Recursion - III

Foundation Course on Data Structures & Algorithm - Part I

Love Babbar • Lesson 21 • May 28, 2022

→ i/p → string = $\boxed{\text{"abc"}}$

n = 3

P.S → $2^3 = \boxed{8}$

o/p → Power Set = {}, a, b, c, ab, bc, ac, abc

$\boxed{60\%}$

Approach → ? Recursion

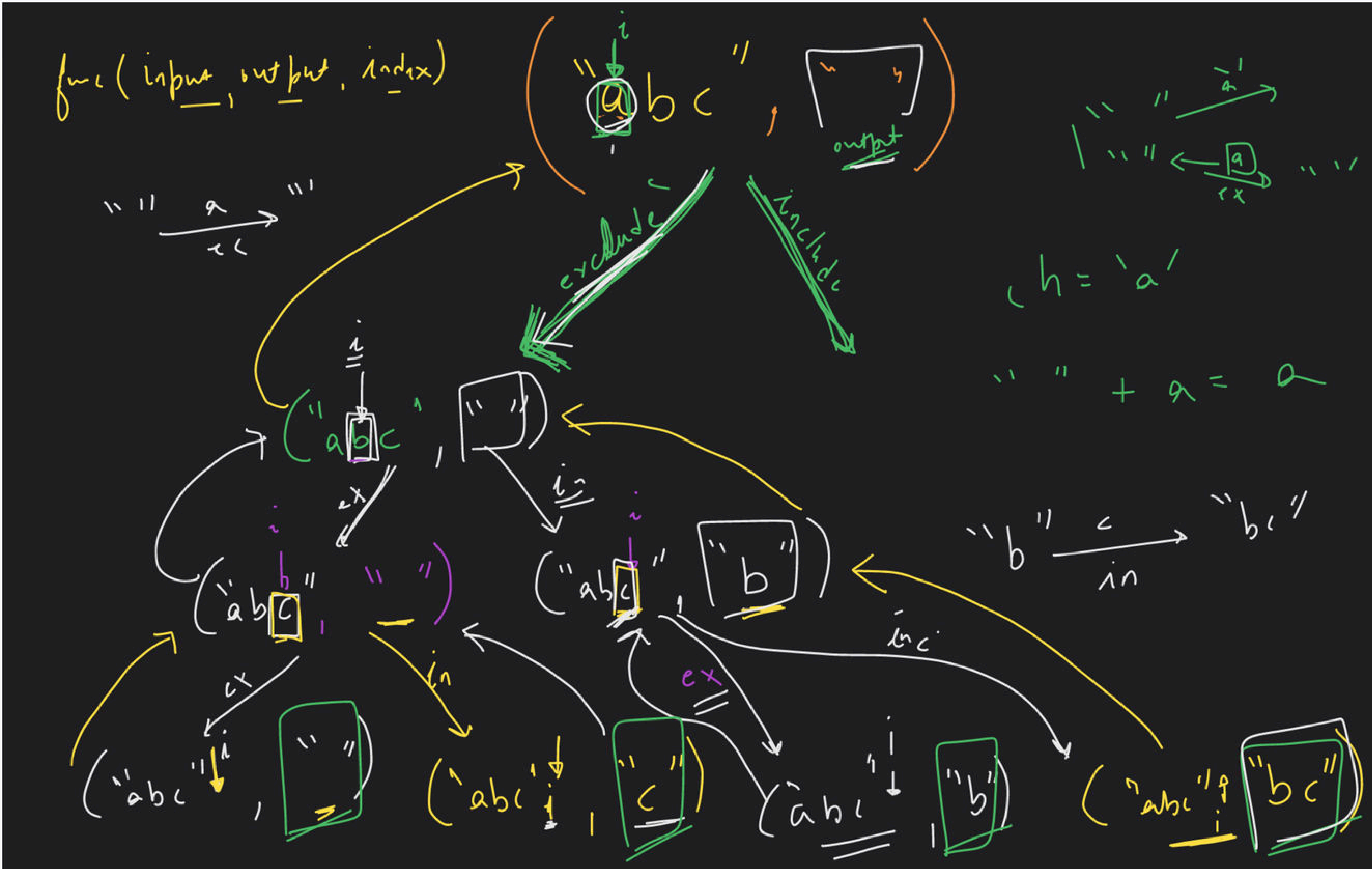
→ i/p → string = "abc"

o/p → Power Set → {{ }, a, b, c, ab, bc, ac, abc} =

include / exclude → Yes or No

a	b	c		
✓	✗	✗	→ a	—
✗	✓	✗	→ b	✓
✗	✗	✓	→ c	✓
✓	✓	✗	→ ab	✓
✗	✓	✓	→ bc	✓
✓	✗	✓	→ ac	✓
✓	✓	✓	→ abc	✓ ... ✓
✗	✗	✗		

