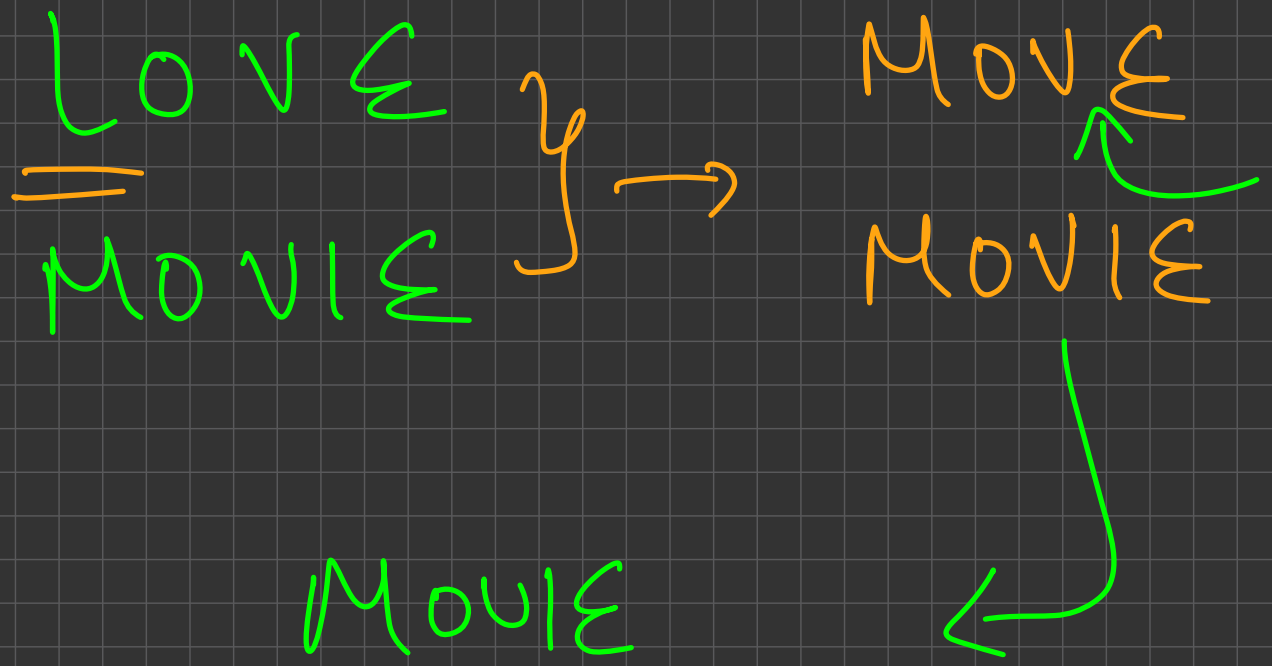
LOVE → LOIE

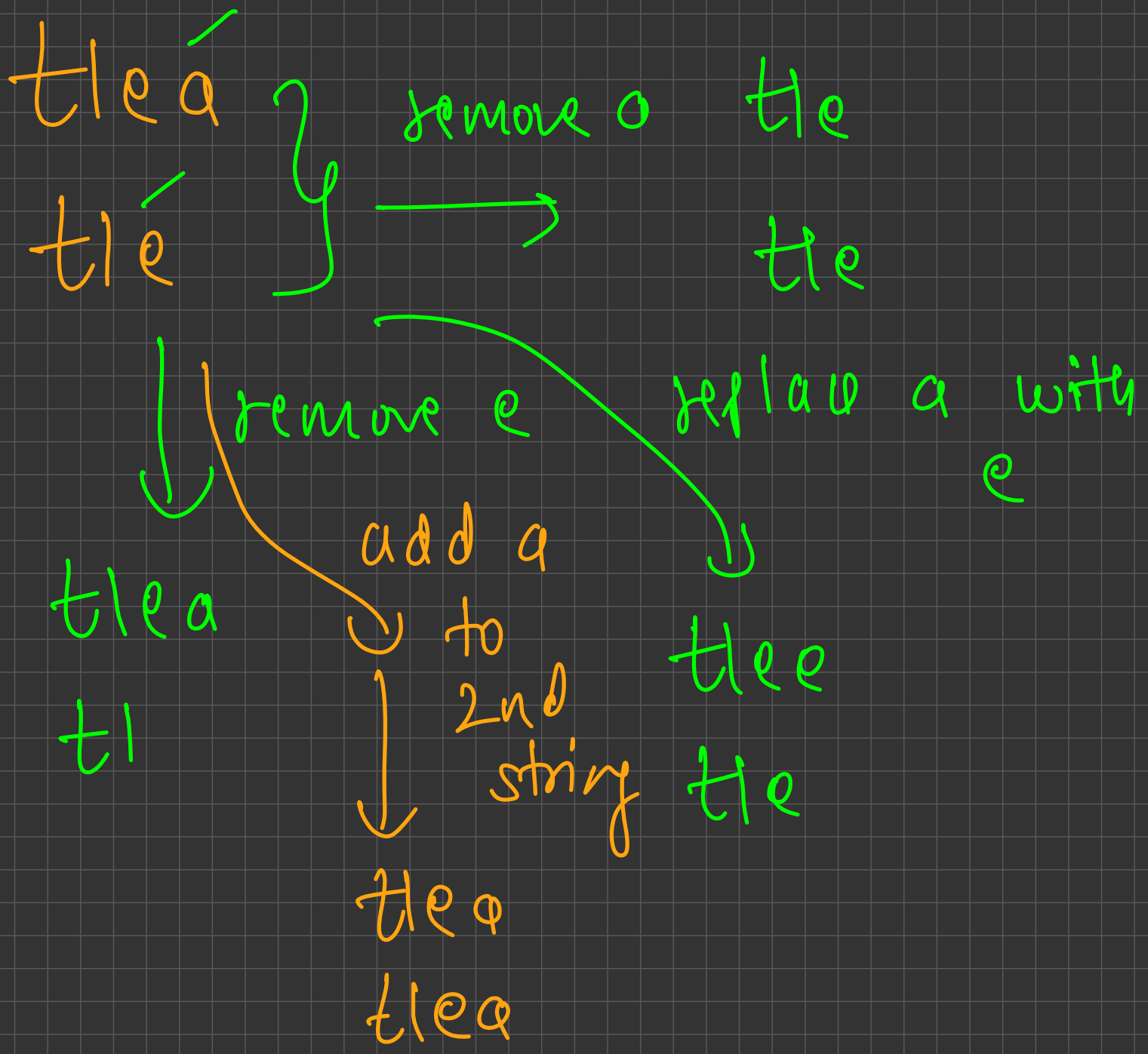
MOVIE → MOVIE

{ LOIE, MOIE }

MOIE

MOIE


LOVE
=
MOVIE
MOVE
MOVIE
MOVIE
MOVIE


tlea
tlé
remove o
tle
tle
remove e
replace a with e
tlea
tl
add a to 2nd string
tlee
tle
tlea
tlea

