Mega Class: DnC/Backtracking

Special class

Love Babbar • July 21, 2024

① LC-22: Generate Parentheses

7 min ⇒ Read & Think

()

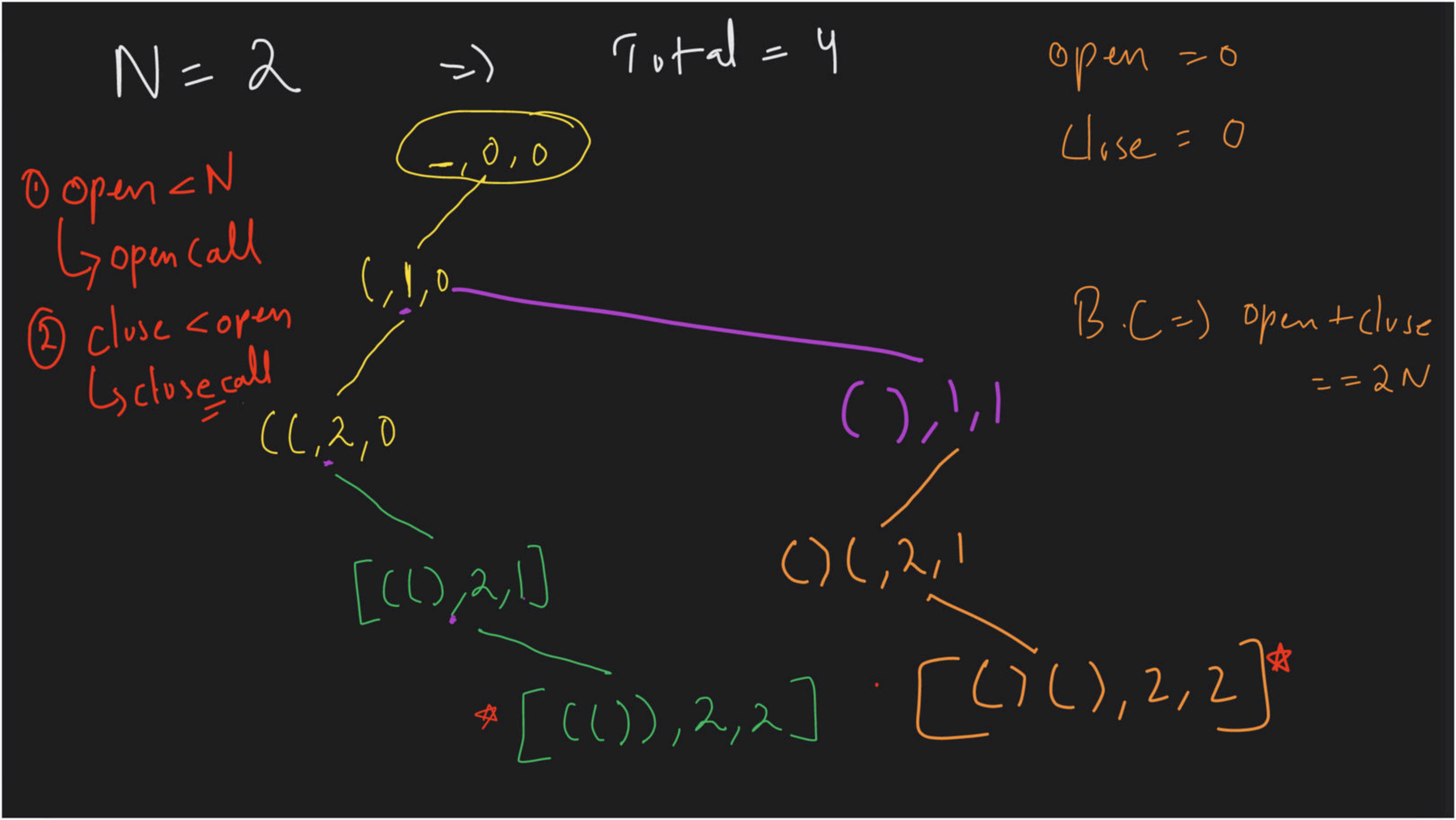
)(?

① Jay : valid → track ✓

② Yash : open '(' se start ✓

③ Mukesh : Track open & close count. ✓

⇒ Restricted RE


N = 2 => Total = 4
open = 0
close = 0
_,0,0
① open < N
↳ open call
② close < open
↳ close call
(,1,0
(),1,1
((,2,0
[(()),2,2]
()(,2,1
[()()),2,2]
B.C => open + close
== 2N

