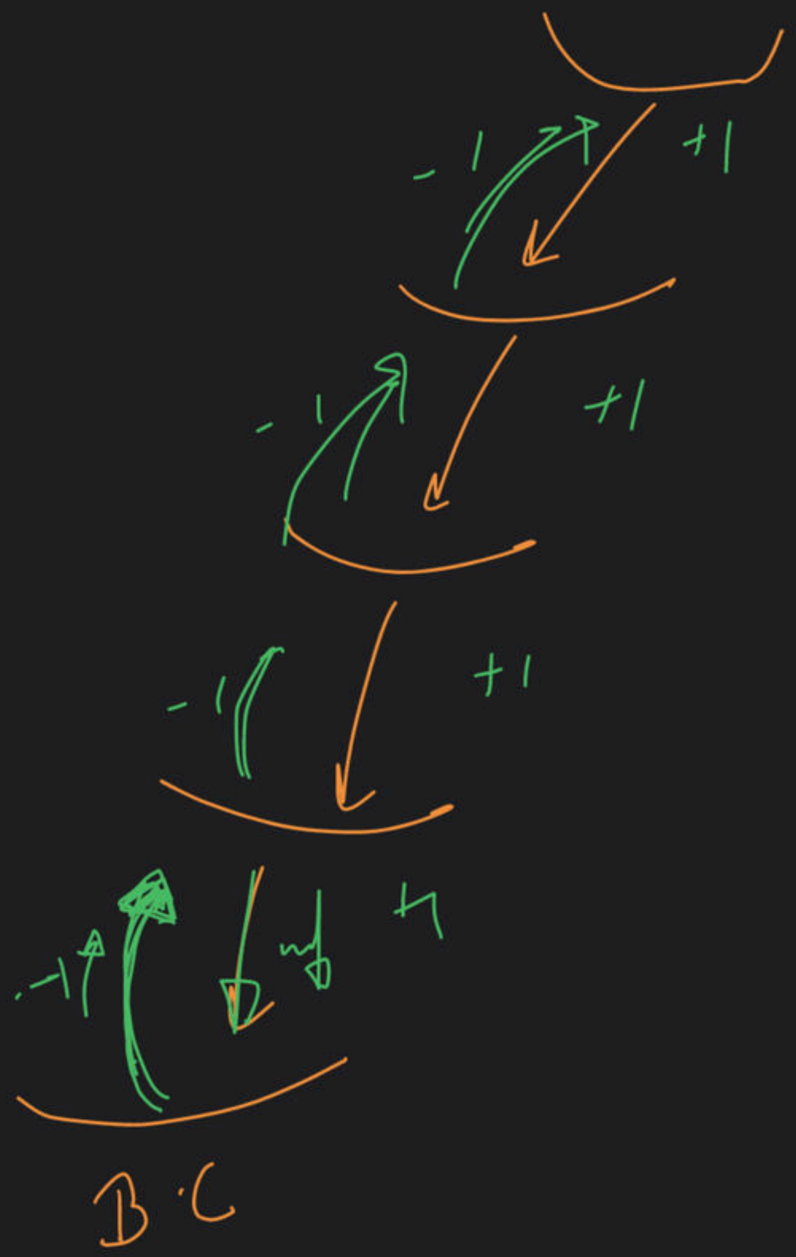
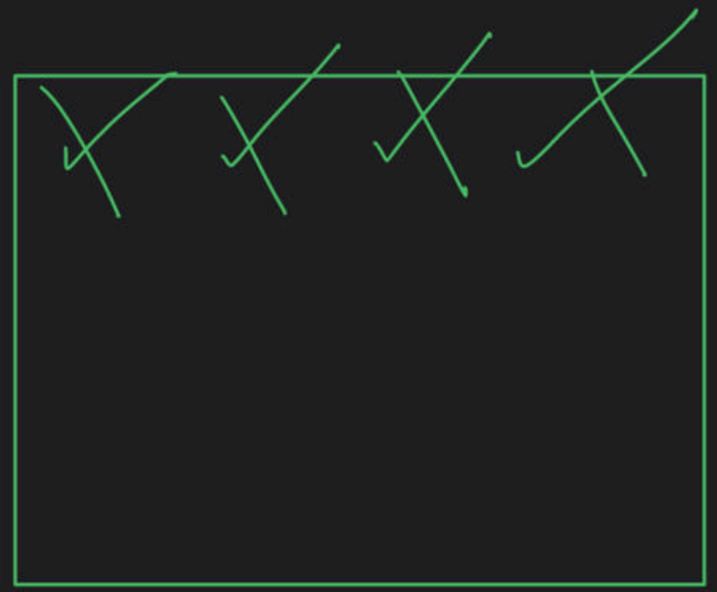
DnC && Backtracking Class - 2