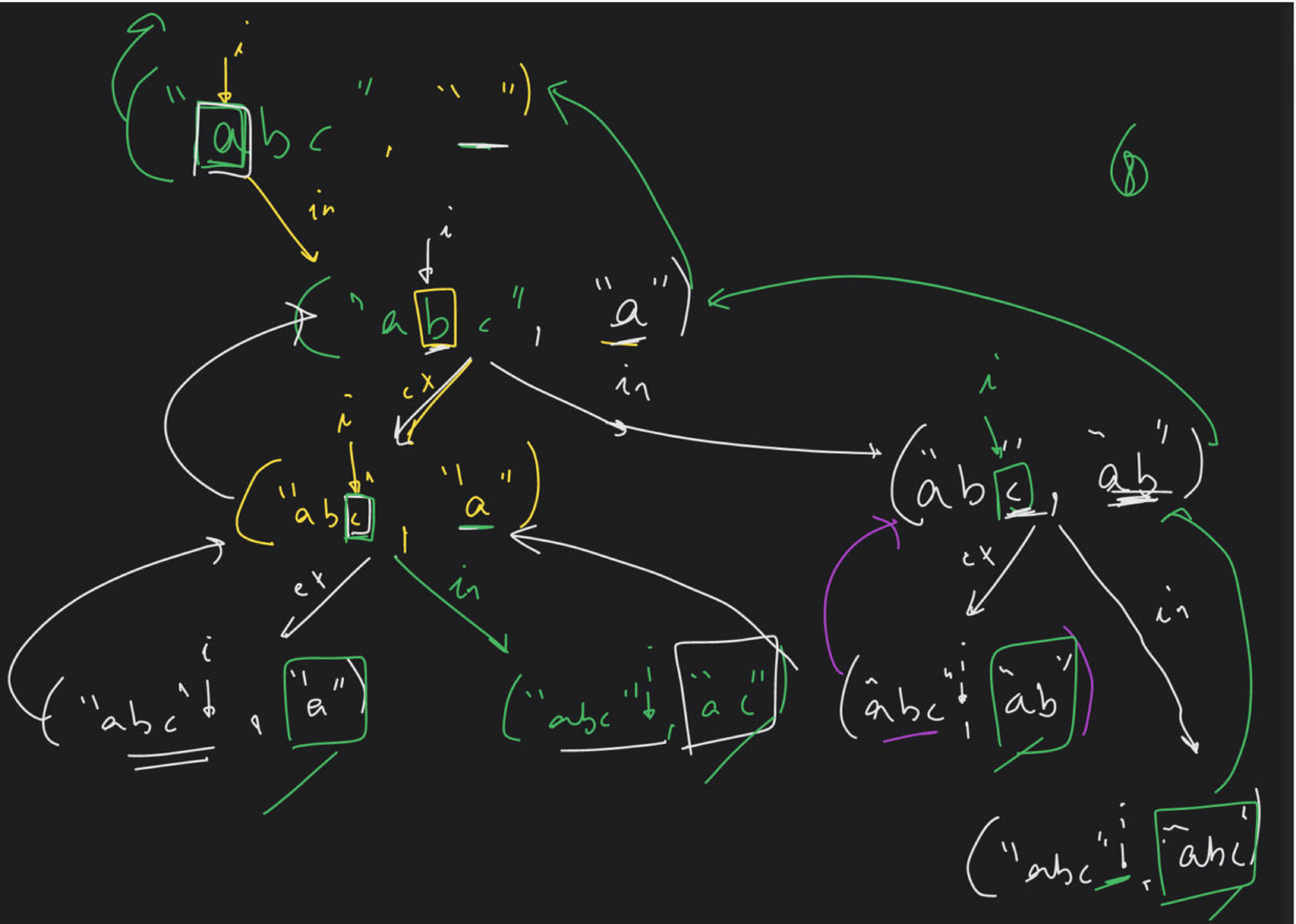
func( )
{
func( )
func( )
}



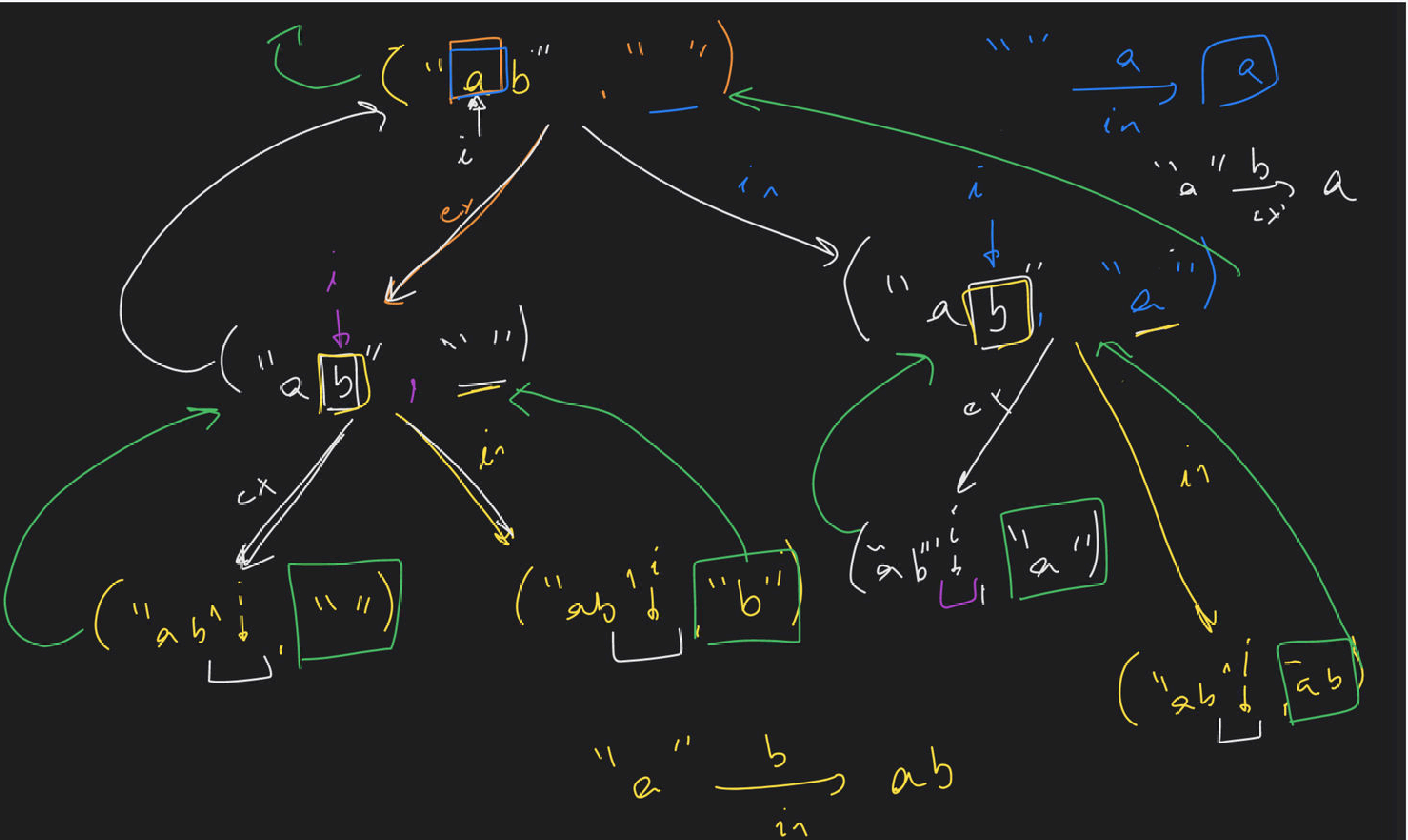
output = ""

Base case

```
if(i == input.length)
{
    cout<<output;
    return;
}
```


8
i
("abc", "")
in
i
("abc", "a")
ex
in
i
("abc", "a")
i
("abc", "ab")
ex
in
ex
in
i
("abc", "a")
i
("abc", "ac")
i
("abc", "ab")
i
("abc", "abc")

string = "ab"   op → $\underbrace{a \quad b \quad ab \quad \{\}}$



("ab", "")
i
ex
in
("ab", "")
i
("ab", "a")
i
ex
in
ex
in
("ab", "")
("ab", "b")
("ab", "a")
("ab", "ab")
"" $\xrightarrow[in]{a}$ a
"a" $\xrightarrow[ex]{b}$ a
"a" $\xrightarrow[in]{b}$ ab

→ Combination in a String of digit

i/p → "123"

