n → open (
close)

n = 3 (×3)×3

= ( ) ( ) ( ) ✓

(( )) ( )

( ) (( )) ✓

((( ))) ✓

(( ) ( )) ✓

X

((( ))) (

((a+b) + c) + (d+c)

⇓

(( )) ( ) ✓

((a+b) + c) + (d+c)

① last opened parantheses closes first

② no. of opened parantheses is never less than

no. of closed paranthesis

open >= closed

at the end of string open = closed

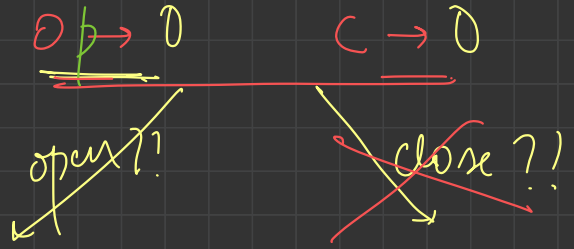
n = 3

open = 3

close = 3

((())) → all allowed orderings

i = 0

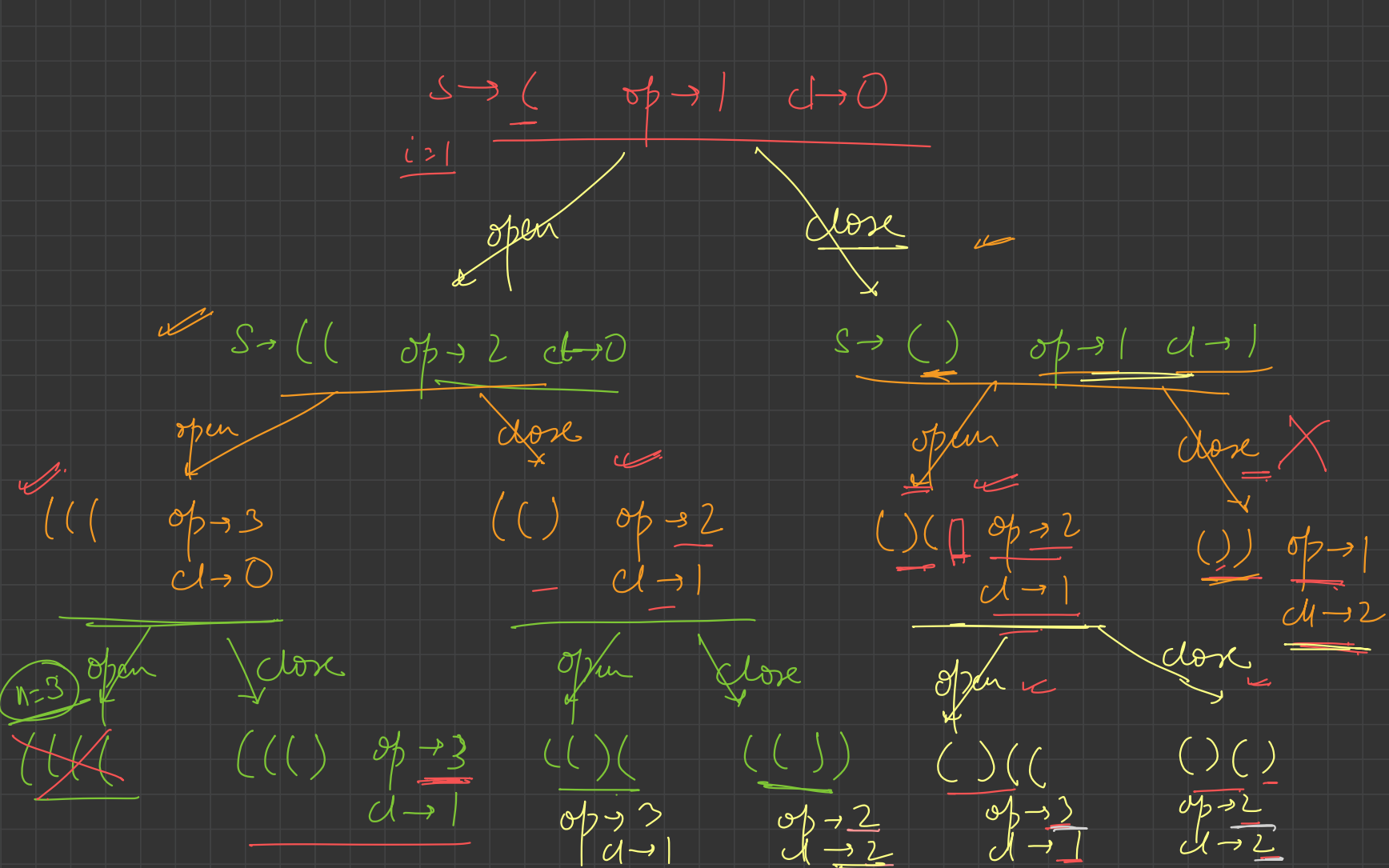


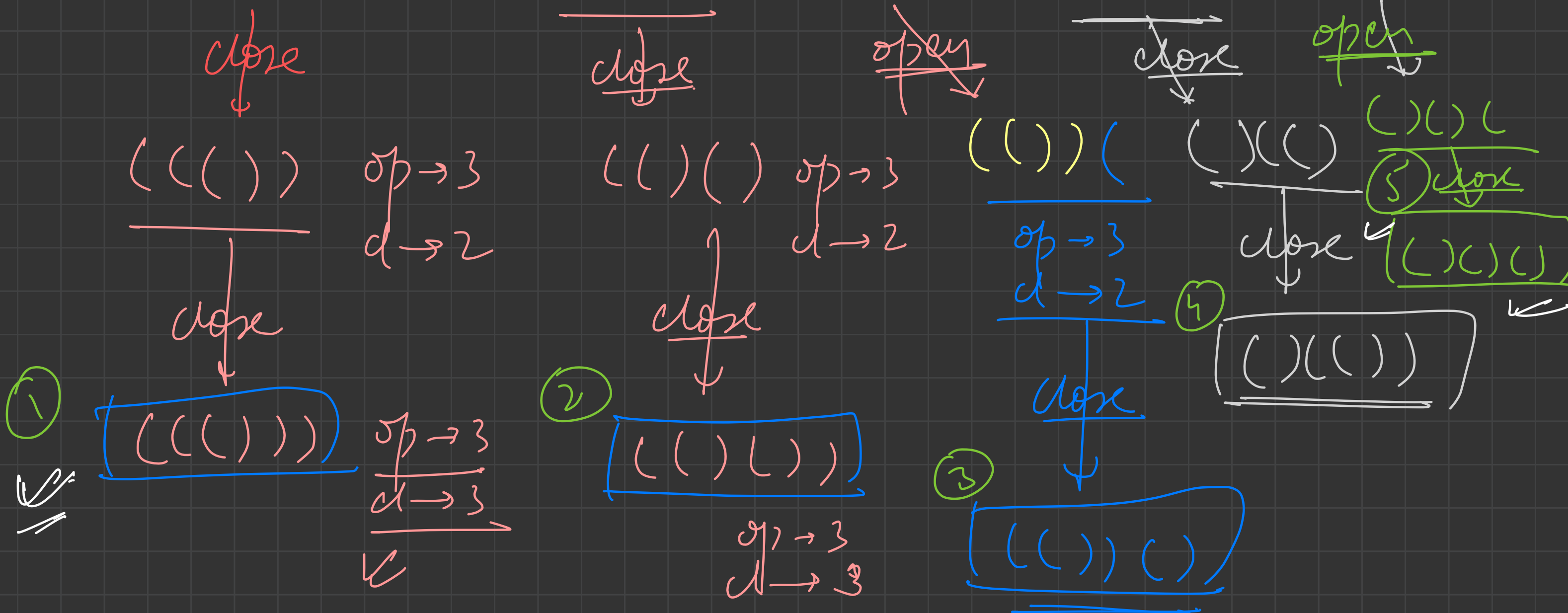