① if (o + c == 2n)
{
// store out string
return;
}

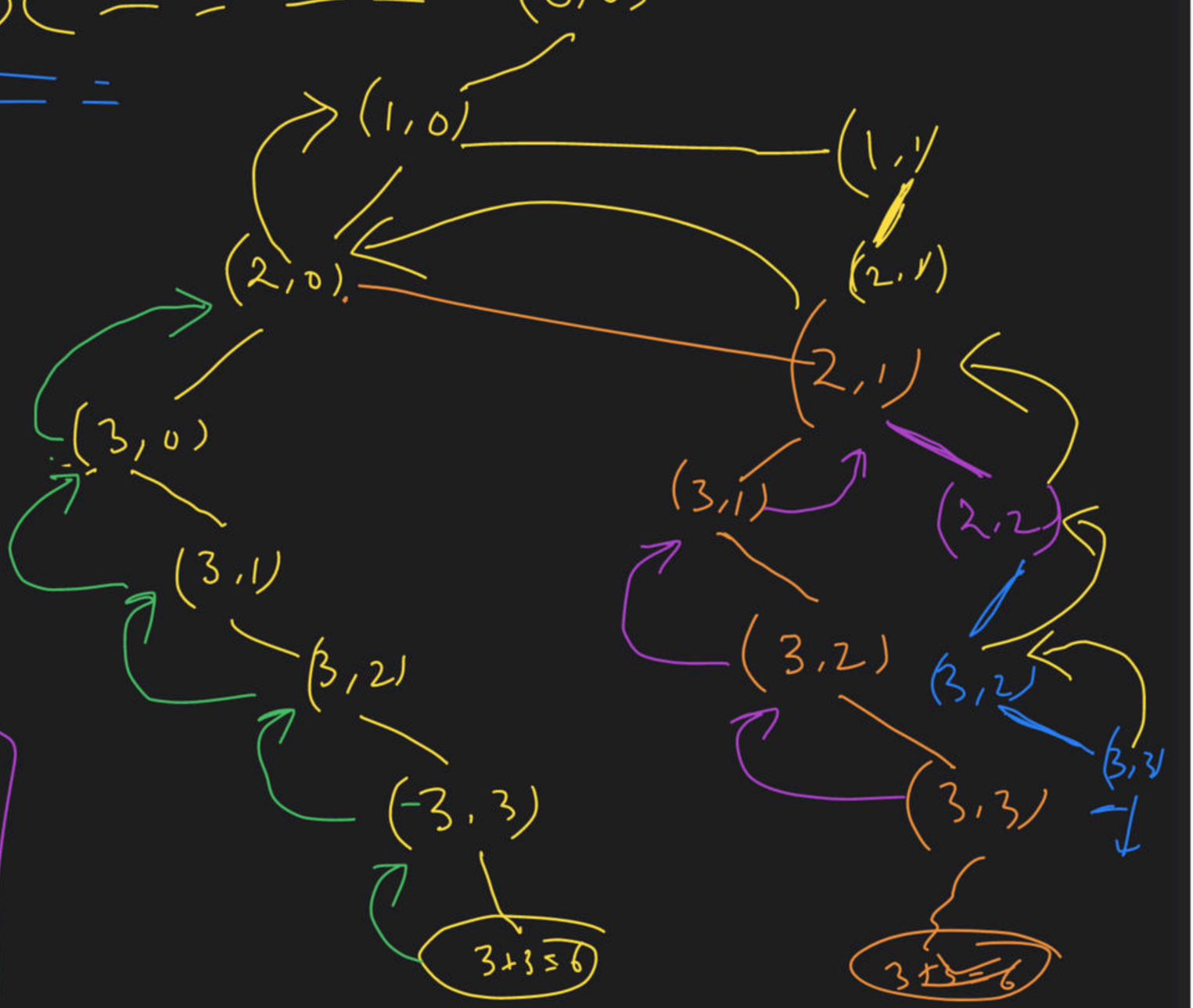
② if (o < n)
'(' → push
Rε(open+1, close)
pop

③ if (c < o)
')' → push
Rε(o, c+1)
pop

① ((( )))
② (( ) ( ))
③ (( )) ( )