o/p → 1 _ 2 _ 3

1 _ 23

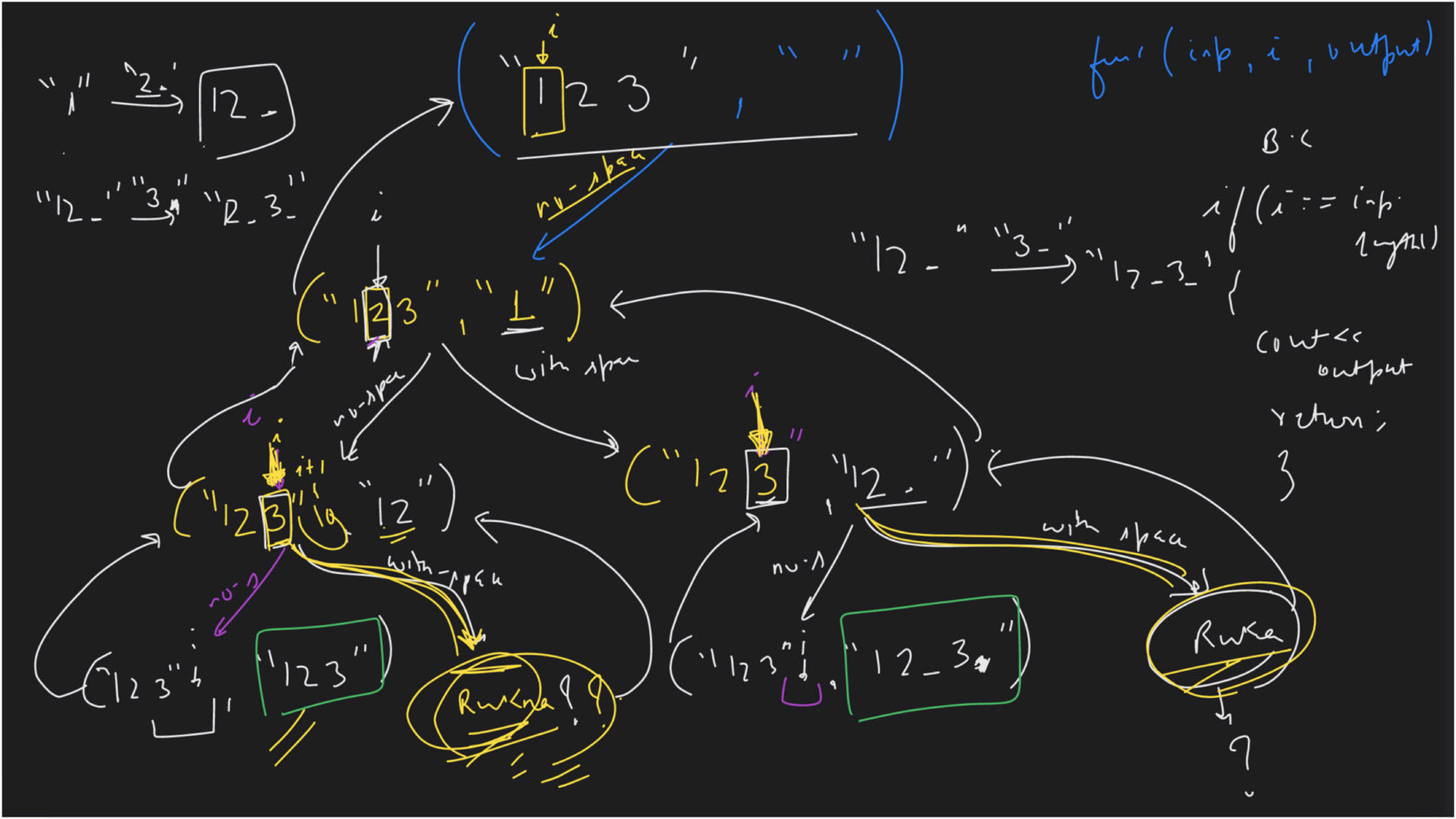
12 _ 3

123

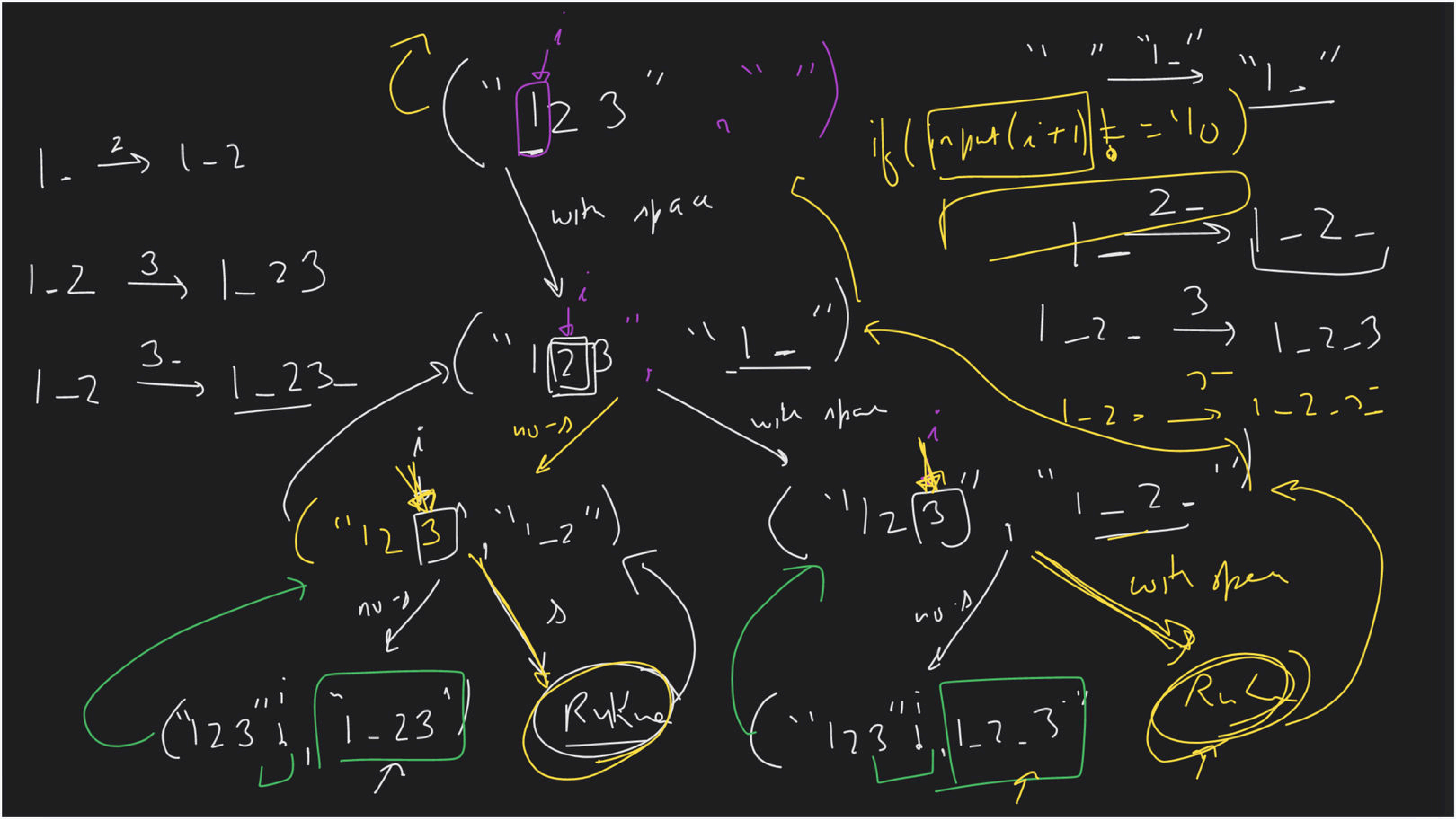
→ observation ?

① space daale

② space nahi daale

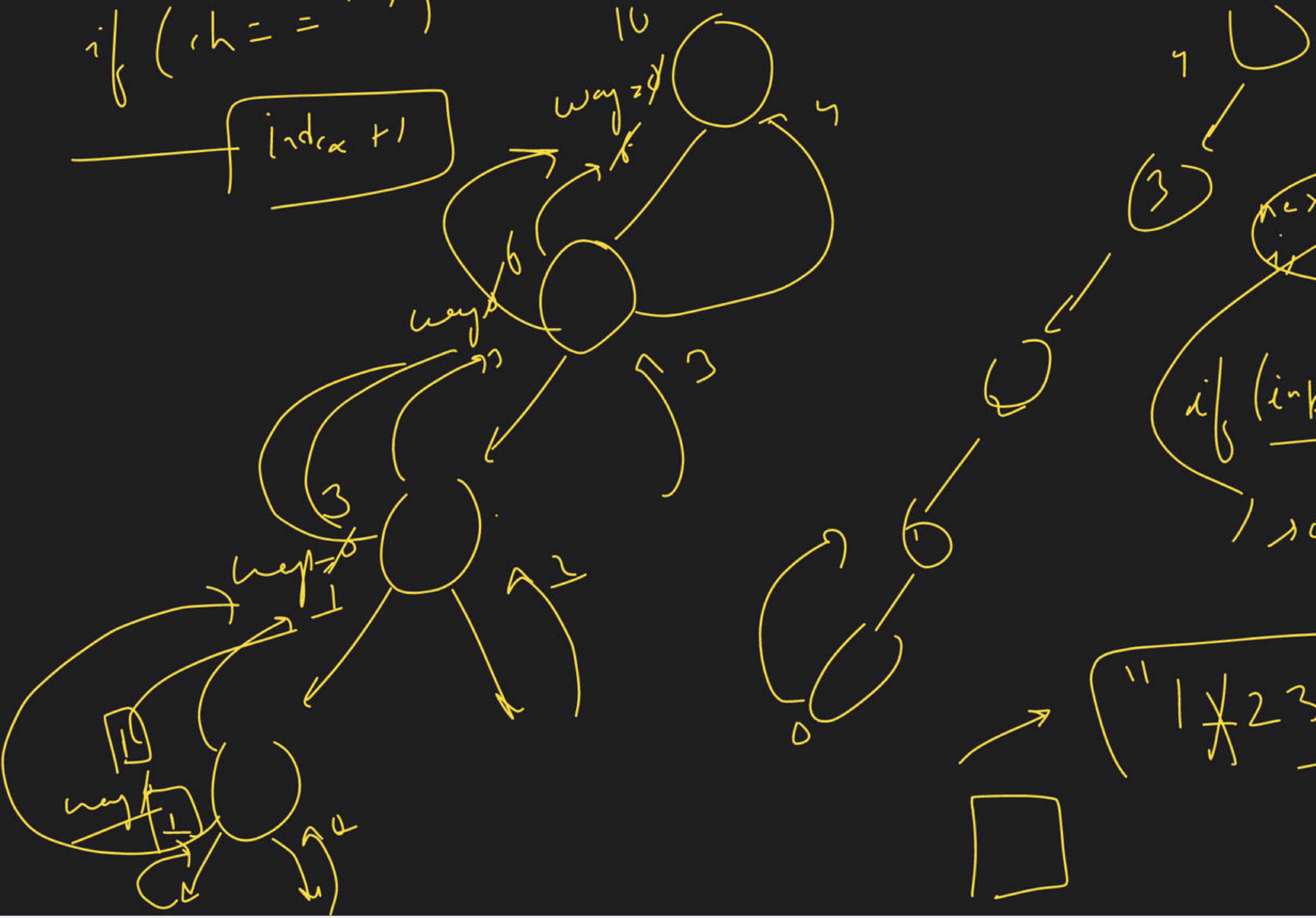


```
fun(inp, i, output)

B.C

if (i == inp.length())
{
    cout << output
    return;
}
```