Special class

Love Babbar • Oct 20, 2023


-1
+1
-1
+1
-1
+1
B.C

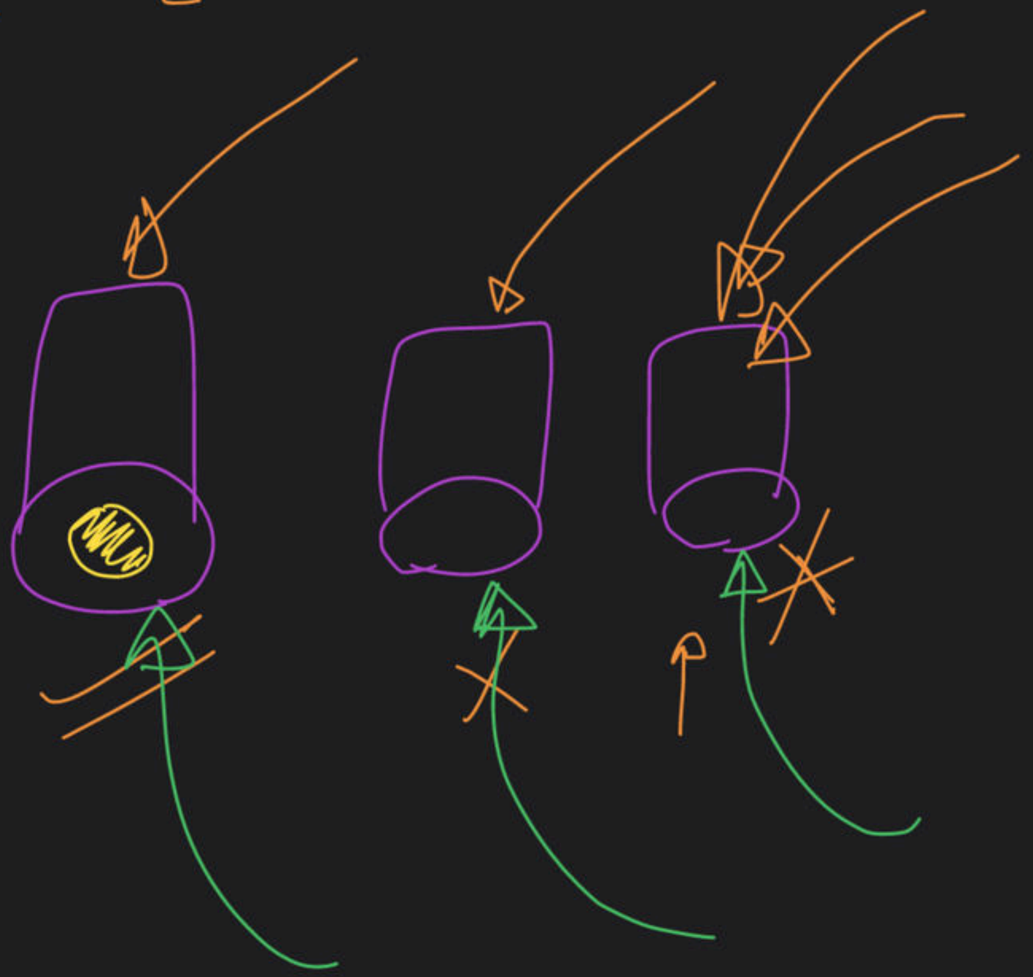


Backtracking :-

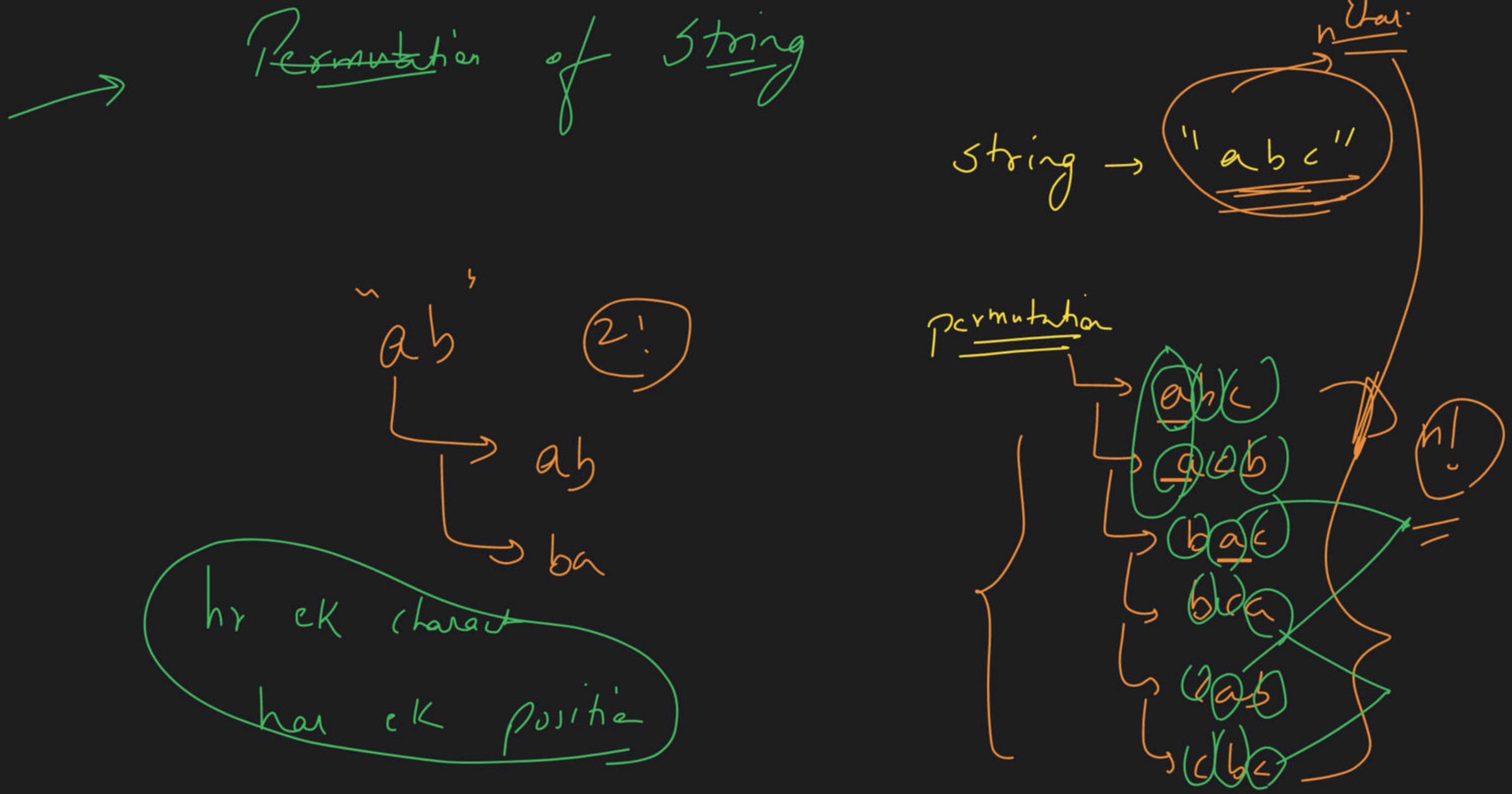
explore all possible sol^n

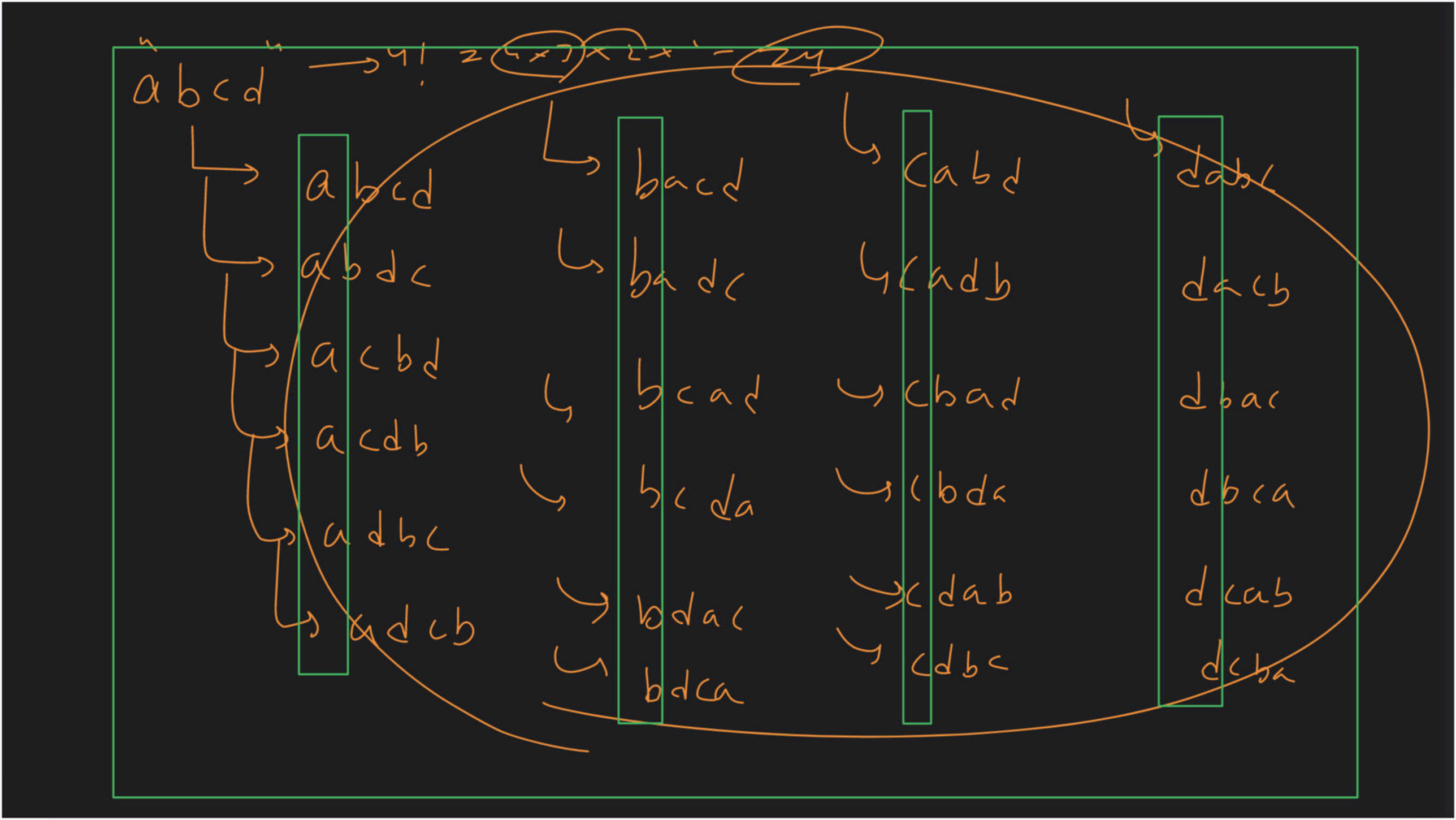
explore case

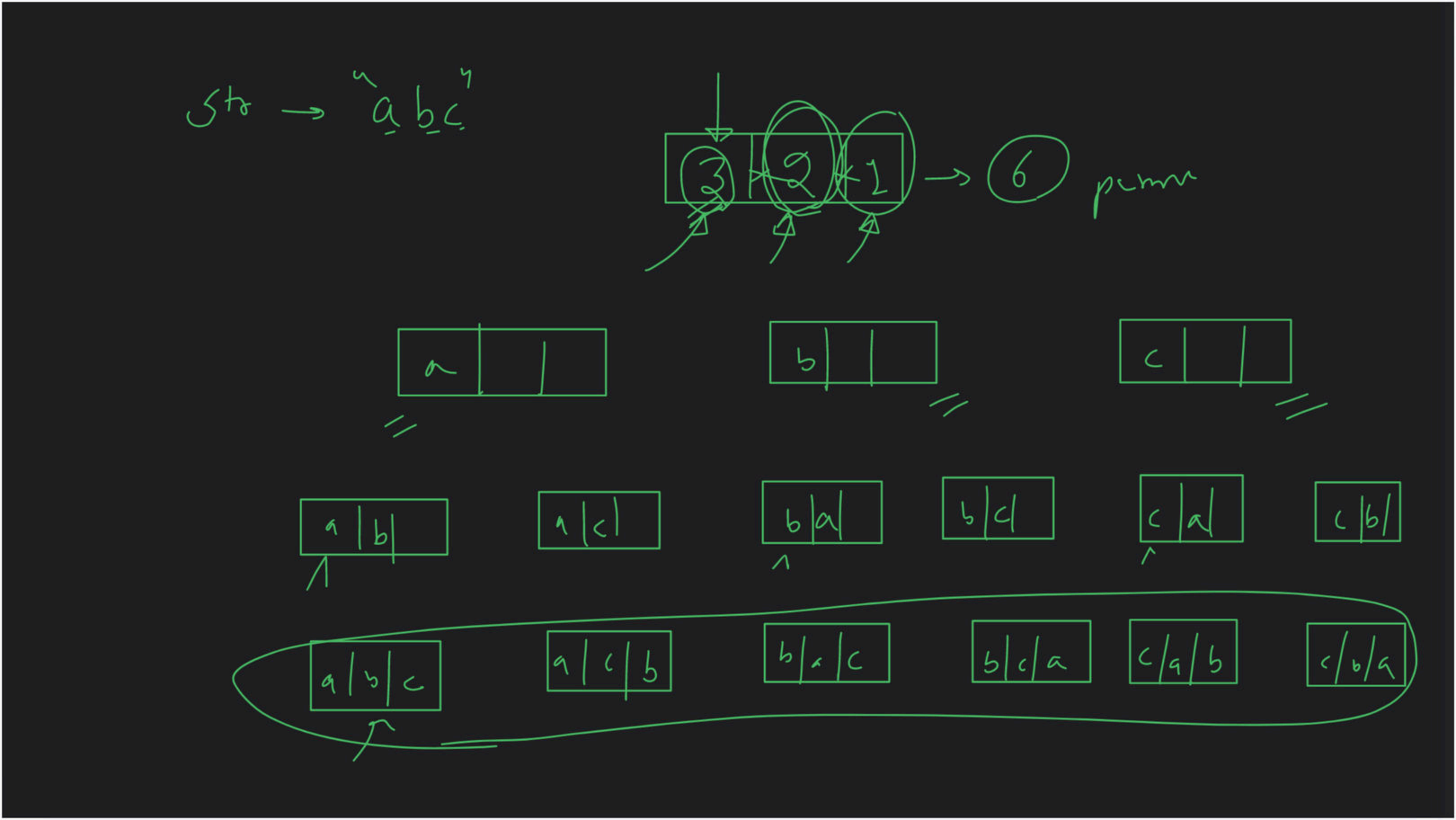
dubara explore X

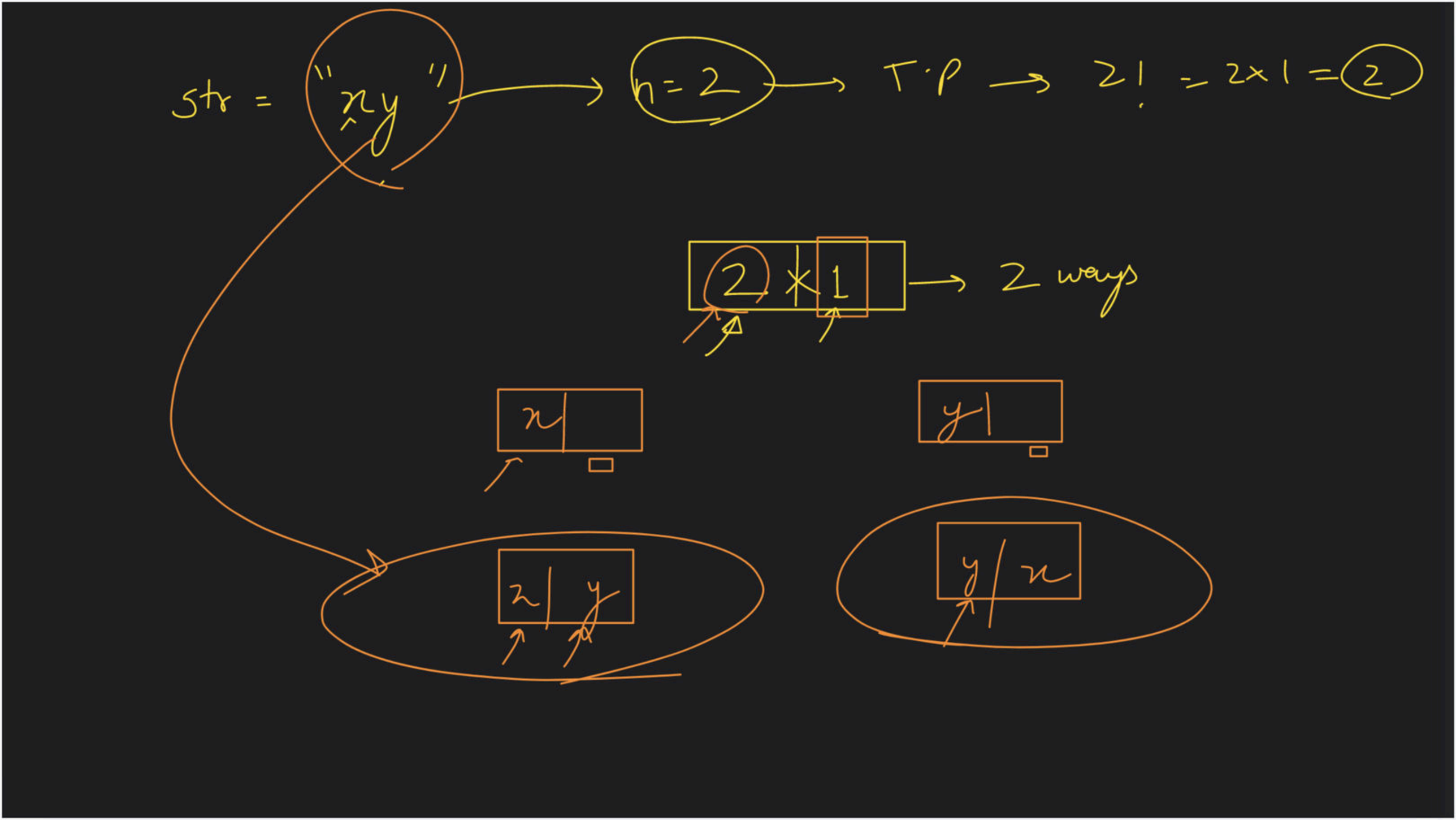