tlea

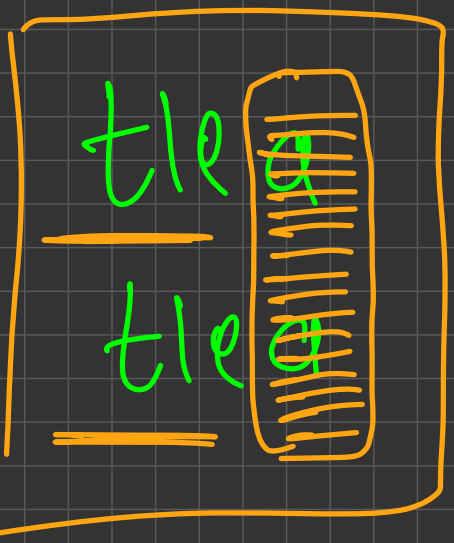
tle

remove a
1st string

tle

tle

add a
to 2nd string

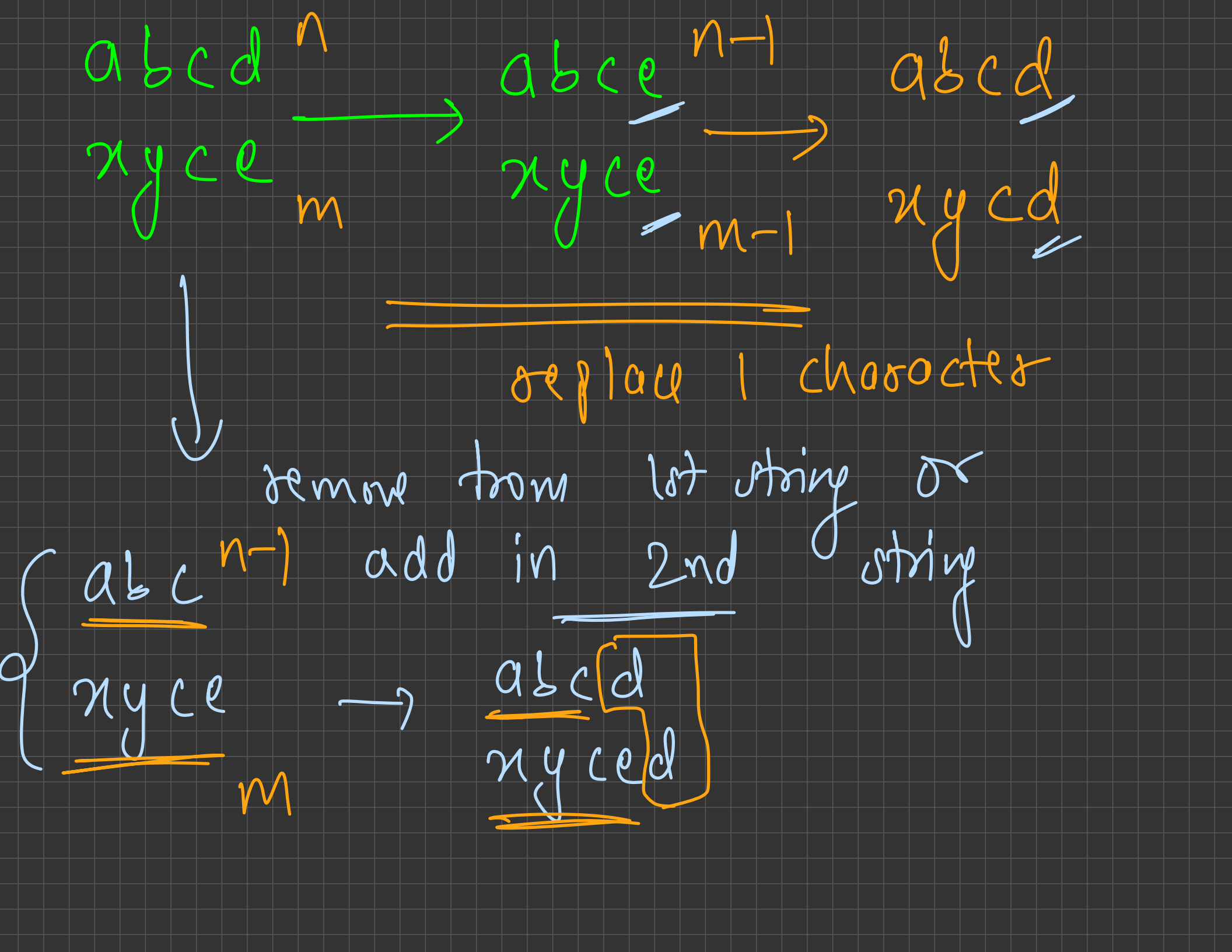


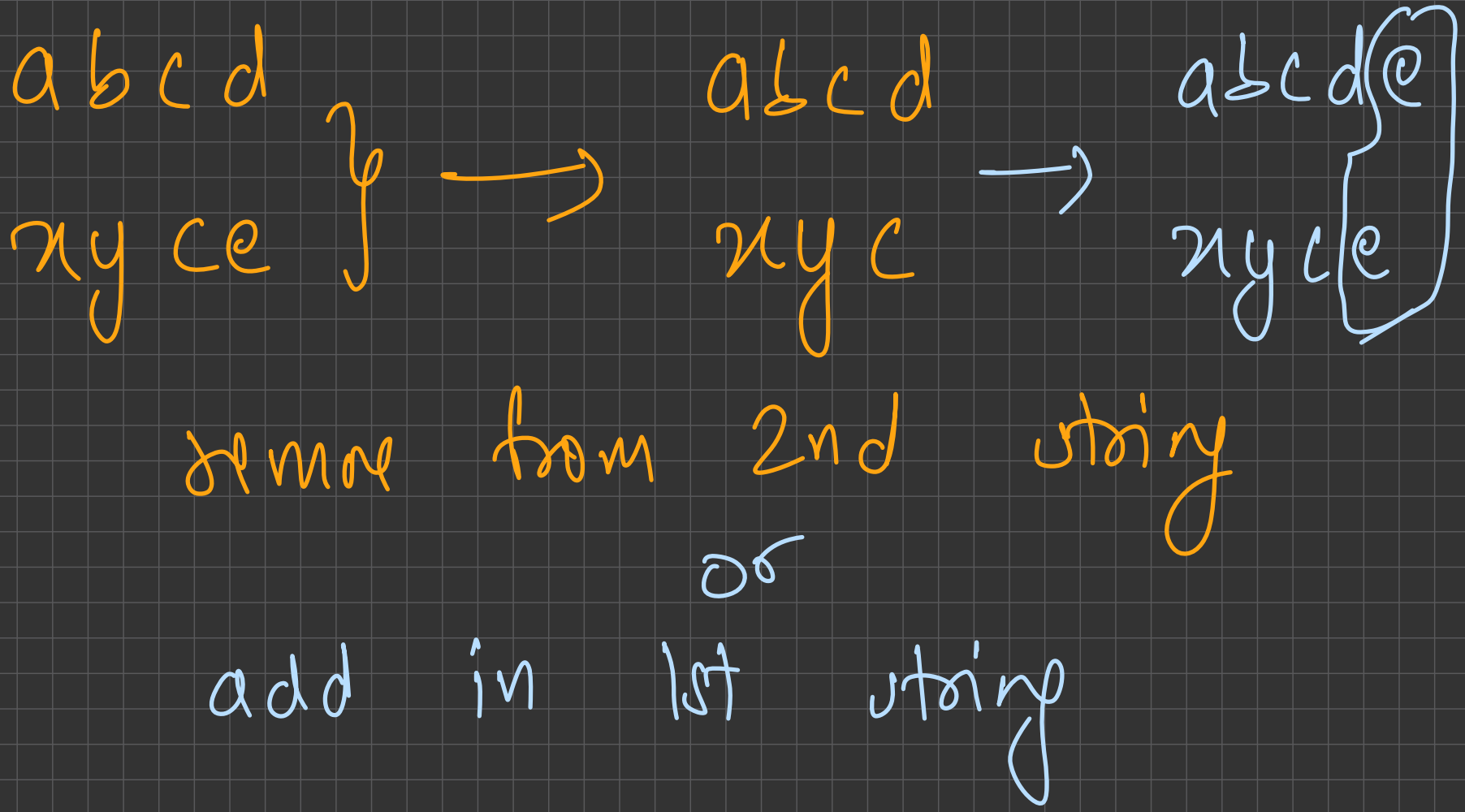
A of length n

B of length m

① last char of A = last char of B

② $A[i] \neq B[0]$


abcd n
xyce m
abce n-1
xyce m-1
abcd
xycd
replace 1 character
remove from 1st string or
abc n-1 add in 2nd string
xyce m
abcd
xyced