("123", "")
i
with space
("123", "1_")
i
no-s
with space
("123", "1_2")
i
("123", "1_2_")
i
no-s
s
no-s
with space
("123"i, "1_23")
Return
("123"i, "1_2_3")
Ret
1_ → 2 → 1_2
1_2 → 3 → 1_23
1_2 → 3_ → 1_23_
"" → "1_" → "1_"
if(input(i+1) != '\0')
1_ → 2_ → 1_2_
1_2_ → 3 → 1_2_3
1_2_ → 3_ → 1_2_3_

if (ch == ' ')

index+1

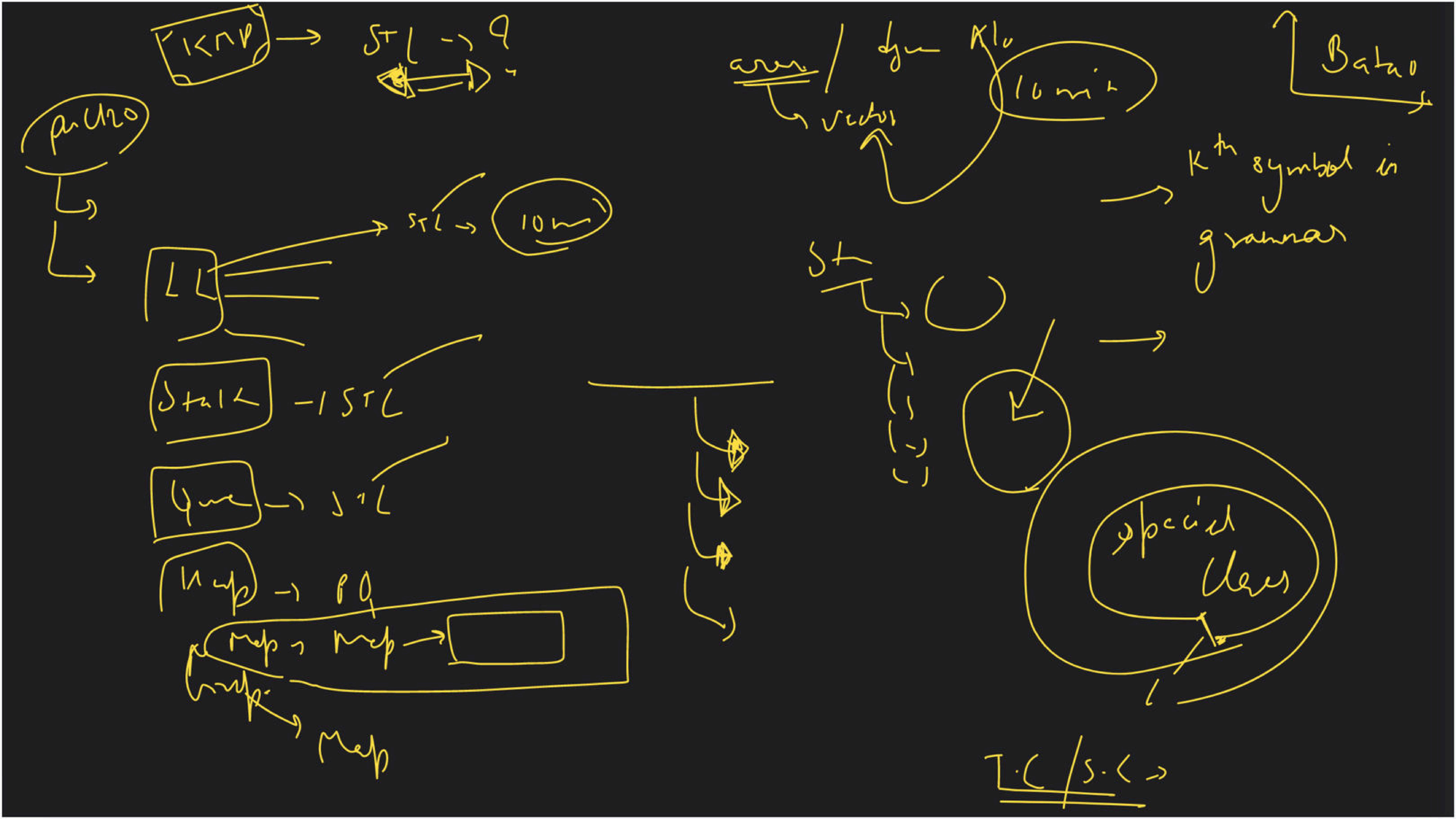


next character is valid ~~character~~

if (input[i+1] != '\0')

second Recursion call

"1 * 23"


KMP → STL → ?
ArU20
LL
STL → 10 min
Stack → STL
Queue → STL
Map → PQ
Map → Map
Umap
Map
Array / dyn Alloc
vector
10 min
Batao
Kth symbol in grammar
Str
special Cases
T.C / S.C →

