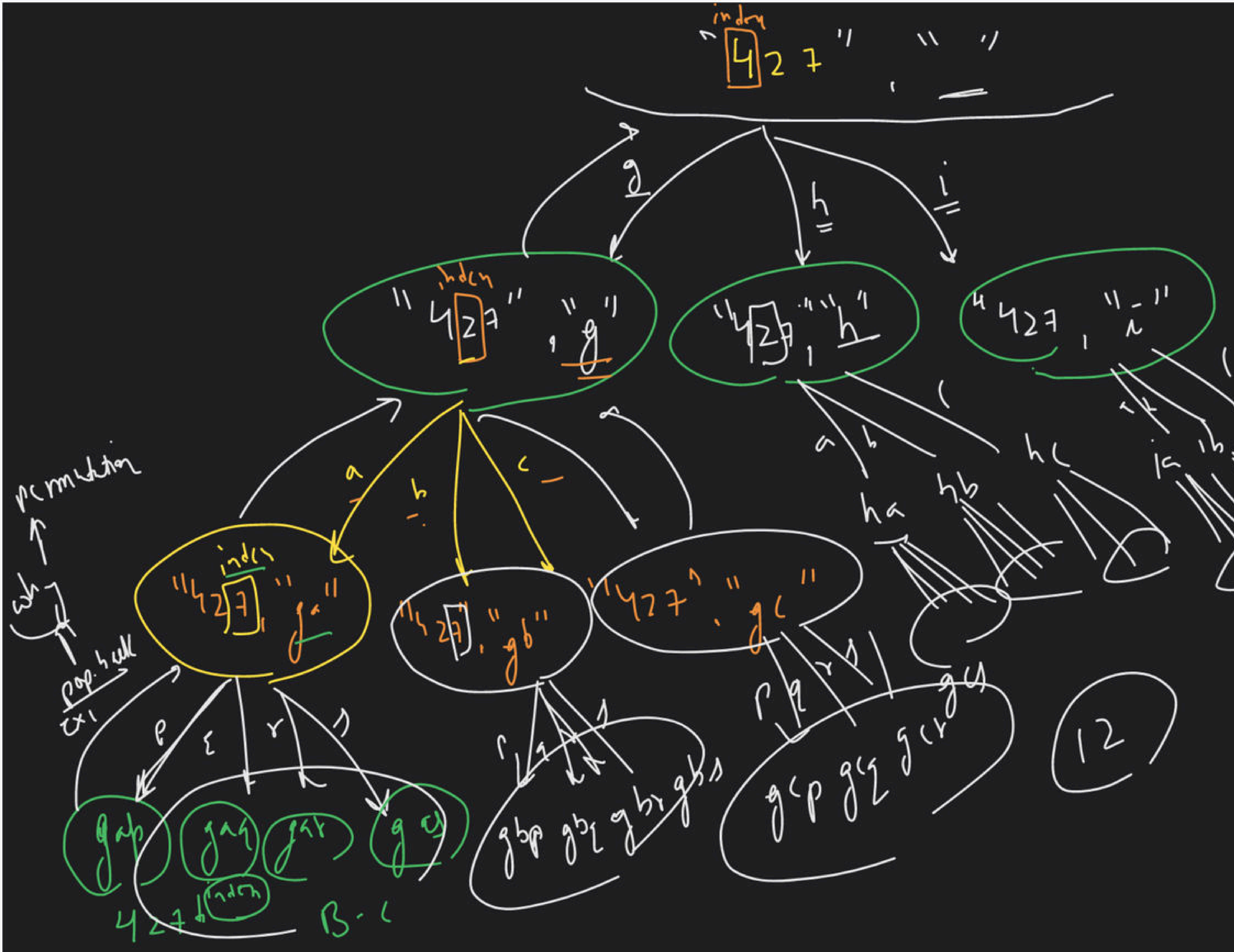
```
loop
 └→ inc char
 └→ Rec call
 └→ excl char BT
```



```
main ( )
 └→ vector <string> ans
 └→ int index = 0
 └→ string output = " "
 └→ ~~ba~~vector <string> mapping
                   ^
 └→ solve ( ________ ) ;

solve ( )
{
    //Base case
    if (i >= n)
    { ans.push_back(output)
        return;
    }

    int digit = (digits (index)) - '0';
gh  string value = mapping (digit]
    for ( ________ )
    {  //inc
       //Rec
       //BT
    } }
```



index = 0

n = 3

0 >= 3

r K

int digit

= digits[index] - '0'