out = ( ) (— — — — — —

N = 3



② LC240: Search a 2D Matrix - II

5 min : Read + Think

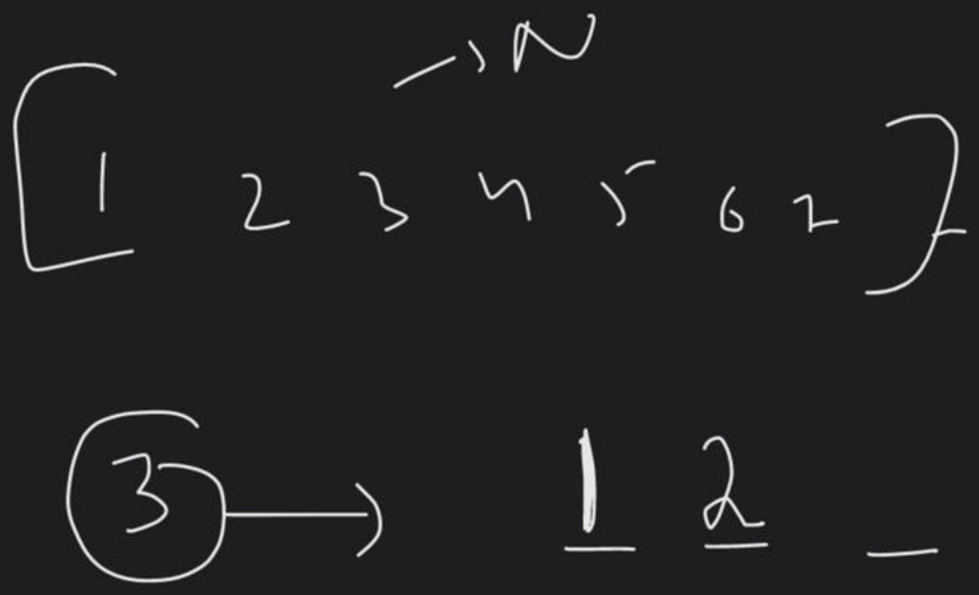
TRY
↳ B.S
iterative

↙
find movement to
Row or col →

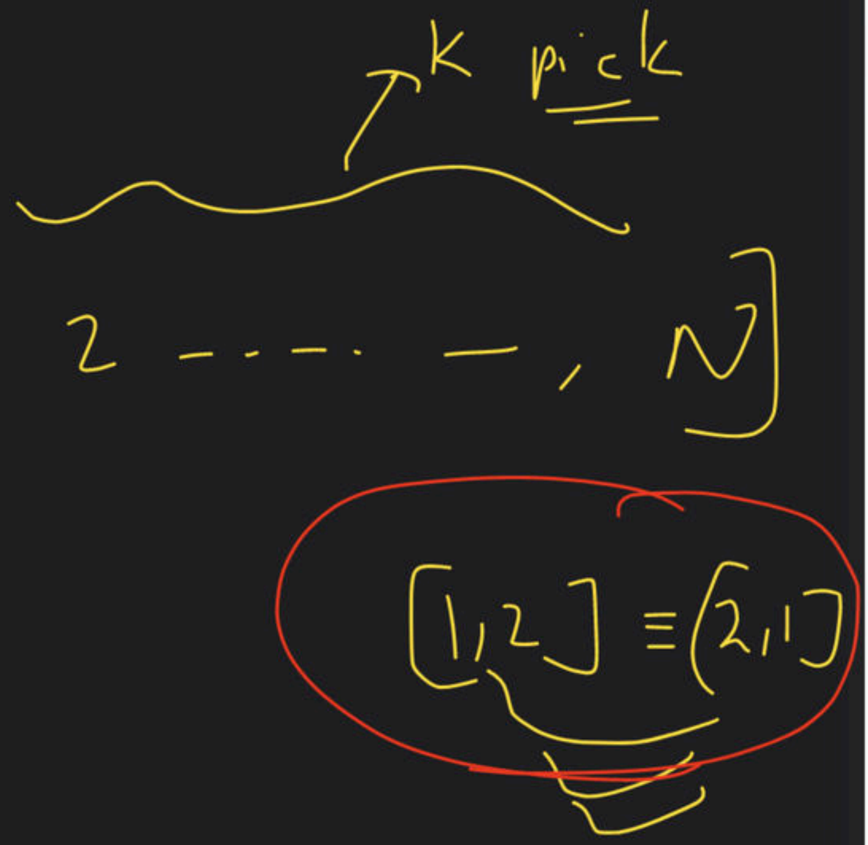
③ Combinations LC = 77.