abcd
xyce
abcd
xyc
abcd
xyce
remove from 2nd string
or
add in 1st string

dp[i][j] = edit distance b/w first i characters of A and first j characters of B

abcd ← i

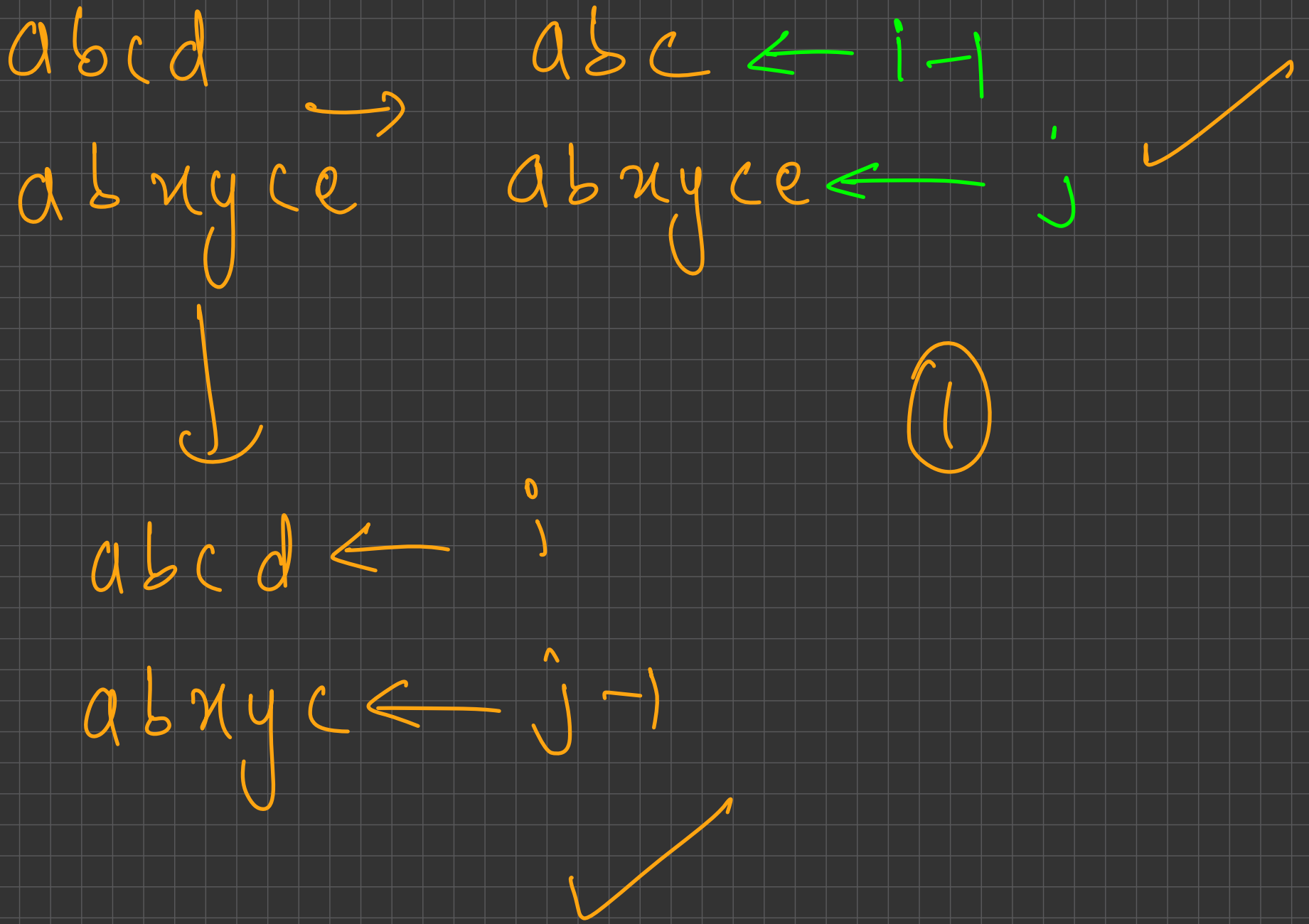
abxycde ← j

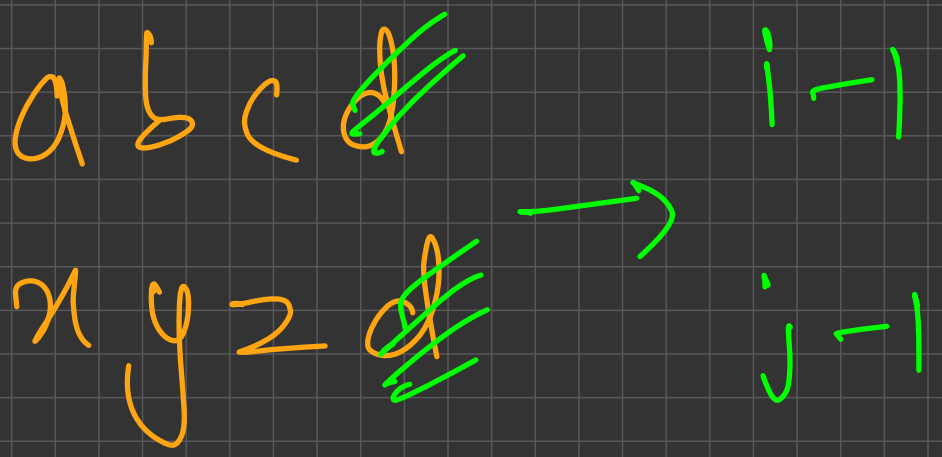
abc[e] i-1

abxycd[e] j-1 replaced

abc[d] → i-1

abxycd[d] → j-1


abcd
abxycde
abc
i-1
abxycde
j
1
abcd
i
abxyc
j-1


a b c d
x y z d
i-1
j-1

if $A[i-1] == B[j-1]$

$dp[i][j] = dp[i-1][j-1]$

else

① replace $dp[i][j] = dp[i-1][j-1] + 1$

② remove from 1st string $dp[i][j] = dp[i-1][j] + 1$

③ remove from 2nd string $dp[i][j] = dp[i][j-1] + 1$

abc → ab → a → ""