→ find all even binary sequences (2n) with same sum of first & second half bits

i/p → $\boxed{n = 2}$

now-1

o/p →

0101   0110

1111   0000

1001   1010

Approach

$\boxed{n=1}$

00

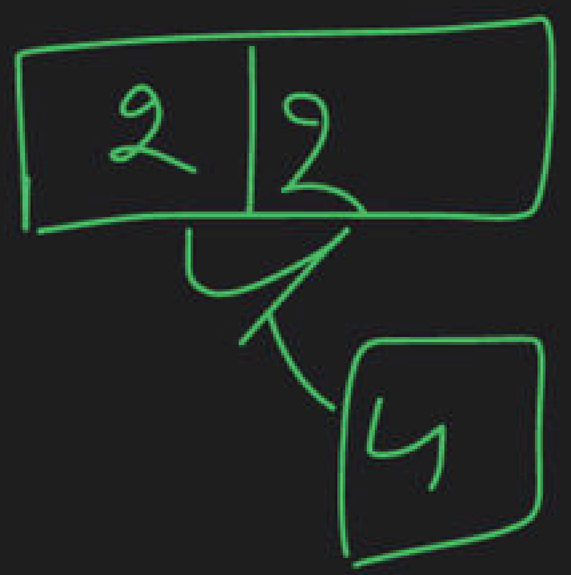
01 ✗

10 ✗

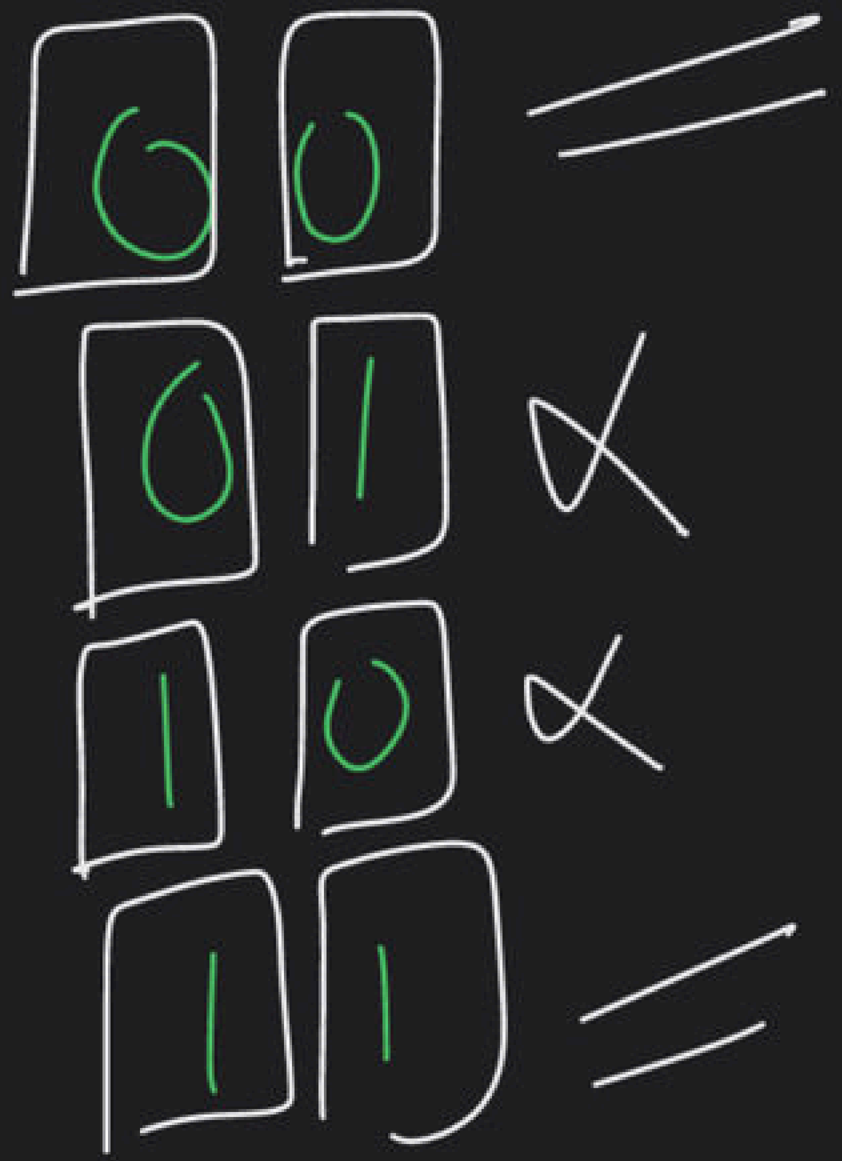
11

→ $n = 1$ → $\{00, 11\}$

$2n$ → string → $2n = 2 \times 1 = 2$

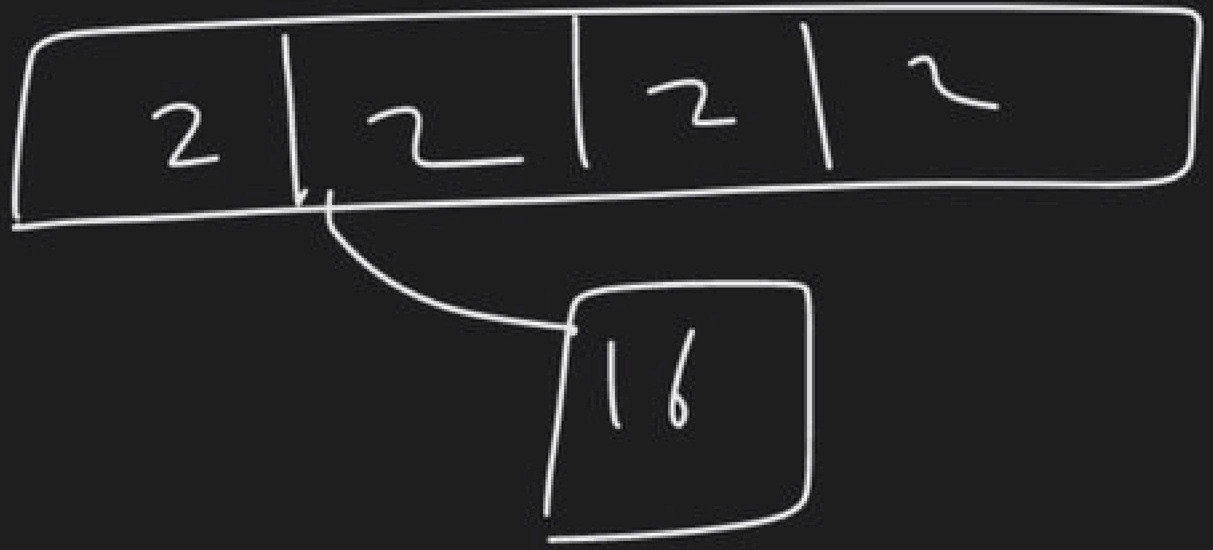


0/1



→ n = 2

str → 2×n = 2×2 = 4



0/1

0 0 0 0 → ✓

0 0 0 1 ✗

0 0 1 0 ✗

0 0 1 1 ✗

0 1 0 0 ✗

0 1 0 1 → ✓

0 1 1 0 → ✓

0 1 1 1 ✗