5 min → Read & Think

=) N , K



3 → 1 2 _



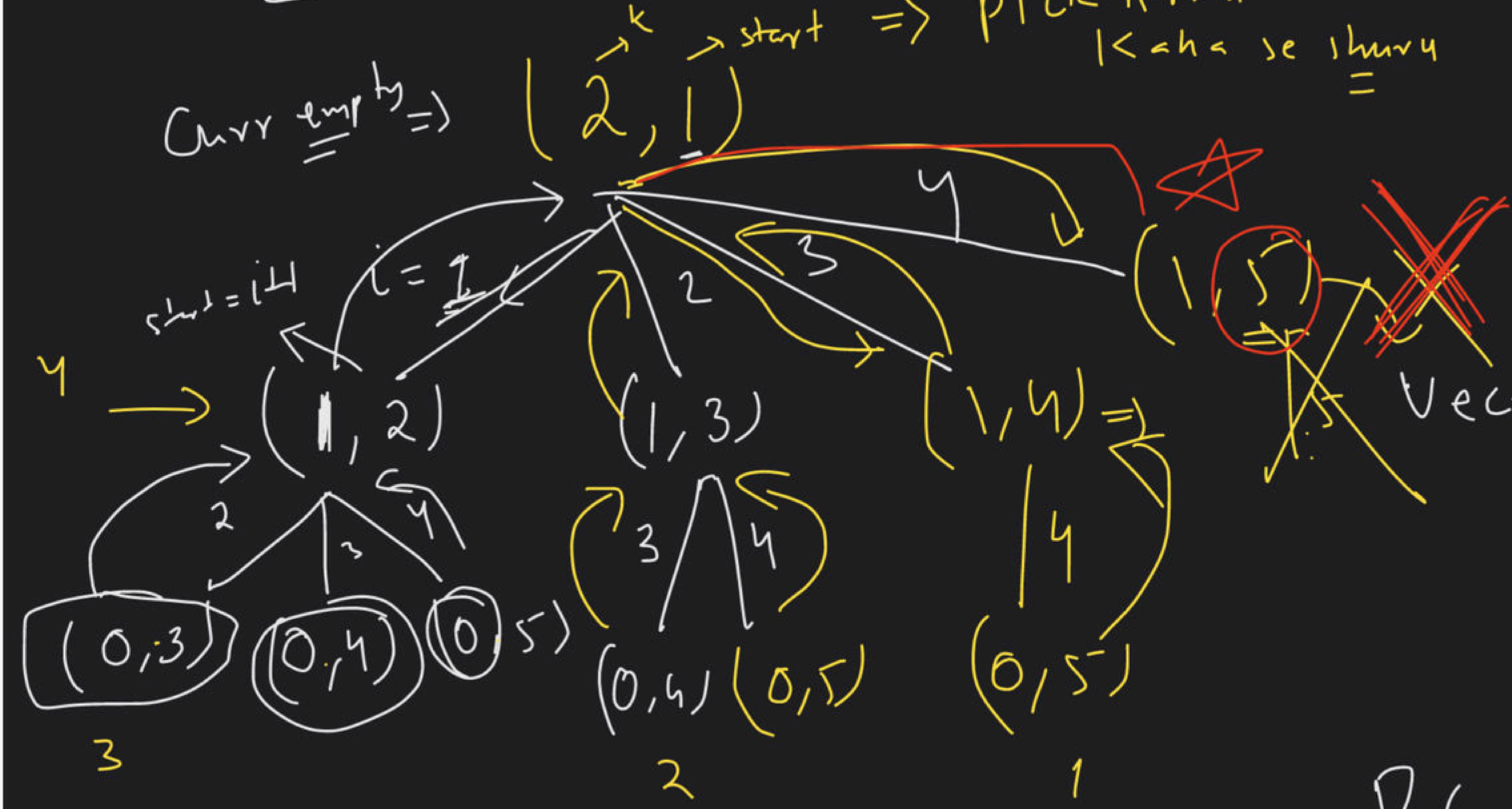
N = 4, K = 2

Curr → [ ]

[1, 2, 3, 4]

pick krna

kaha se shuru



(1, 2)
(1, 3)
(1, 4)
(2, 3)
(2, 4)
(3, 4)

Vector <int> Curr

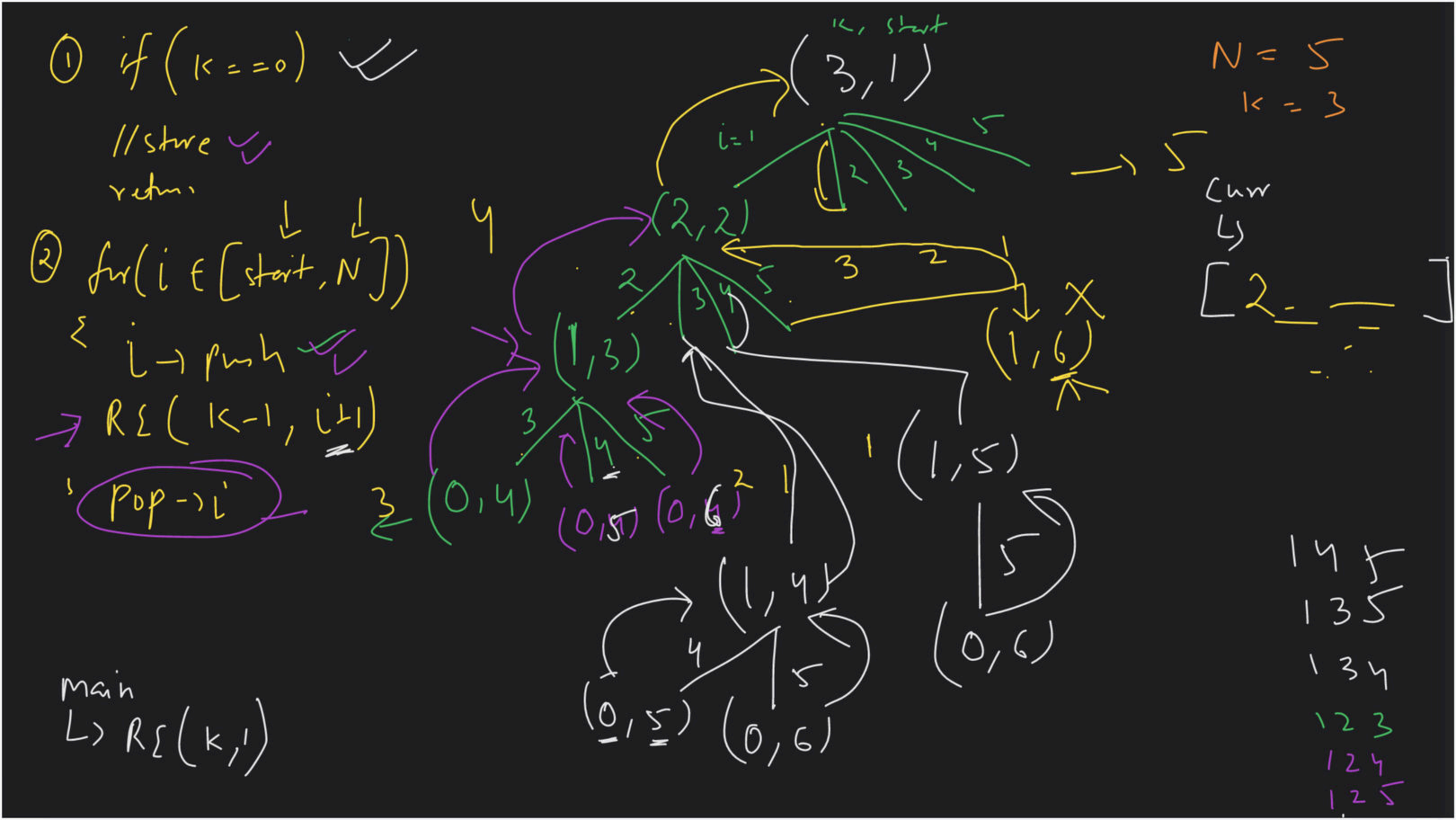
↓

I'll form combi. in it.