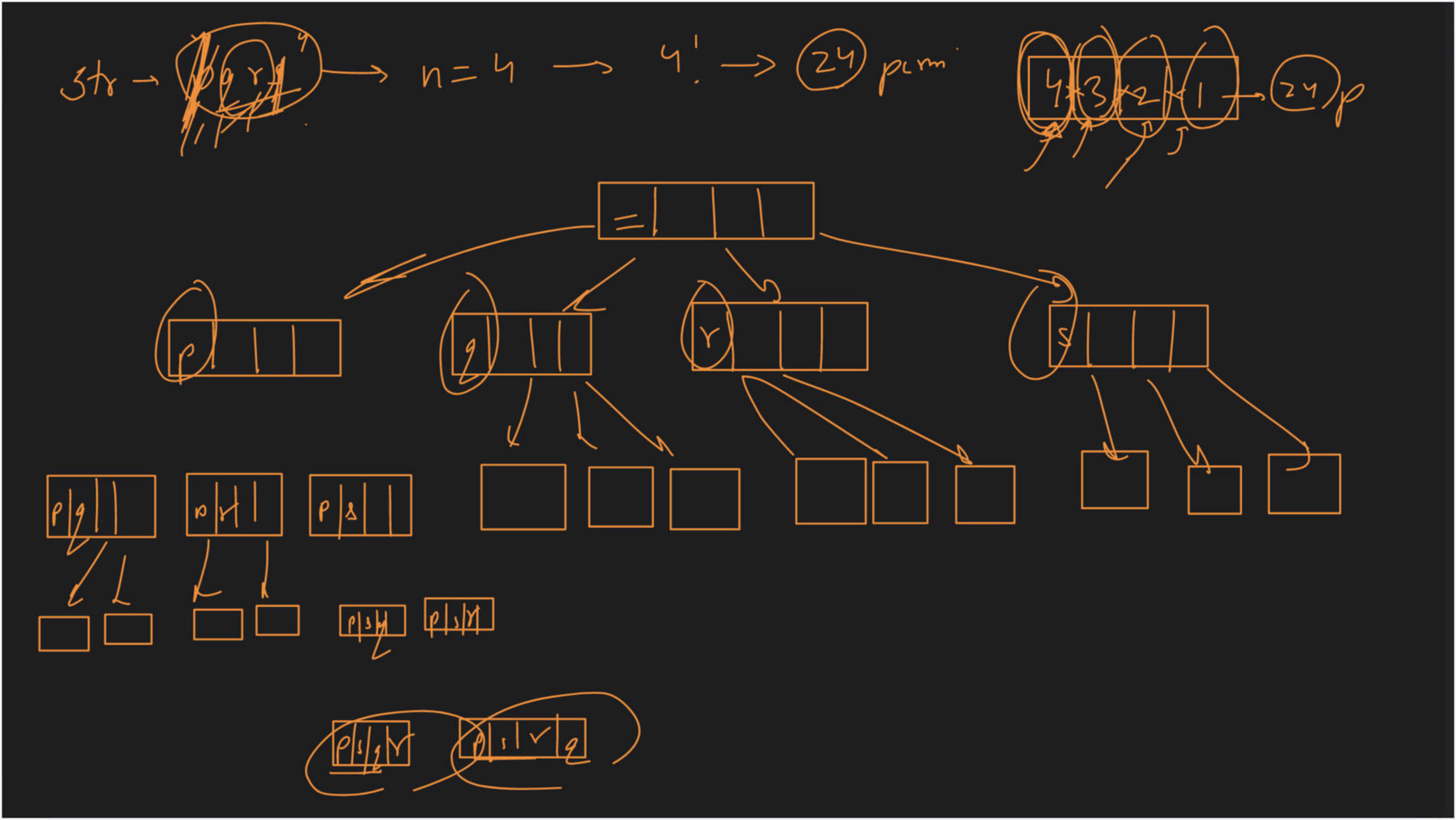
Permutation of String



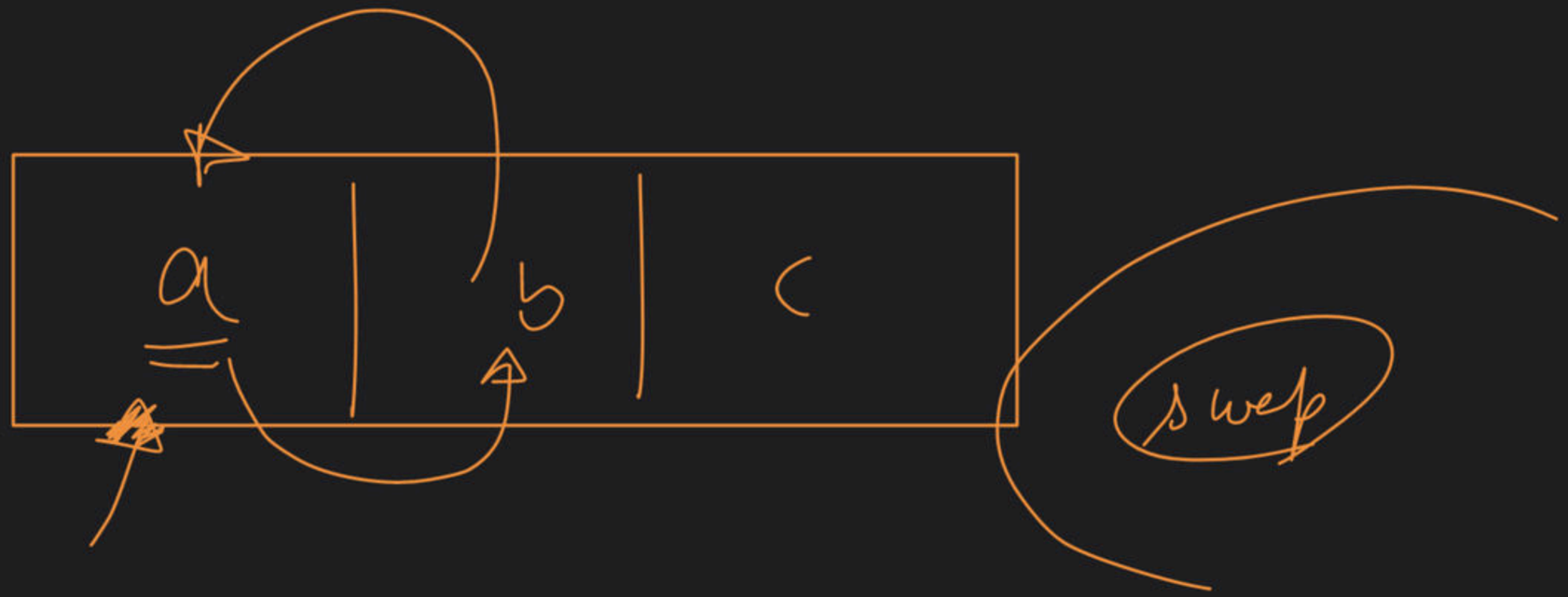

abcd
abcd
abdc
acbd
acdb
adbc
adcb
bacd
badc
bcad
bcda
bdac
bdca
cabd
cadb
cbad
cbda
cdab
cdba
dabc
dacb
dbac
dbca
dcab
dcba


Str → "a b c"
3 2 1
6
perm
a
b
c
a b
a c
b a
b c
c a
c b
a b c
a c b
b a c
b c a
c a b
c b a