1 0 0 0 ✗

1 0 0 1 → ✓

1 0 1 0 → ✓

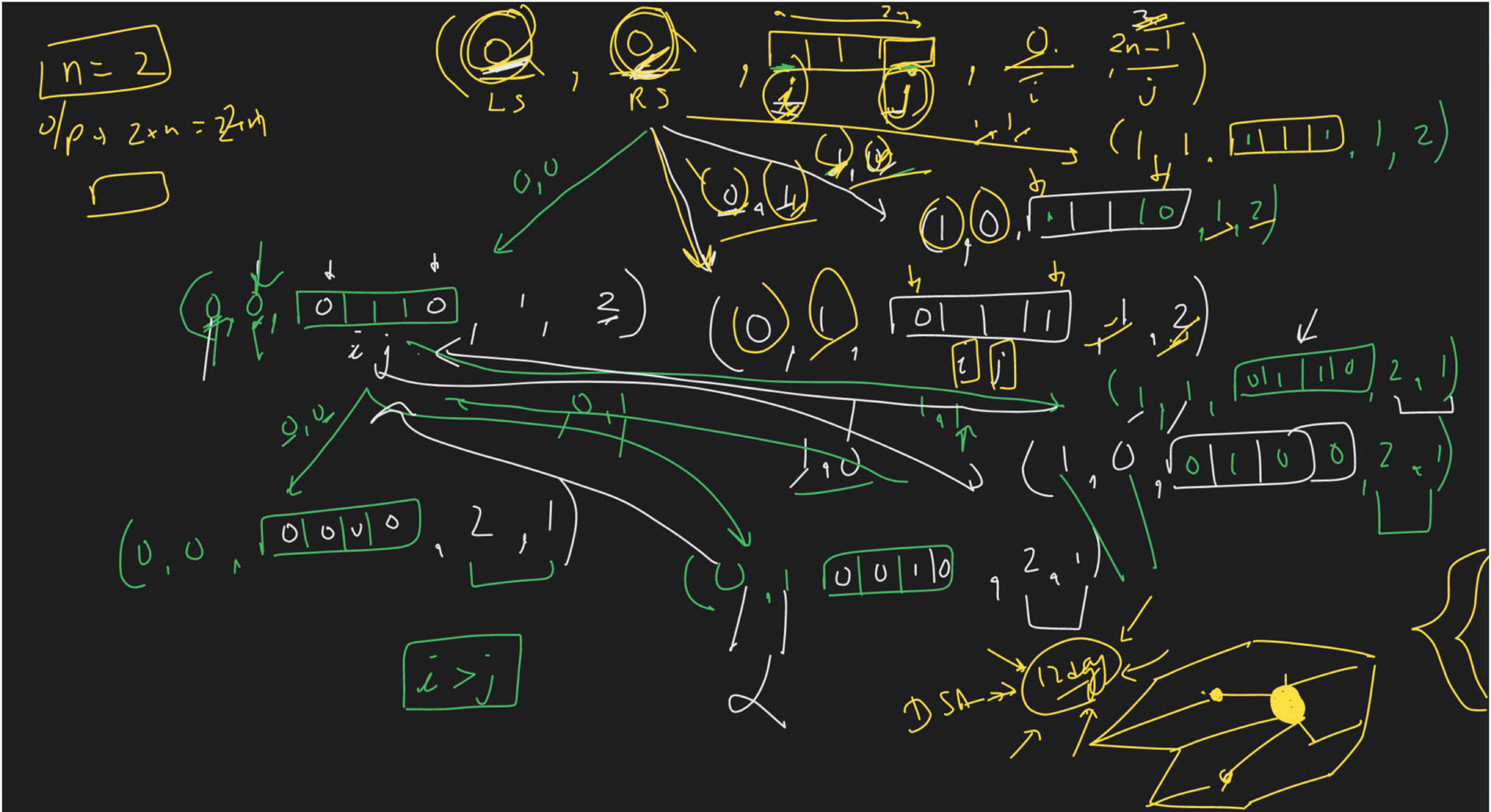
1 0 1 1 ✗

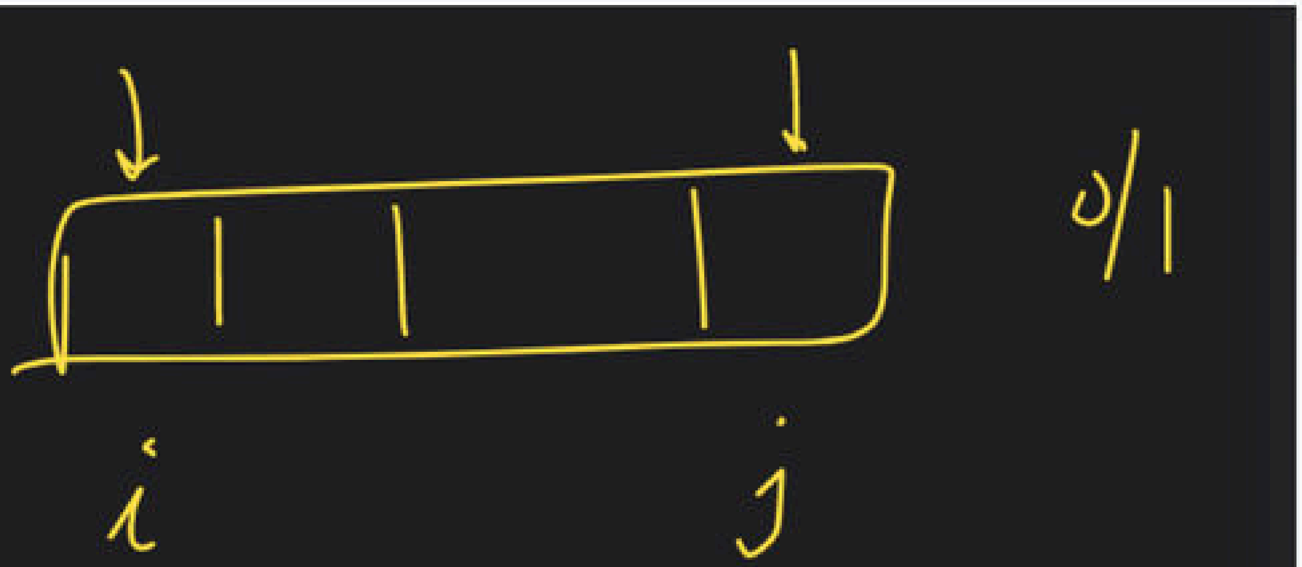
1 1 0 0 ✗

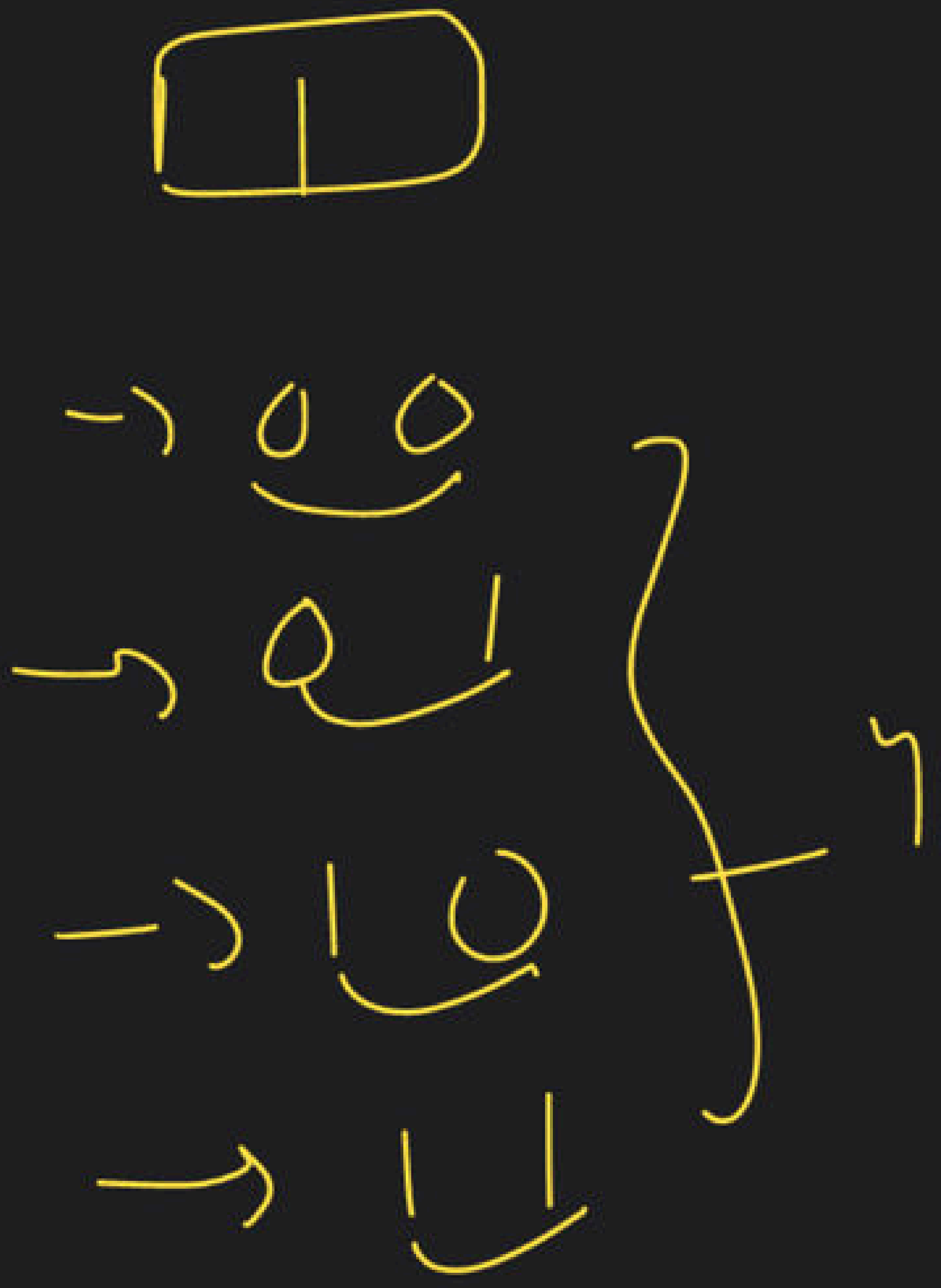
1 1 0 1 ✗

1 1 1 0 ✗

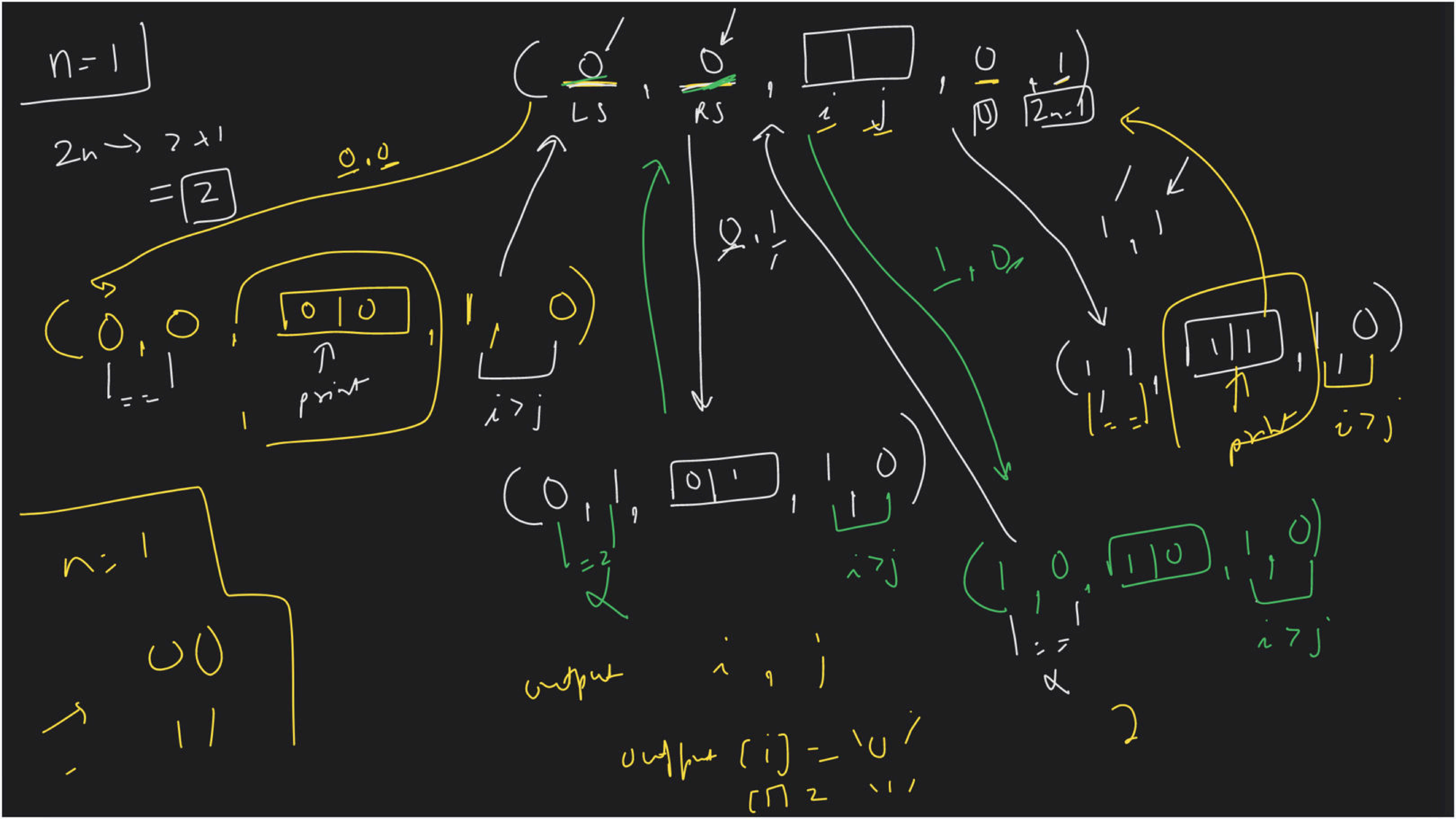
1 1 1 1 → ✓


n = 2
o/p → 2*n = 2*n
LS
RS
2n
0
i
2n−1
j
3n
(1, 1, 1|1|1|1, 1, 2)
0, 0
0, 1
1, 0
(1, 0, 1|1|1|0, 1, 2)
(0, 0, 0|1|1|0, 1, 2)
i j
(0, 1, 0|1|1|1, 1, 2)
i j
(1, 1, 0|1|1|0, 2, 1)
0, 0
0, 1
1, 0
1, 1
(1, 0, 0|1|0|0, 2, 1)
(0, 0, 0|0|0|0, 2, 1)
(0, 1, 0|0|1|0, 2, 1)
i > j
DSA