BC → if (k == 0)

✓ (ans.push(curr))

⇒ (start == N)


① if (k==0)
//store
return
② for(i ∈ [start, N])
{ i → push
R[(k-1, i+1)
Pop -> i
main
R[(k,1)
k, start
(3,1)
N = 5
k = 3
i=1
(2,2)
(1,3)
(1,6)
X
(0,4)
(0,5)
(0,6)
(1,5)
(1,4)
(0,5)
(0,6)
(0,6)
Curr
[2 _ _ _ _]
1 4 5
1 3 5
1 3 4
1 2 3
1 2 4
1 2 5

$N = 5$

$k = 3$

$\leftarrow - - -$

$N_{(k} \Rightarrow) \Rightarrow$

$N(k = \frac{N!}{k!\,(N-k)!}$

4 LC = 79, Word Search

⇒ 5 min

↳ Think & Read

A	B	C	E
S	F	C	S
A	D	E	E

Word = (ABCCED)

for (iRow) ≤ I

for (cols

{

{

har positio
se stmt

}

}

B.C

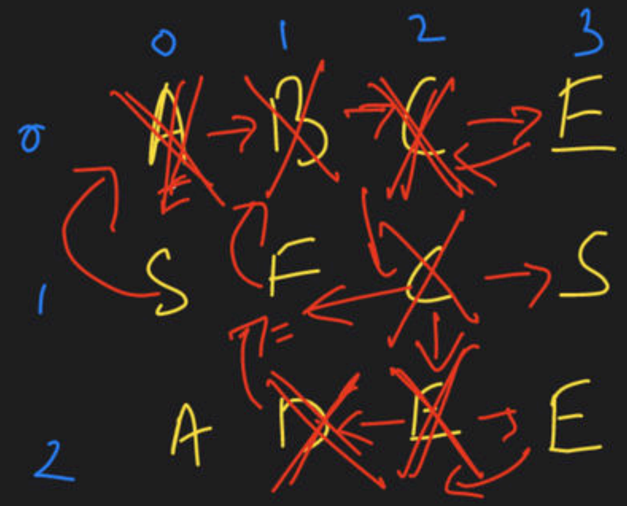
if(row<0 || row >=B.size()

|| col <0 || col >= B[0].size()

B[row][col] != word[i]

Visited → 2D matrix

↳ Board to visit



① if i == word.size()
return true;

② B.C =

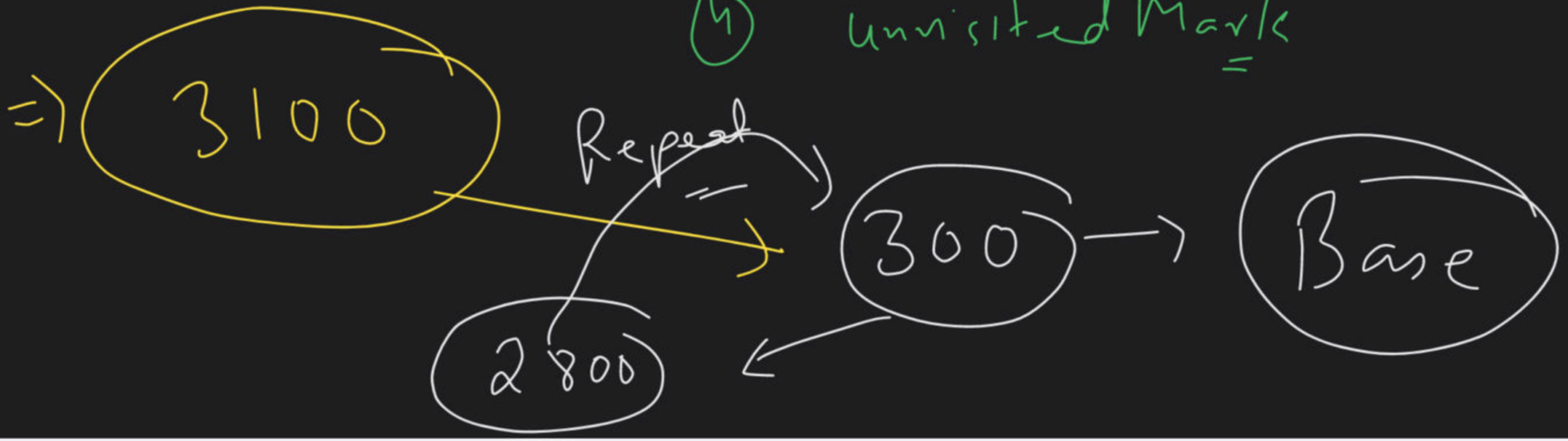
③ Visited Mark.

④ R.Σ(row+1, col, i+1) //Down

⑤ Unvisited Mark

word[i] == B[r][c]

A B C C E D
i i i i i



(5) N-Queens - II

- Khud krna hai
- Hint → same N-Queens-I solution as it is to count.

⑥ LC ⇒ 473 : Matchsticks to Square

10 Min Break

Back at ☆☆ 04:52 PM

5 → org. Break.

5 → Think & Read

$\Rightarrow$ [1, 1, 2, 2, 2]

Sides = 4

All sides are of equal length

I can make square if I can divide My array into 4 equal subsets.