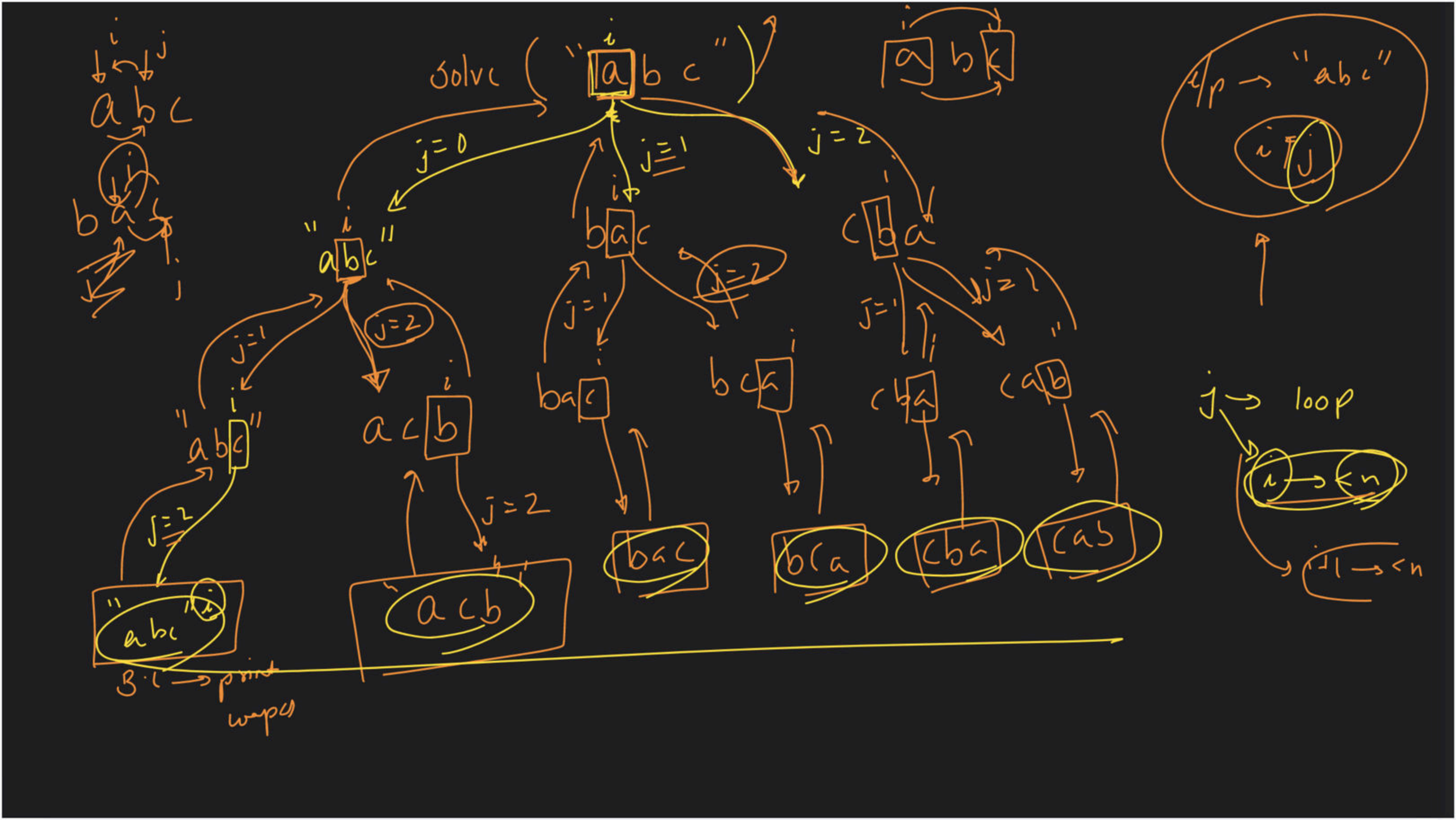

str = "xy"
n = 2
T.P → 2! = 2 x 1 = 2
2 x 1
→ 2 ways
x
y
z y
y x



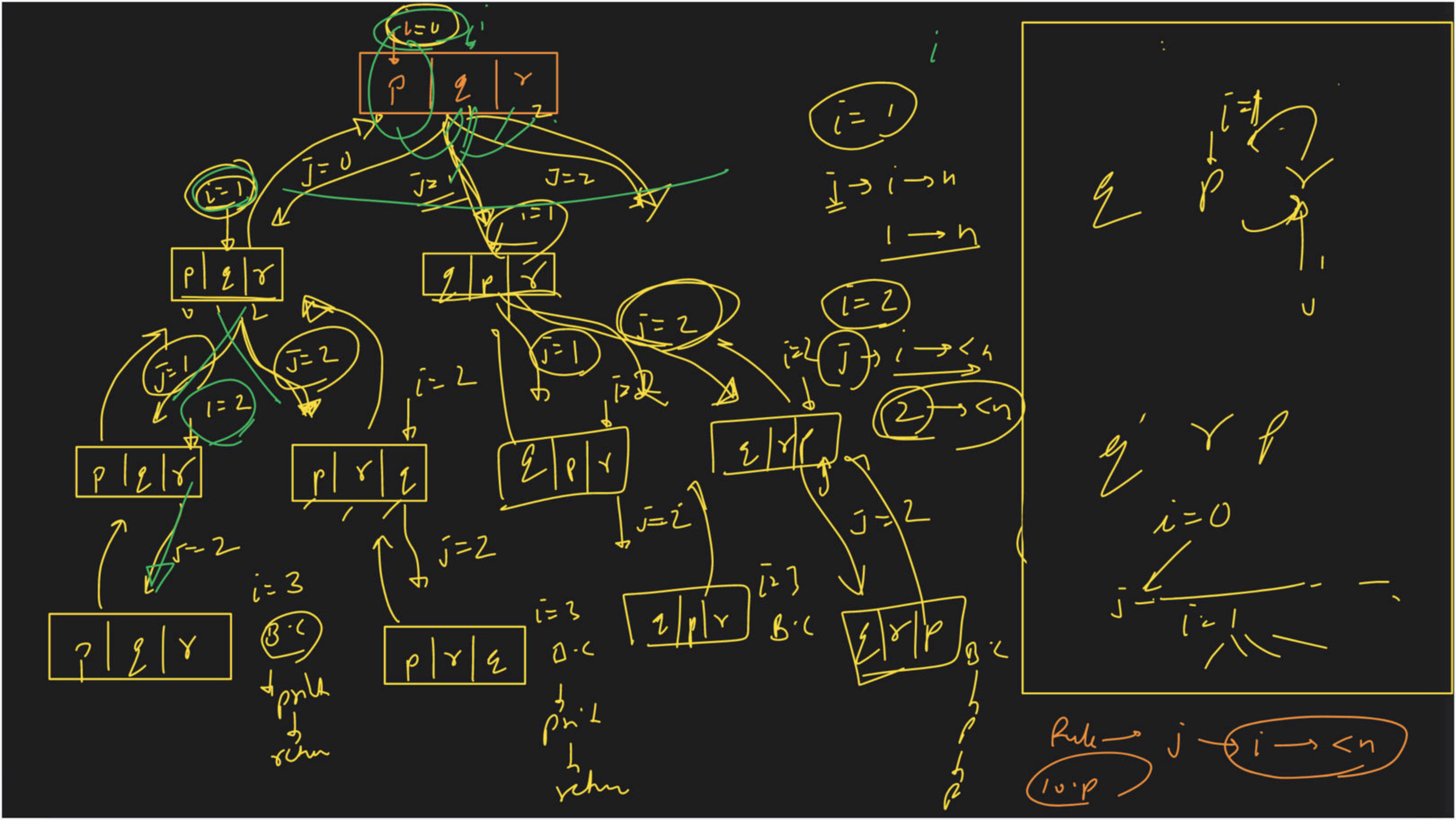
str → p q r s → n = 4 → 4! → 24 perm
4 × 3 × 2 × 1 → 24 p
=
p
q
r
s
p q
p r
p s
p s q
p s r
p s q r
p s r q


a
b
c
swap

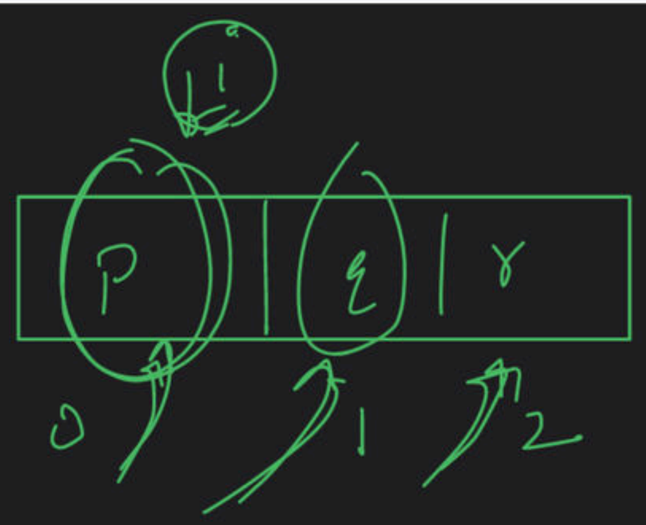
b
a
c

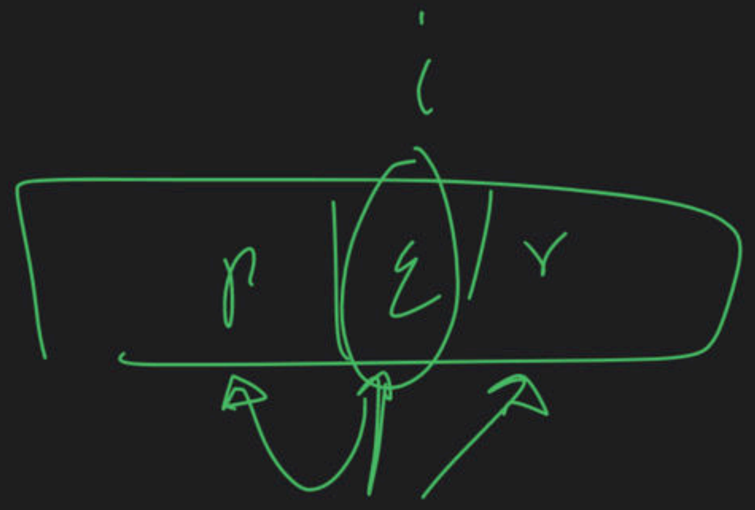


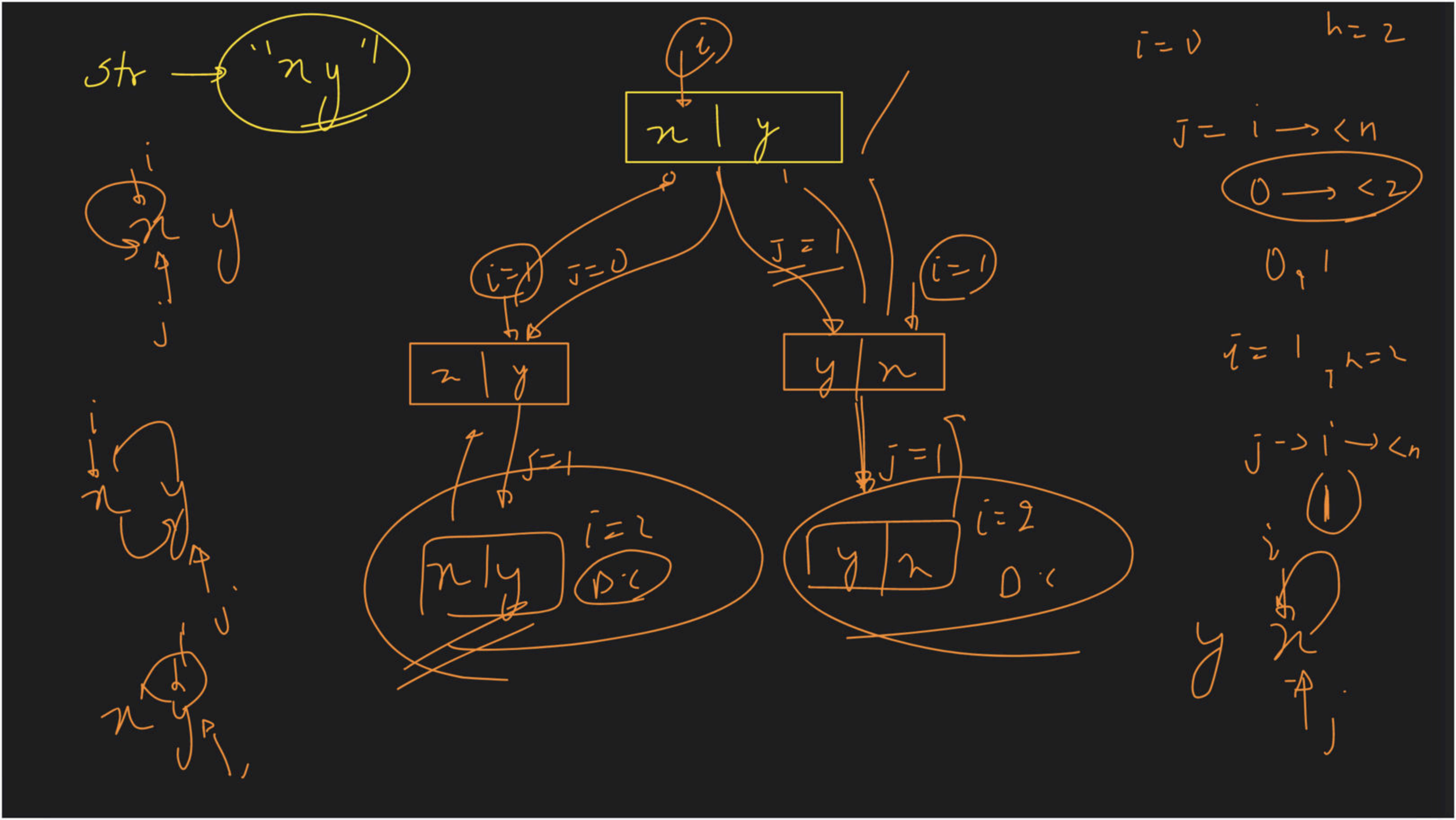
solve ("abc")
j=0
j=1
j=2
abc
bac
cba
acb
bca
cab
i/p → "abc"
j → loop
i → <n
i+1 → <n
B.C → print
wapes