op → 1 c → 0

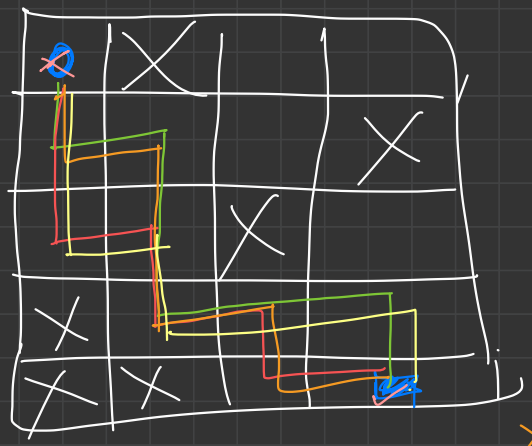
s → (

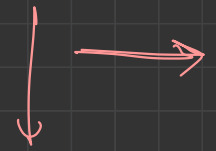
op → 0 c → 1

s →)


S→ (
op→1
cl→0
i=1
open
close
S→ ((
op→2
cl→0
S→ ()
op→1
cl→1
open
close
X
(((
op→3
cl→0
(()
op→2
cl→1
()(
op→2
cl→1
())
op→1
cl→2
open
close
n=3
((((
((()
op→3
cl→1
(()(
op→3
cl→1
(())
op→2
cl→2
()((
op→3
cl→1
()()
op→2
cl→2


close
op → 3
cl → 2
close
op → 3
cl → 3
close
op → 3
cl → 2
close
op → 3
cl → 3
open
op → 3
cl → 2
close
close
open
close
close
1
2
3
4
5