12day


0/1
i
j

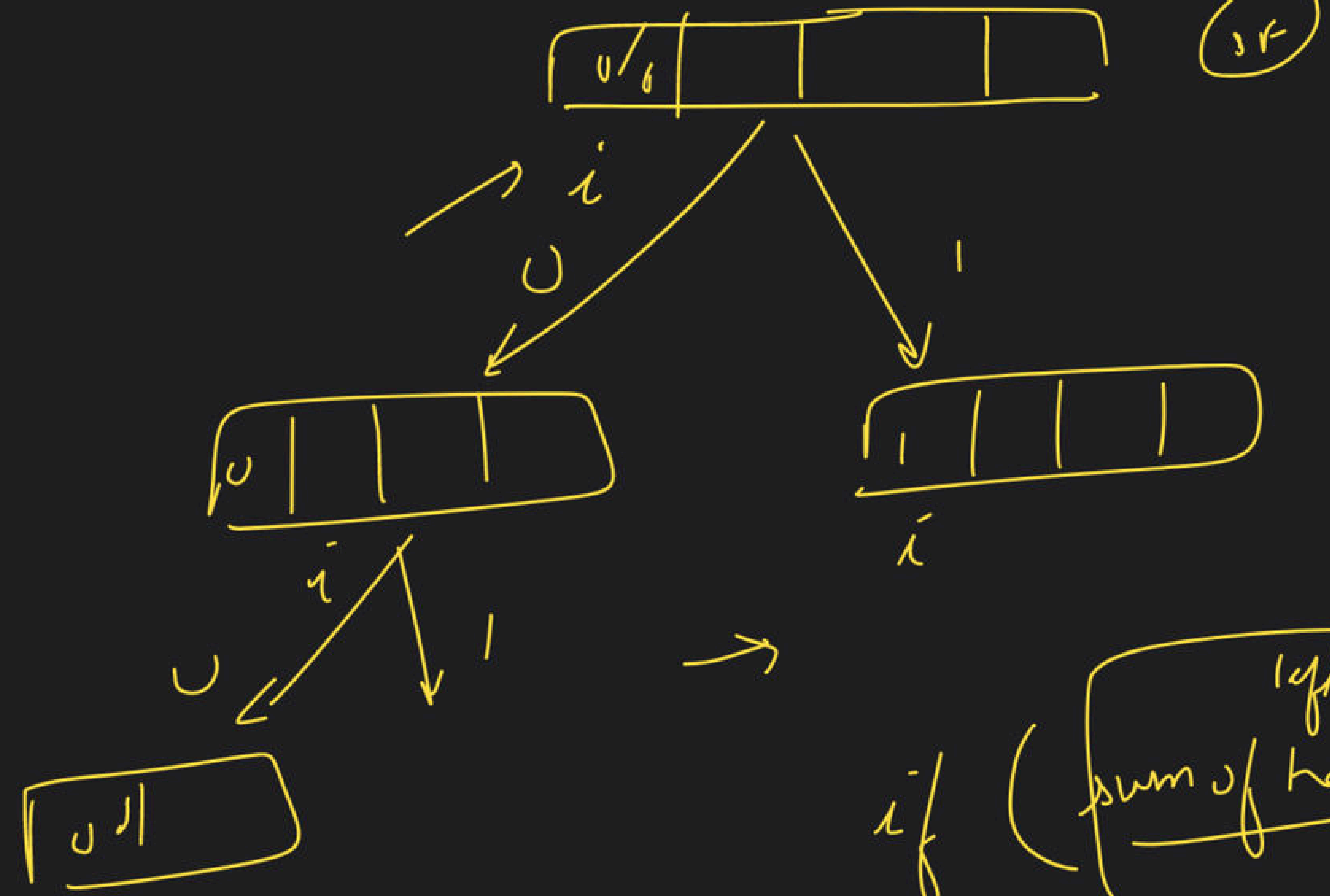

0 0
0 1
1 0
1 1
4

```
if( i > j )
{
if(leftsum = RightSum)
        cout output
return;
}
```


n = 1
2n → ? + 1
= 2
(0 , 0 , [|] , 0 , 1)
LS
RS
i
j
0
2n-1
0 , 0
0 , 1
1 , 0
1 , 1
(0 , 0 , [0|0] , 1 , 0)
==
print
i > j
(0 , 1 , [0|1] , 1 , 0)
= 2
i > j
(1 , 1 , [1|1] , 1 , 0)
==
print
i > j
(1 , 0 , [1|0] , 1 , 0)
==
i > j
n = 1
()
→))
output i , j
output [i] = '(' ;
[j] = ')'
2

n = 2

LOVEB



i > j

↳ f(n)

O(1)

sum

if (sum of half (left) = sum of right half)

O(1)