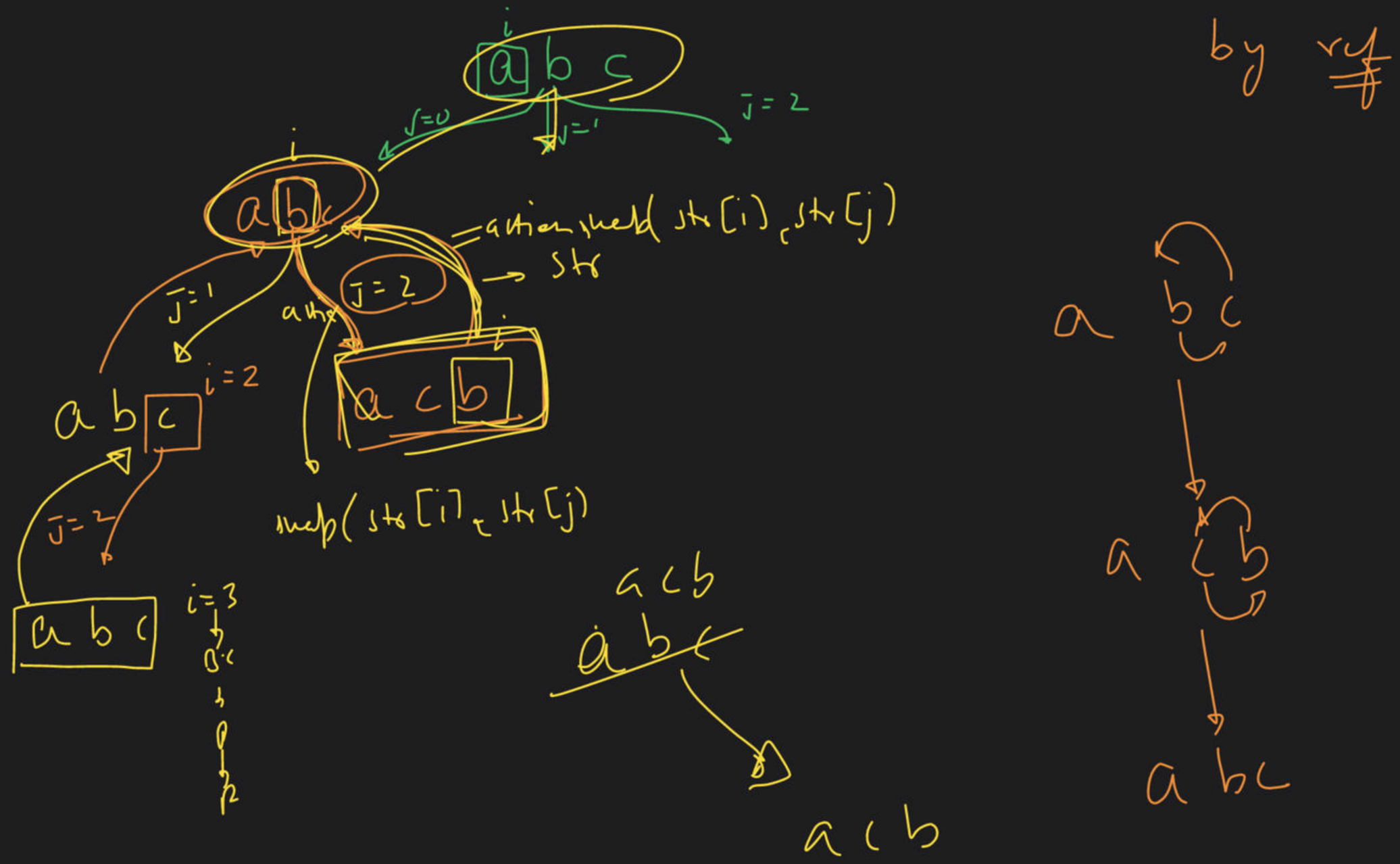


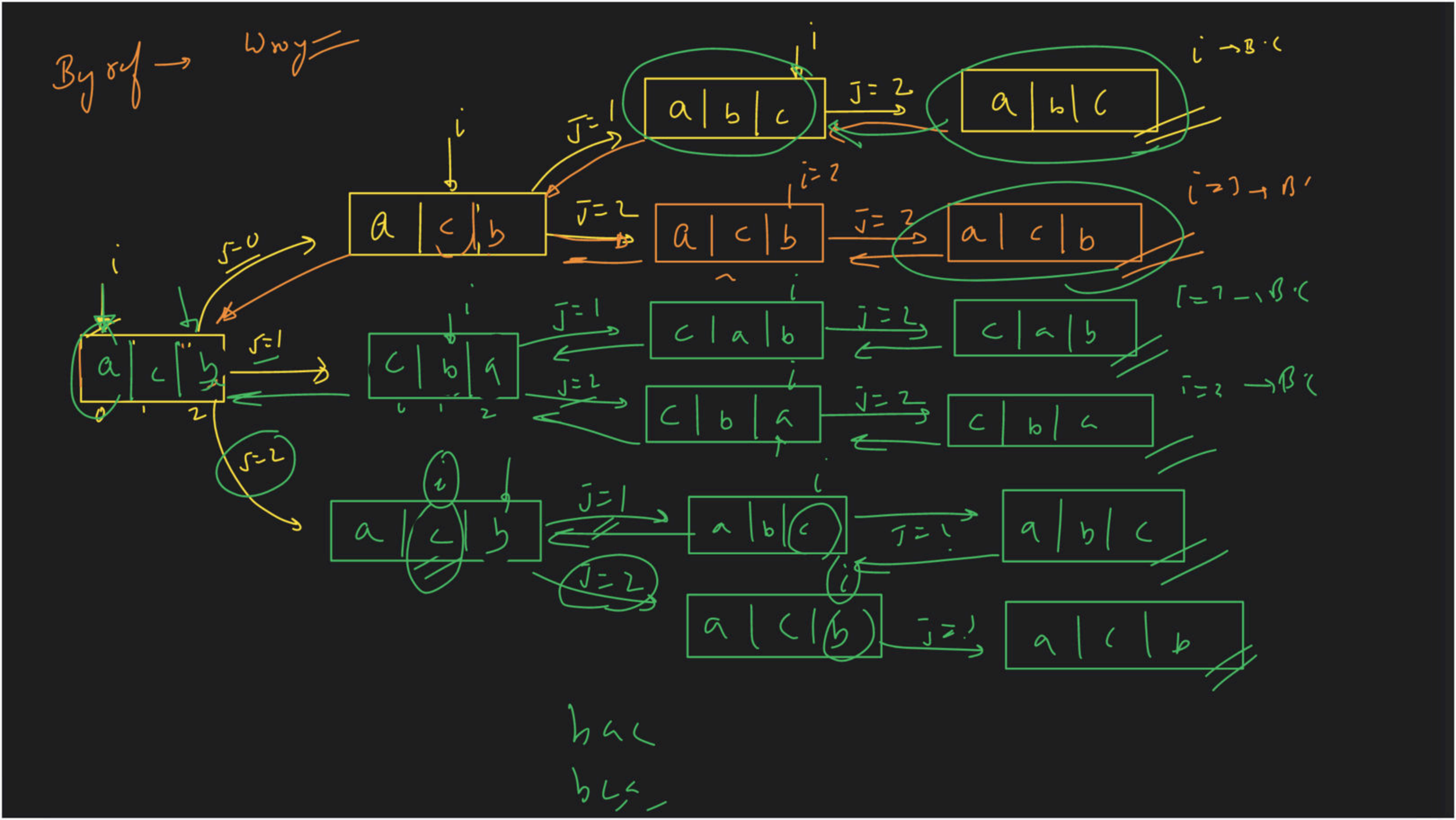
i=0
p | q | r
j=0
j=1
j=2
i=1
p | q | r
q | p | r
i=2
p | q | r
p | r | q
q | p | r
q | r | p
i=3
B.C
print
return
i=3
B.C
print
return
i=3
B.C
i=1
j → i → n
1 → n
i=2
j → i → <n
2 → <n
i=0
j
i=1
Rule → j → i → <n


i
p
q
r
0
1
2

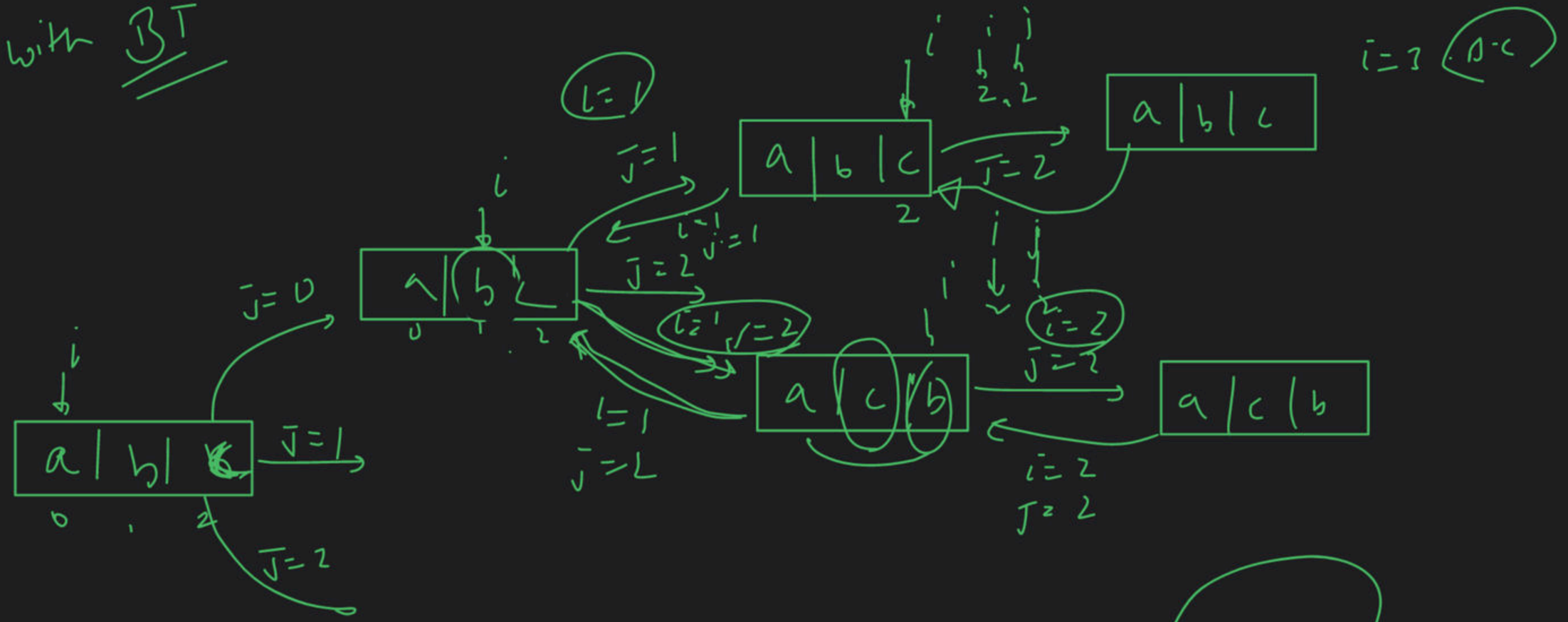

i
p
q
r


str
"xy"
i
x | y
i=1
j=0
j=1
i=1
z | y
y | x
j=1
i=2
x | y
i=2
y | x
i=0
h=2
j=i→<n
0→<2
0, 1
i=1
h=2
j→i→<n
1
i
j