① array.length < 4 ⇒ [2,2,2]

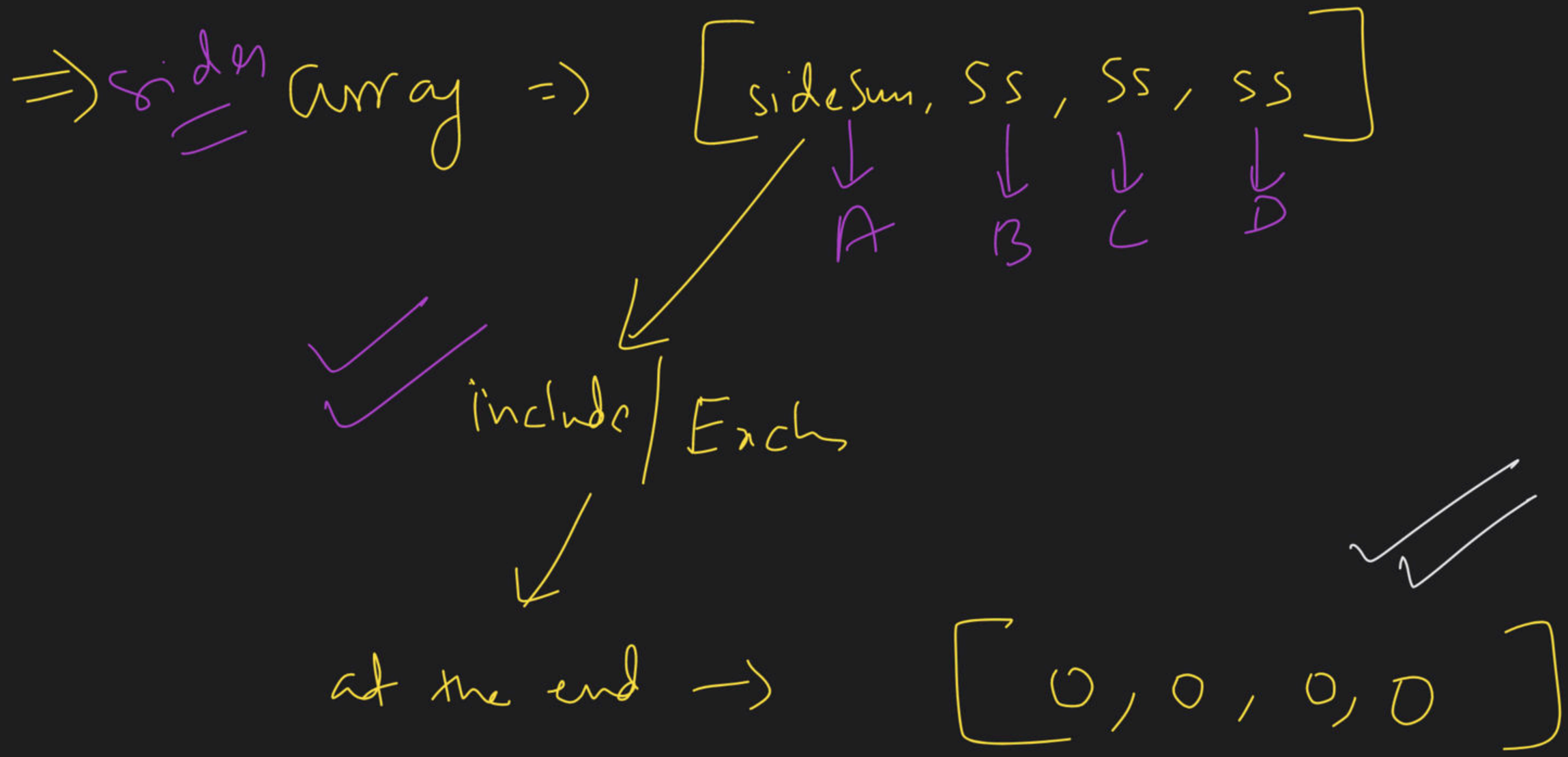
Return false.