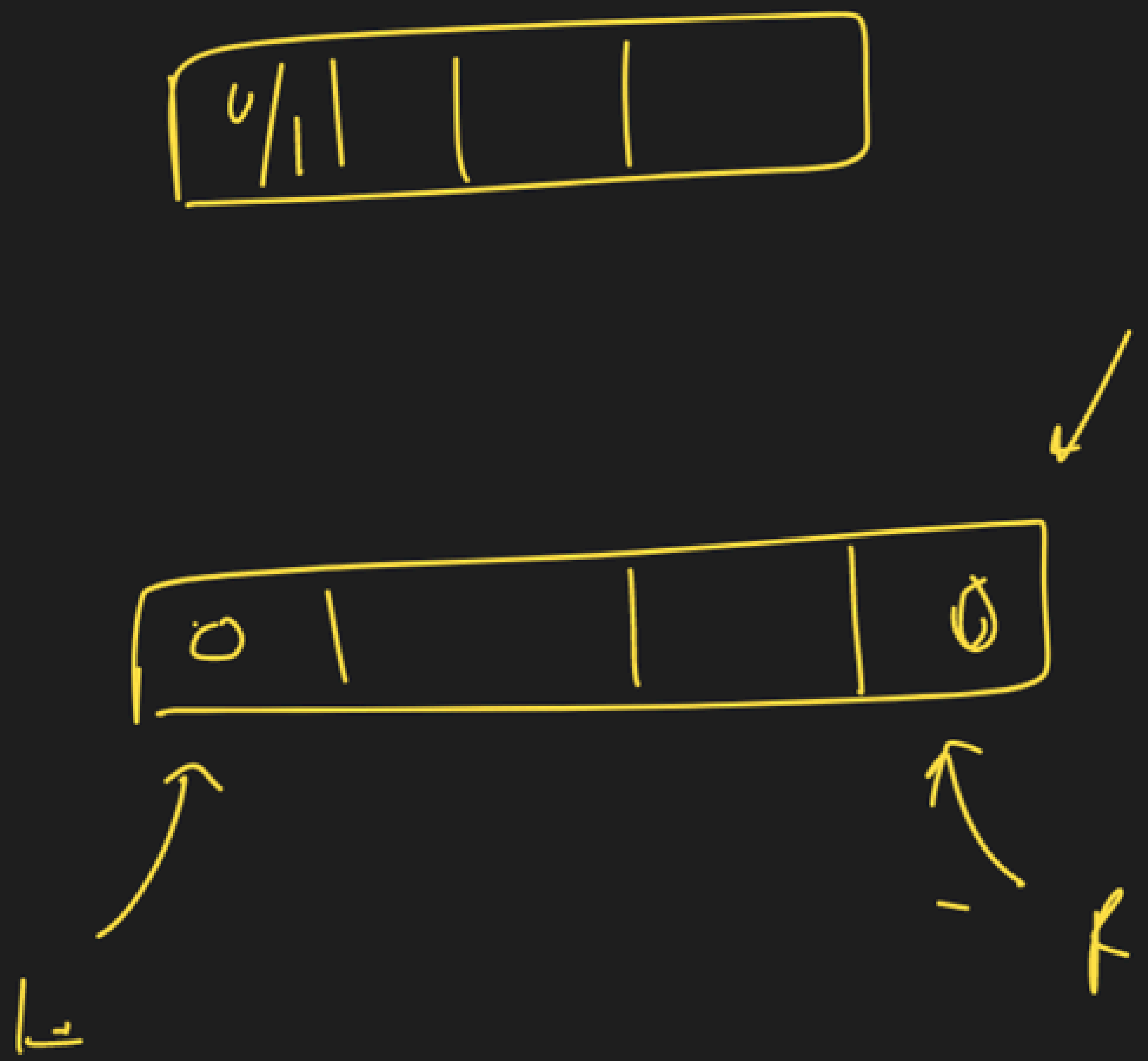
→ Course → OS, DBMS, OOPS, CN

⇓

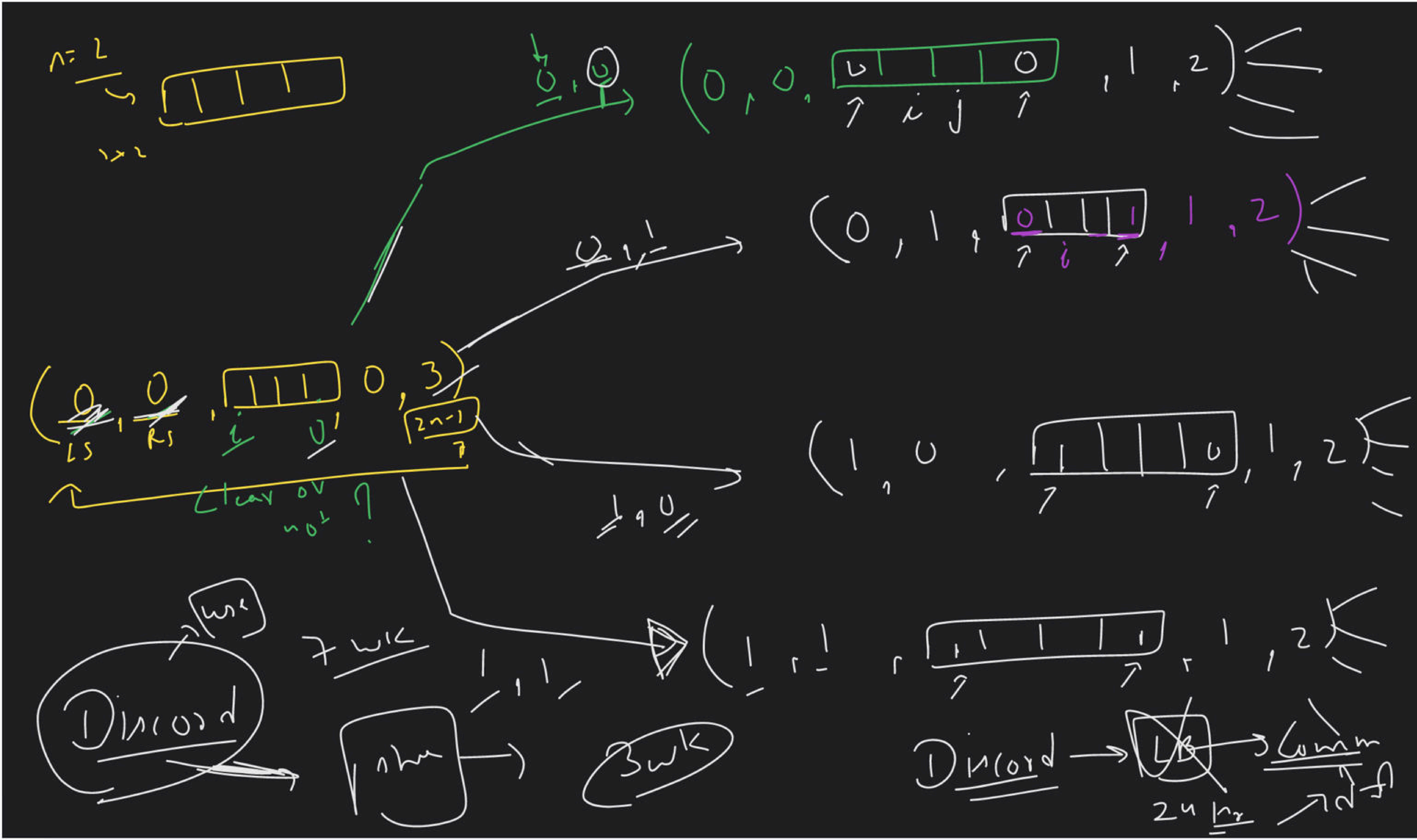
1 Buy

4 day →

30%. →


0/1
0
0
L
R

$n = 2$



n = 2
1×2
(0 , 0 , [| | |] 0 , 3)
LS
RS
i
j
2n-1
Clear or not ?
0 , 0
(0 , 0 , [0 | | | 0] , 1 , 2)
i j
0 , 1
(0 , 1 , [0 | | | 1] , 1 , 2)
i j
1 , 0
(1 , 0 , [1 | | | 0] , 1 , 2)
1 , 1
(1 , 1 , [1 | | | 1] , 1 , 2)
Discord
wk
7 wk
1 hr
3 wk
Discord → LB → Comm
24 hr

shed -2 2hr

NB

Reminder 2h

Pointer

T.Y

→ Rec h

DW

1.5hr

LL Tre DP hr

2hr+

→ incl. ex

→

DW

3.4