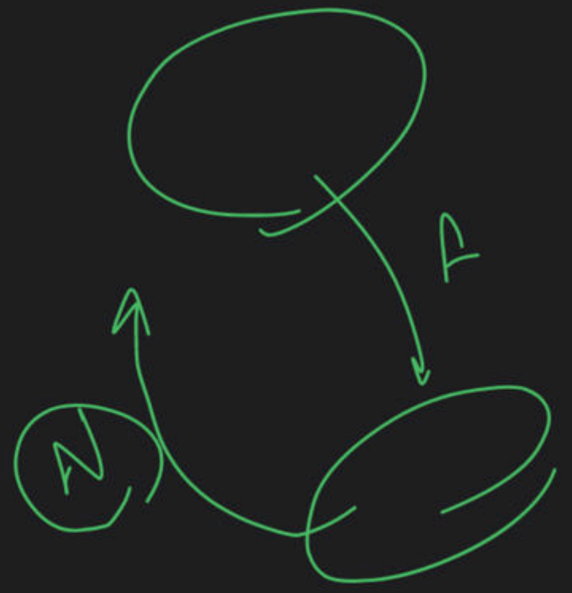

by ref
i
a b c
j=0
j=1
j=2
i
a b c
action swap(str[i], str[j])
str
j=1
j=2
action
i
a c b
i=2
a b c
j=2
swap(str[i], str[j])
a b c
i=3
a c b
a b c
a c b
a c b
a b c
a b c


By ref →
Way=
i
j=0
j=1
j=2
a | b | c
a | c | b
c | a | b
c | b | a
i=1 → B.C
i=2 → B.C
i=3 → B.C
bac
bca

with BT



a | b | c
0 1 2
i
j = 0
j = 1
j = 2
a | b | c
0 1 2
i
j = 1
i = 1
j = 1
j = 2
i = 1
j = 2
i = 1
j = 2
a | b | c
2
i
i j
2 2
j = 2
a | b | c
i = 3
a | c | b
i
i = 2
j = 2
a | c | b
i = 2
j = 2



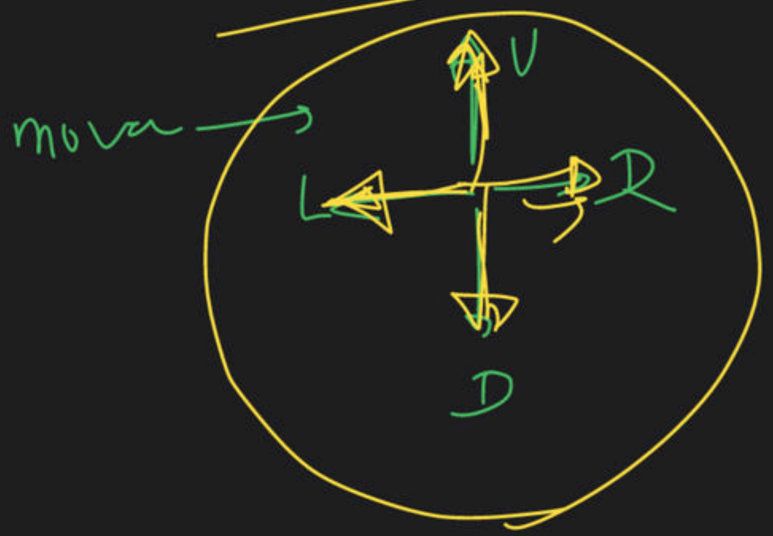
f
N

Rat in a Maze

print all possible ways!

0 → closed path

1 → Open path



	0	1	2	3
0	Sundar Rat 1	0	0	0
1	1	L	0	1
2	1	1	1	0
3	1	L	1	Sundar (1)

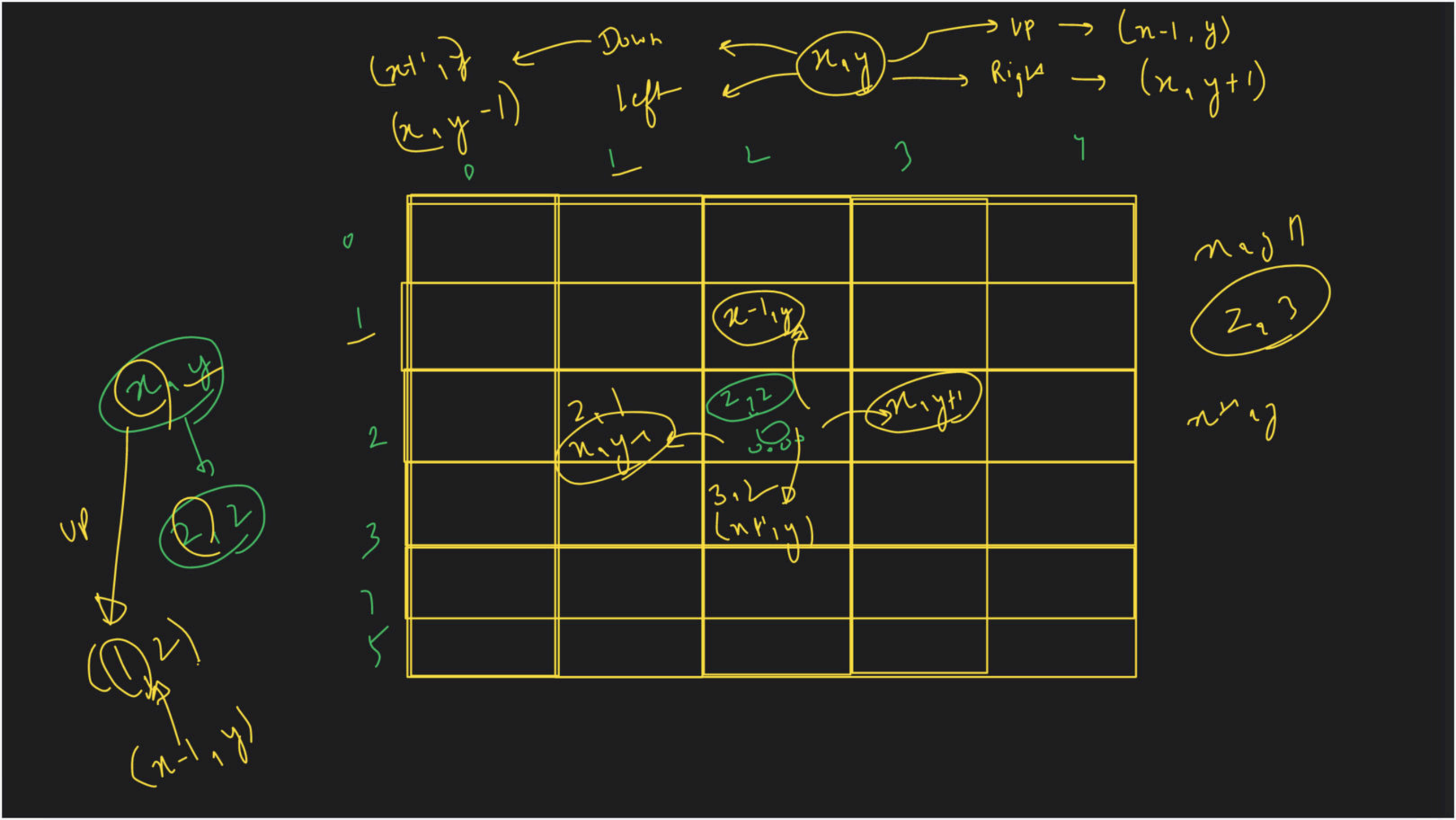
src → (0,0)

dest → (3,3)

(3,3)

Sundar Spotted

reached dest → (3,3)