X not exist

-1

4 paths

④ D D R D R R D

③ D D R D R D R

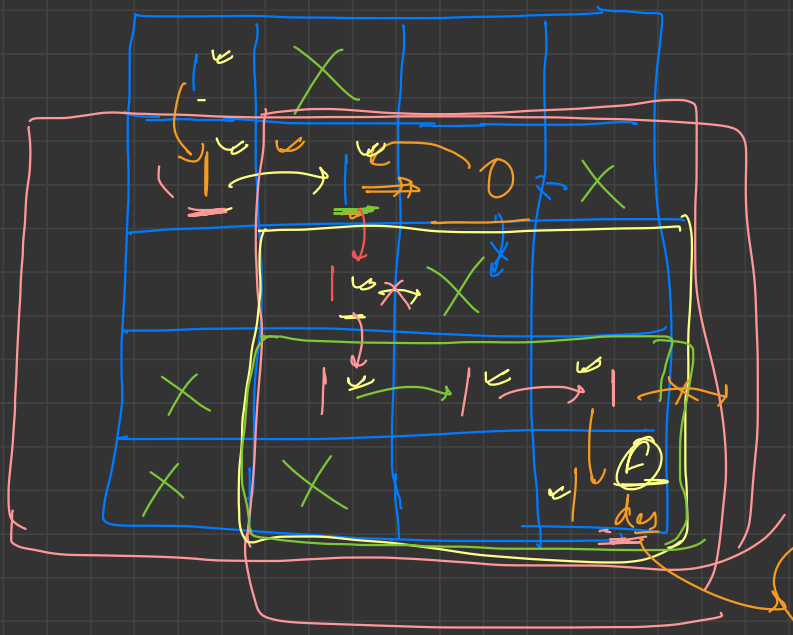
② D R D D R R D

① D R D D R D R

no DD

no RR

right most



① go right

② go down

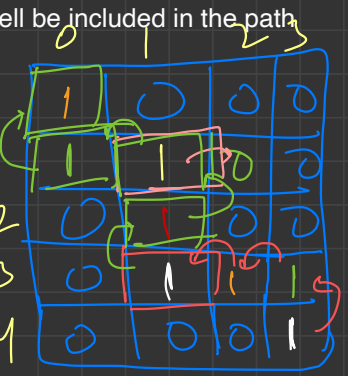
sub-grid

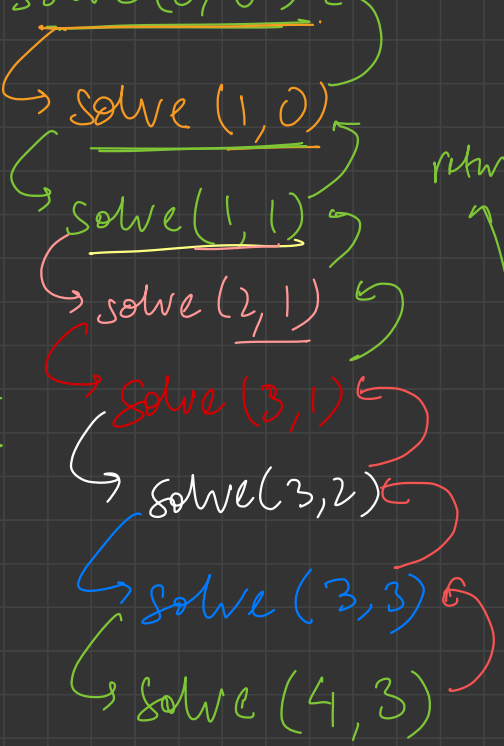
new - problem

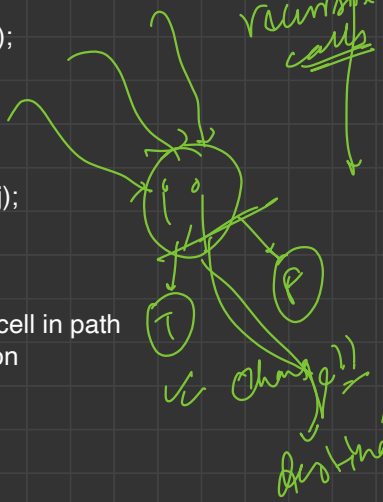
-1

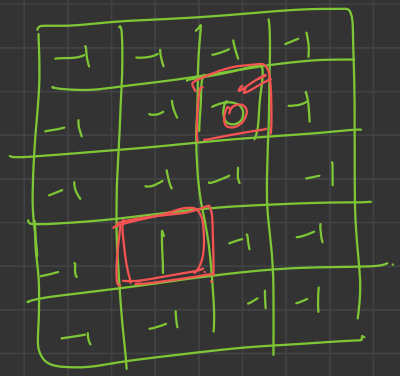
(n-1, m-1)

```
bool solve(int i, int j){
    output[i][j] = true;//this particular cell be included in the path
    //base case
    if(i == n - 1 and j == m - 1){
        //destination cell
        for (int i = 0; i < n; ++i){
            for (int j = 0; j < m; ++j){
                cout<<output[i][j]<<' ';
            }
            cout<<endl;
        }
        return true;
    }
    //go right
    if(j + 1 < m and maze[i][j+1] != 'X'){
        bool pathFoundRight = solve(i, j+1);
        if(pathFoundRight) return true;
    }
    //go down
    if(i + 1 < n and maze[i+1][j] != 'X'){
        bool pathFoundDown = solve(i+1, j);
        if(pathFoundDown) return true;
    }
    //no path found
    output[i][j] = false;//do not include this cell in path
    //as it cannot lead you to the destination
    return false;
}
```









-1 → not yet visited

visited { 0 → no path to dest
1 → path to dest