xyz → xyz → xyz → xyz

dp[i][0] = i

dp[0][i] = i

dp[0][0] = 0

Base case

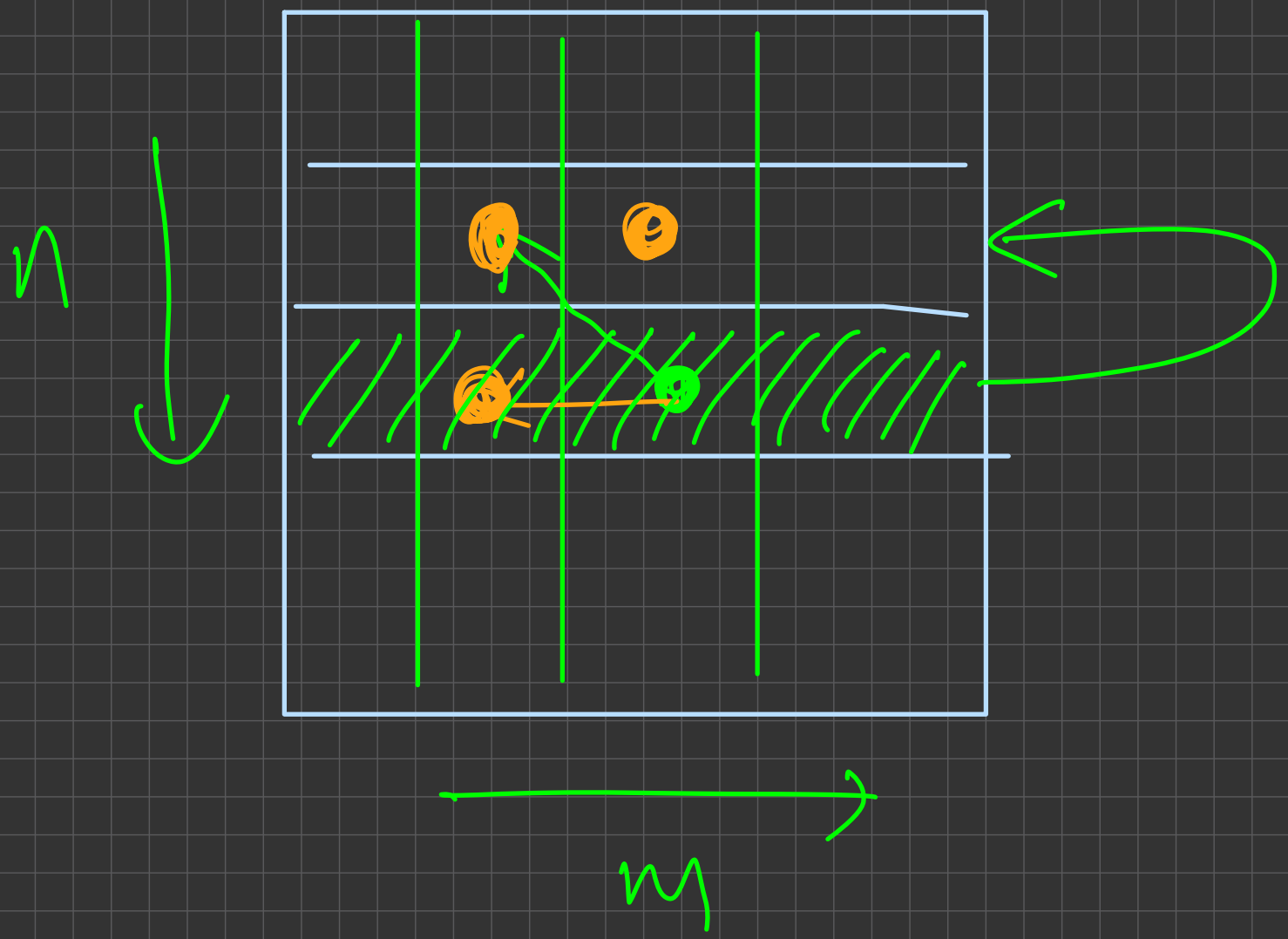
Final subproblem :

$dp(n)[m]$

Time Complexity :

$O(n \cdot m) \cdot O(1)$

Space Complexity : $O(n \cdot m)$



$O(m)$