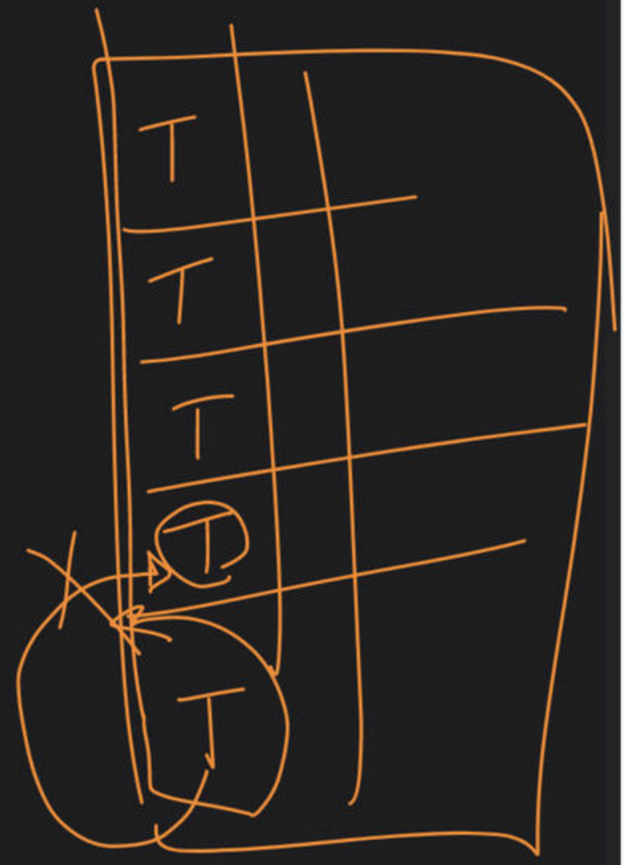

Down
Left
Up → (x-1, y)
Right → (x, y+1)
0
1
2
3
4
0
1
2
3
4
5
Up

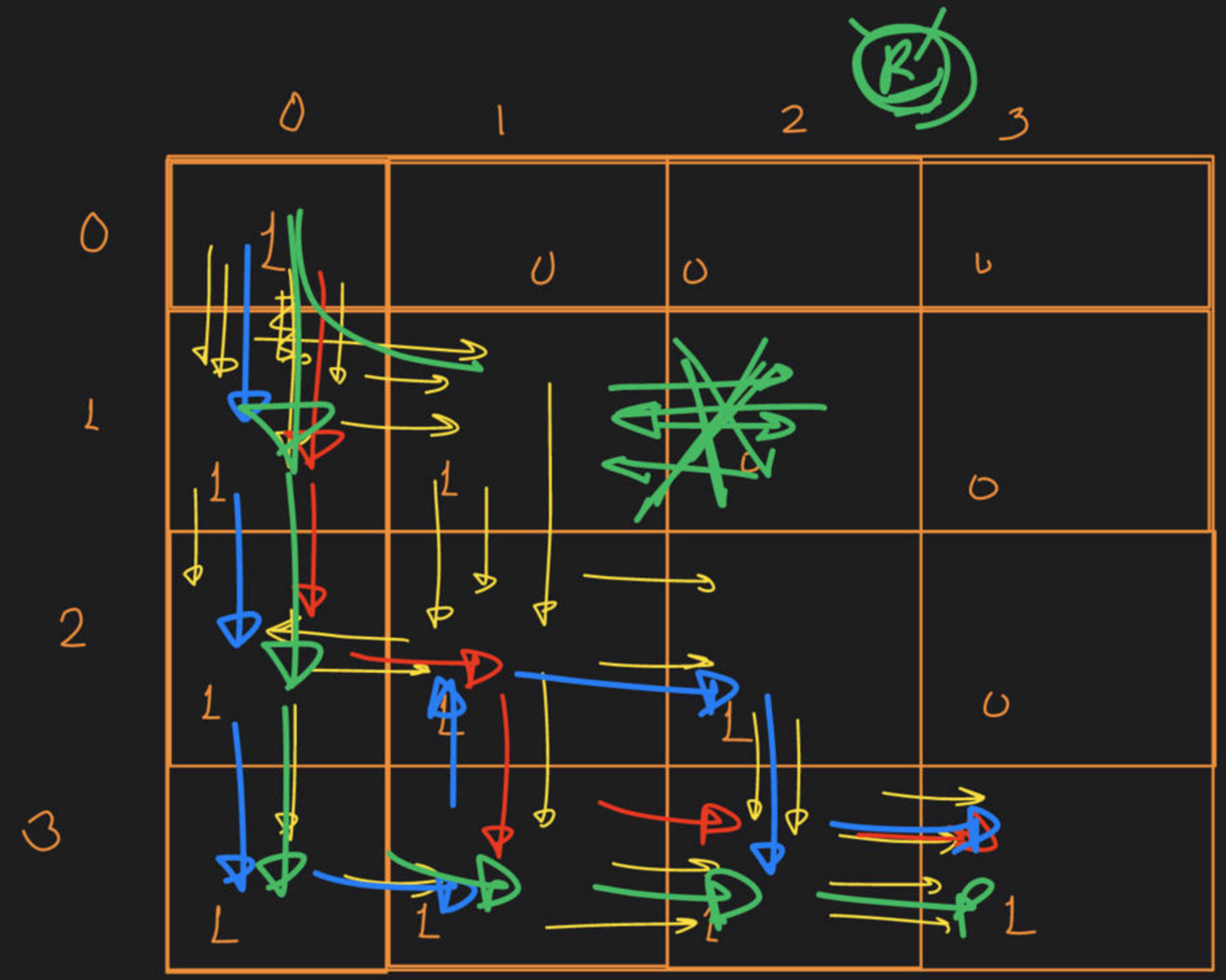
already visit

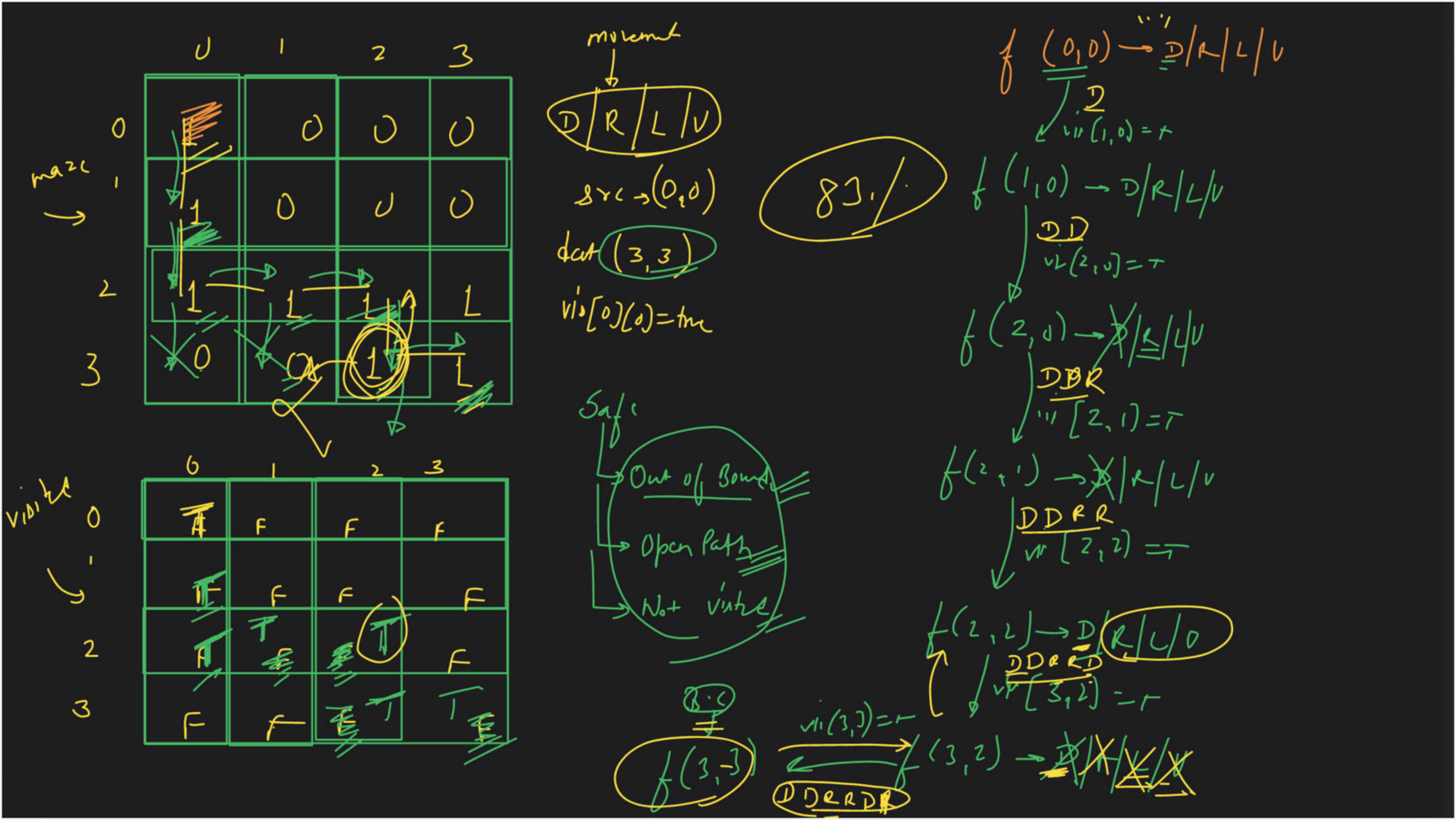
position

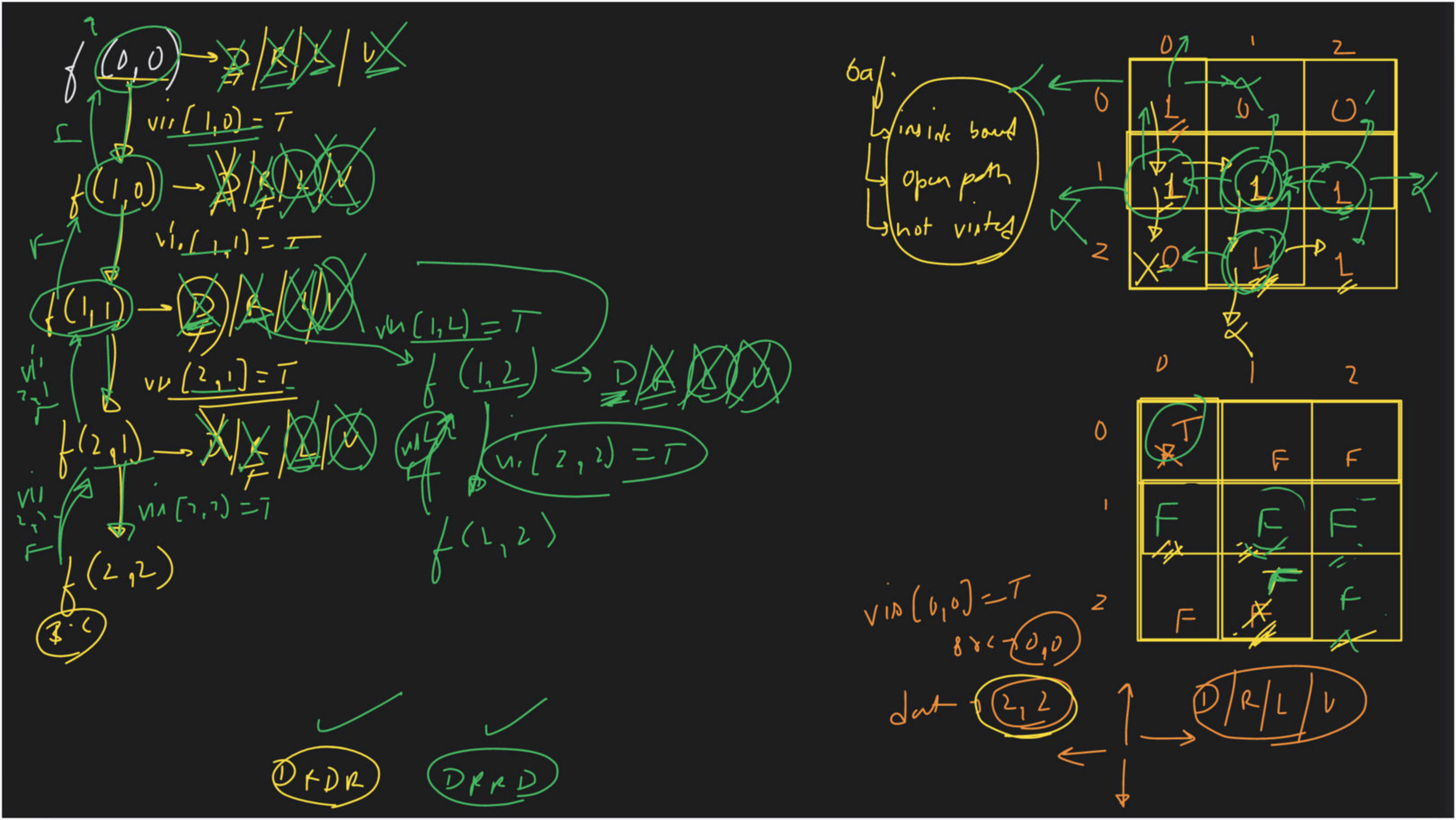
	0	1	2	3
0	1	0	0	0
1	1	1	0	0
2	1	0	1	0
3	1	1	1	0