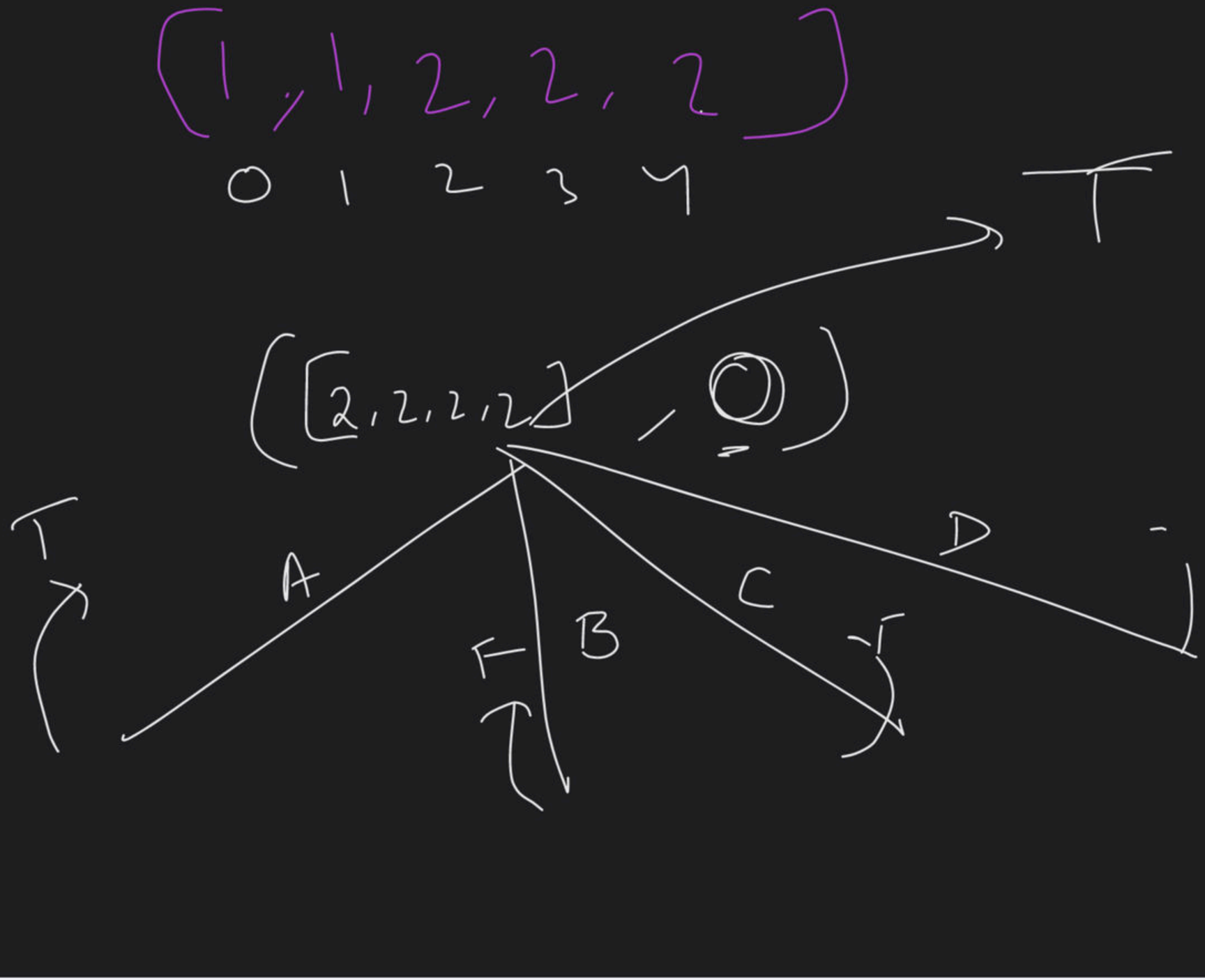
② (Sum % 4 != 0) => Sum not divisible by 4