= ~~'9'~~ - '0'

digit = ~~9~~ ← int

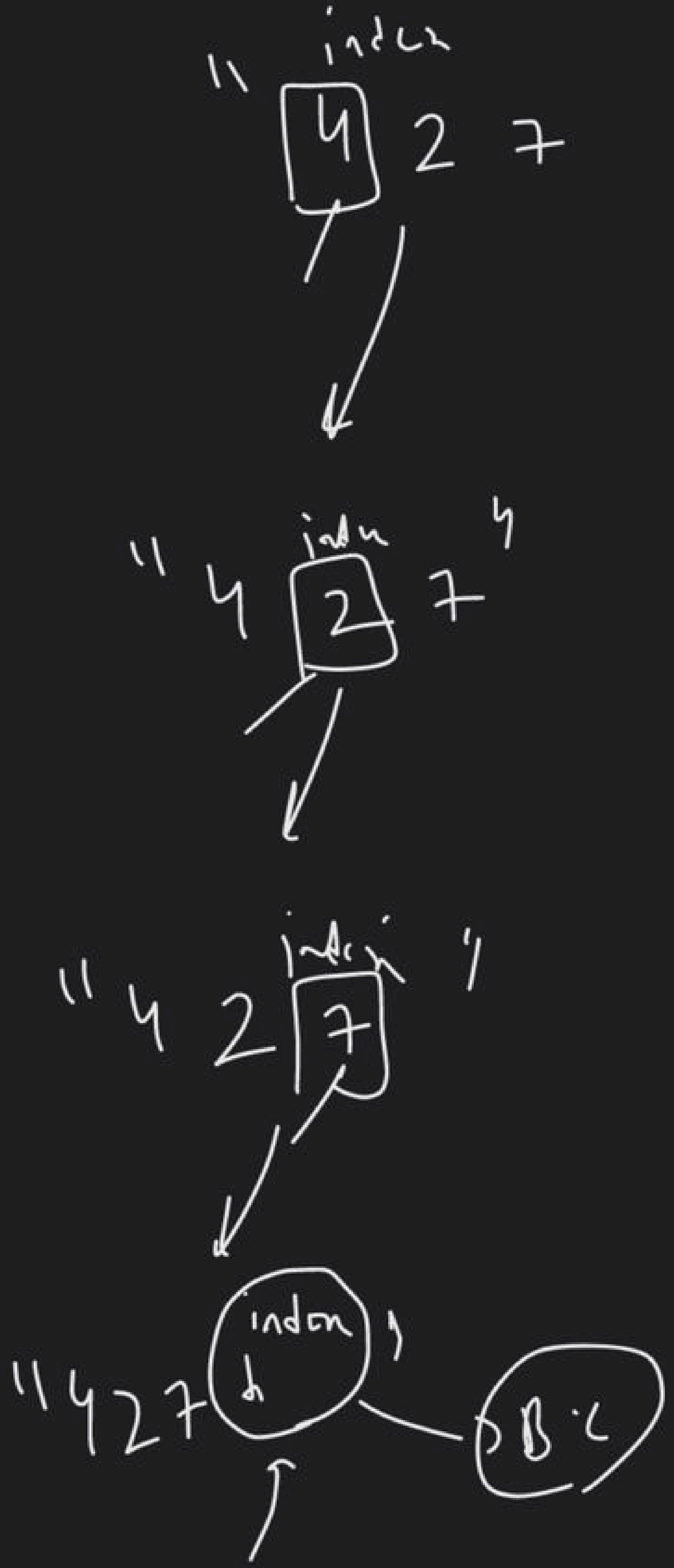
string value =

mapping(digit)

= mapping[~~9~~]

pqrs

string value = "~~ghi~~" ~~'abc'~~


index
" 4 2 7
index
" 4 2 7 "
index
" 4 2 7 "
index
"427
B.C

→ letter tile Combination

→ No: of Square full arrays

→ word - break - 1

→ Sum of all subset XOR