infinite loop

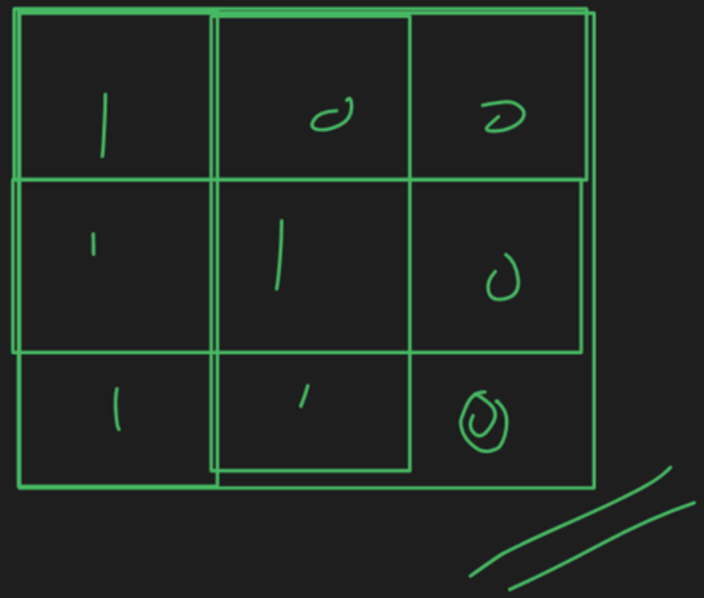


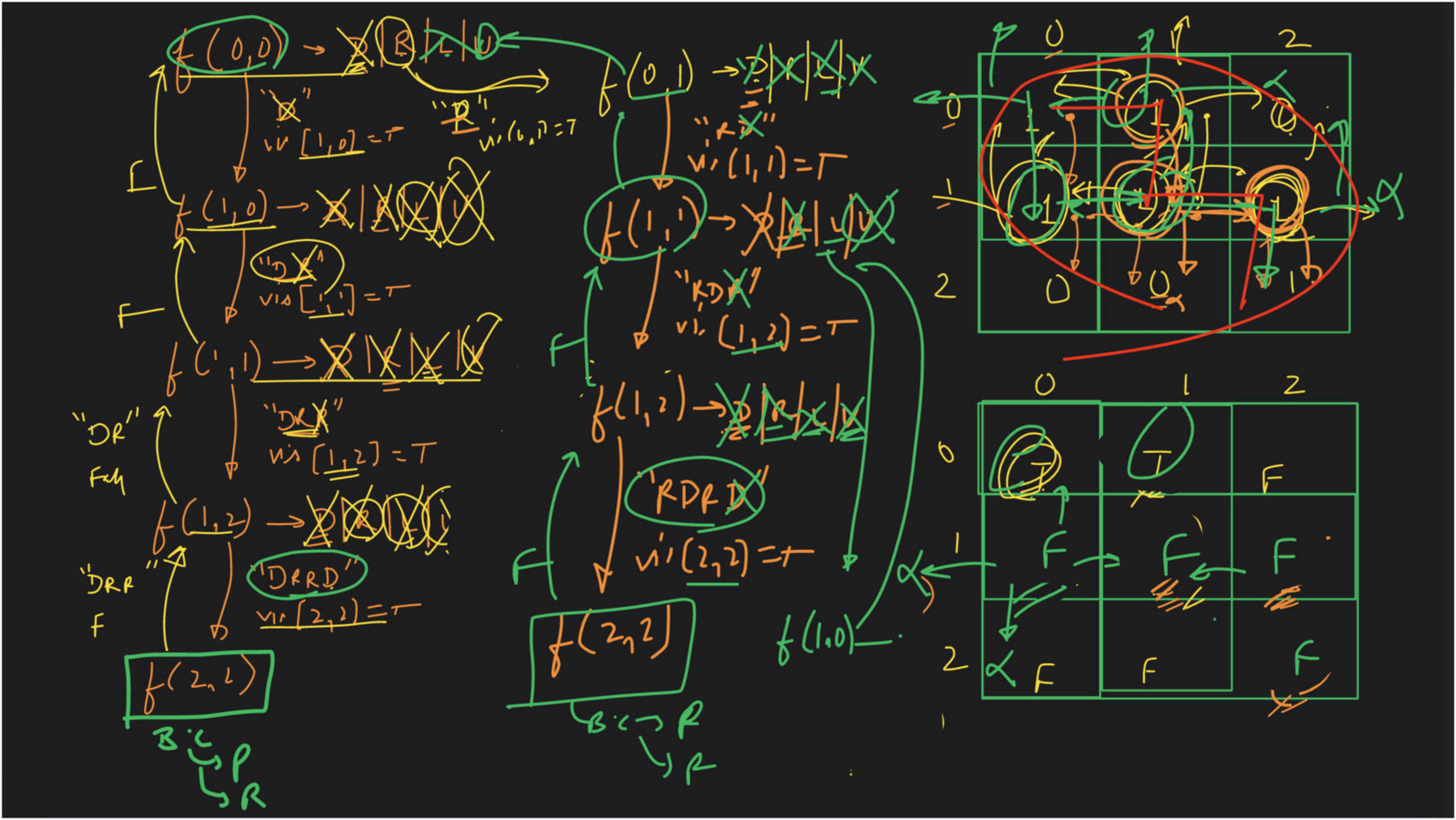

0
1
2
3
0
1
2
3


maze
visited
movement
D R L U
src → (0,0)
dest (3,3)
vis[0][0]=true
Safe
Out of Bound
Open Path
Not visited
83.1
f(0,0) → D/R/L/U
D
vis[1,0]=T
f(1,0) → D/R/L/U
DD
vis[2,0]=T
f(2,0) → X/R/L/U
DDR
vis[2,1]=T
f(2,1) → X/R/L/U
DDRR
vis[2,2]=T
f(2,2) → D/R/L/U
DDRRD
vis[3,2]=T
f(3,2) → X/X/X/X
vis[3,3]=T
DDRRDR
Base case
f(3,3)

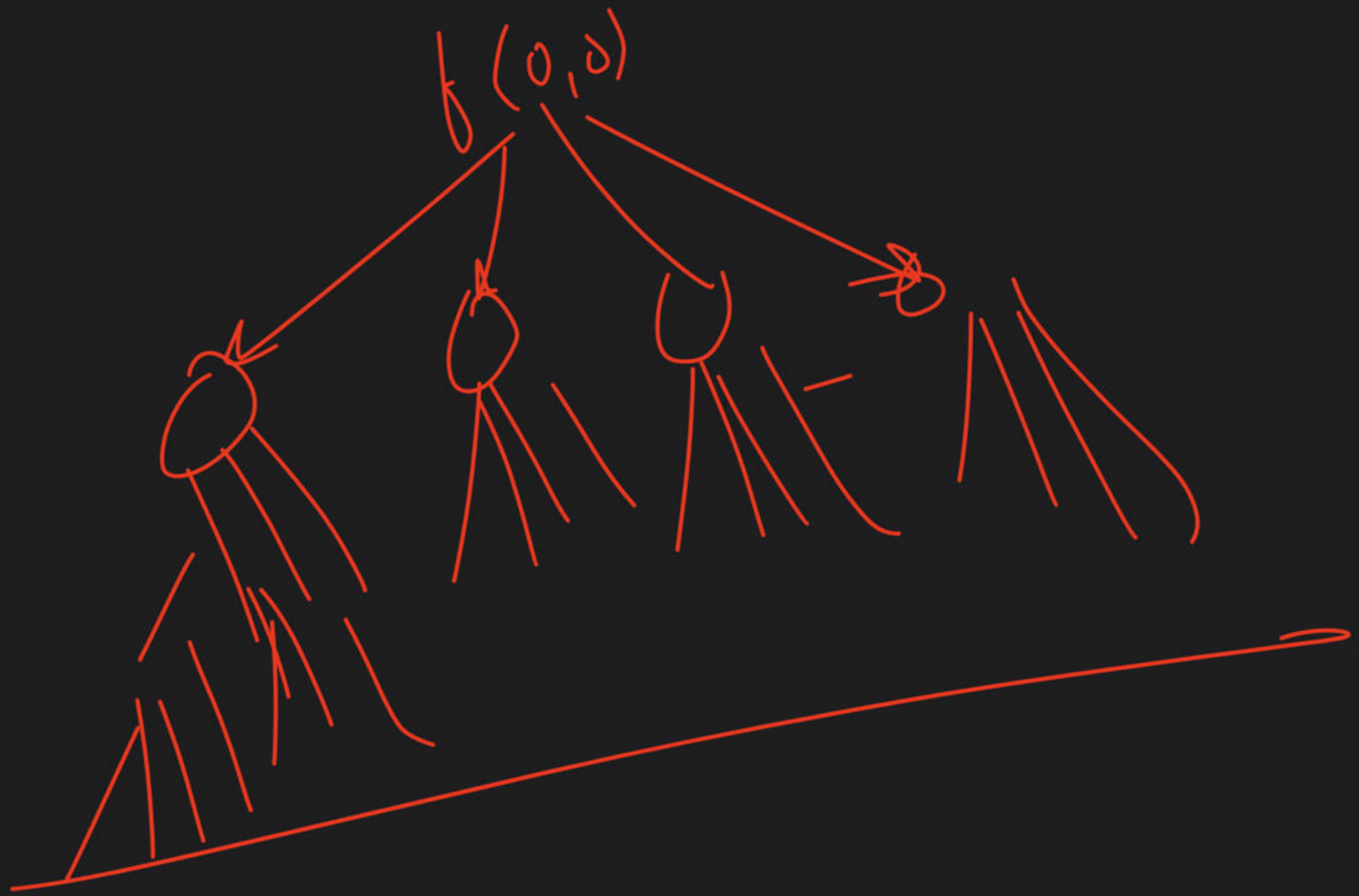


f(0,0)
vis[1,0] = T
f(1,0)
vis[1,1] = T
f(1,1)
vis[1,2] = T
vis[2,1] = T
f(1,2)
f(2,1)
vis[2,2] = T
vis[2,2] = T
f(2,2)
f(1,2)
B.C
DRDR
DRRD
Gaf.
inside bound
open path
not visited
vis[0,0] = T
src → 0,0
dst → 2,2
D/R/L/U





f(0,0) → D | R | L | U
"D" vis [1,0] = T
"R" vis (0,1) = T
f(1,0) → D | R | L | U
"DR" vis [1,1] = T
f(1,1) → D | R | L | U
"DRR" vis [1,2] = T
f(1,2) → D | R | L | U
"DRRD" vis [2,2] = T
f(2,2)
B.C → P → R
"DR" Fah
"DRR" F
f(0,1) → D | R | L | U
"RD" vis (1,1) = T
f(1,1) → D | R | L | U
"RDR" vis (1,2) = T
f(1,2) → D | R | L | U
"RDRD" vis (2,2) = T
f(2,2)
B.C → R → R
f(1,0)
0 1 2
0 1 2
0 1 2
T T F
F F F
F F F


f (0,1)