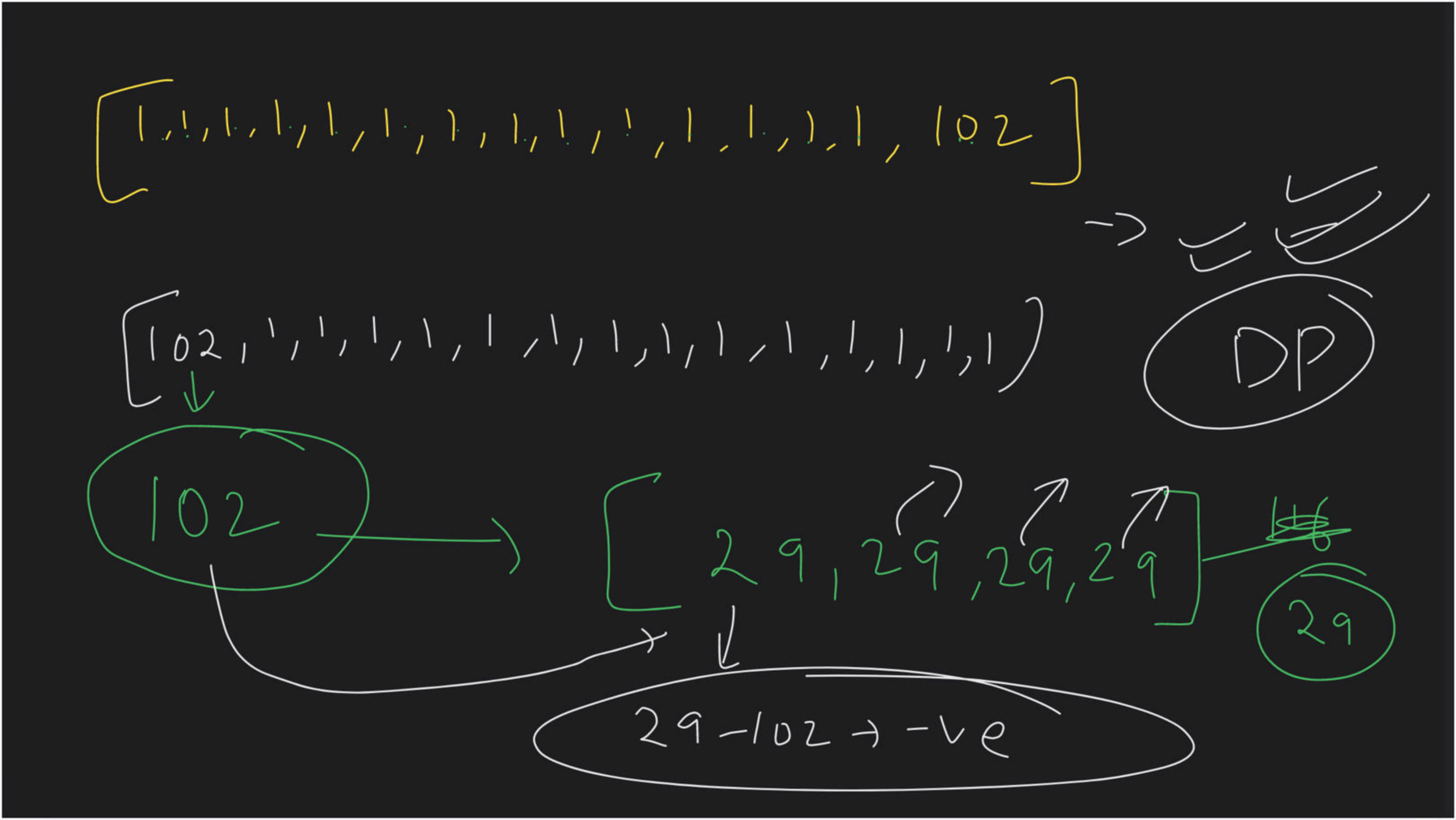
Return false

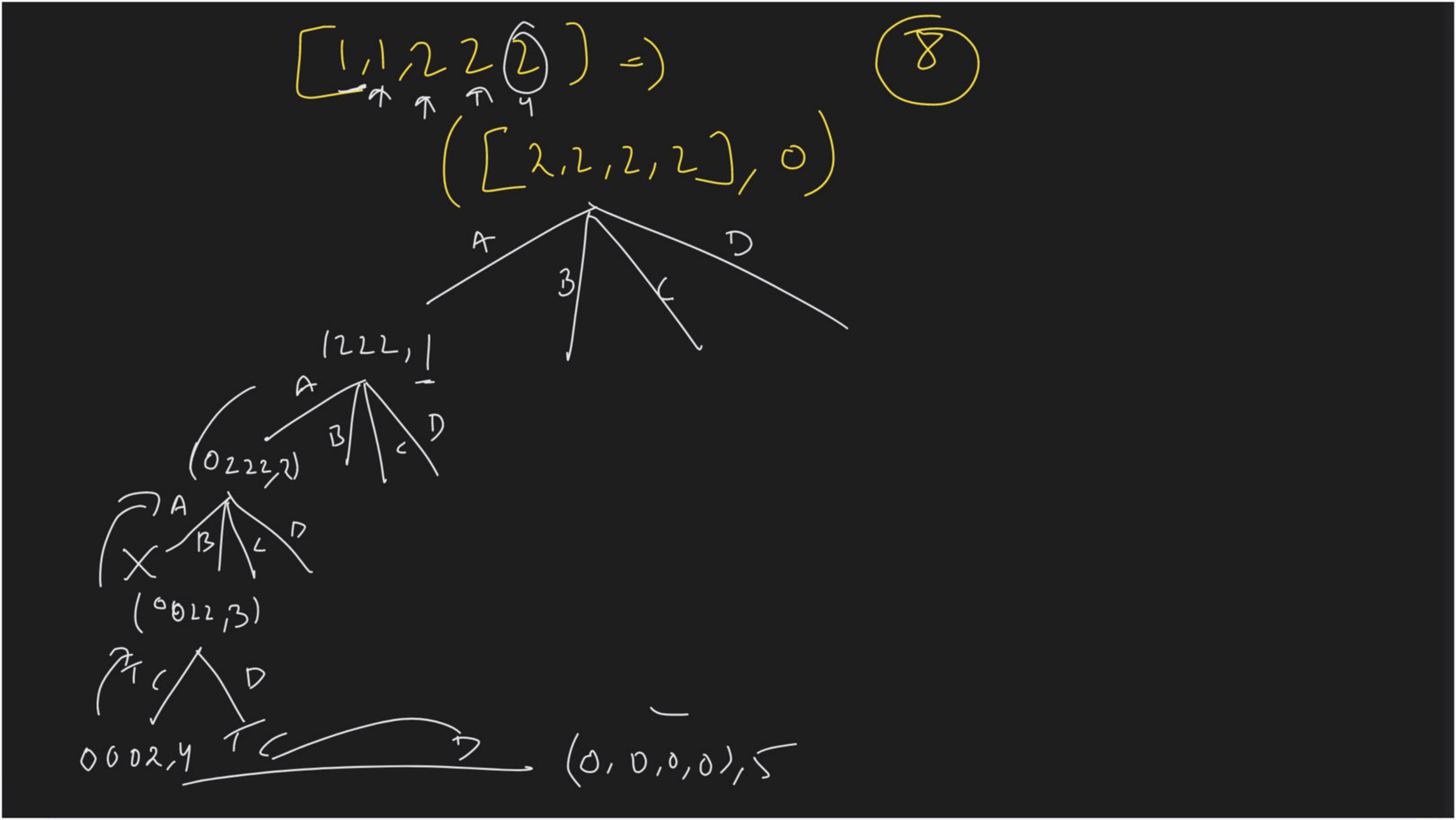
③ SideSum = Sum / 4;

Try to divide array into 4 subsets with sum equal to SideSum


=> sides array => [sideSum, ss, ss, ss]
A B C D
include / Each
at the end -> [0, 0, 0, 0]


[1,1,2,2,2]
0 1 2 3 4
T
([2,2,2,2], 0)
T
A
B
C
D


[1,1,1,1,1,1,1,1,1,1,1,1,1,1,102]
[102,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]
DP
102
[29,29,29,29]
29
29-102 → -ve


[1,1,2,2,2] =>
8
([2,2,2,2], 0)
A
B
C
D
(222,1)
A
B
C
D
(022,2)
A
B
C
D
X
(022,3)
C
D
(002,4
C
D
(0,0,0,0),5