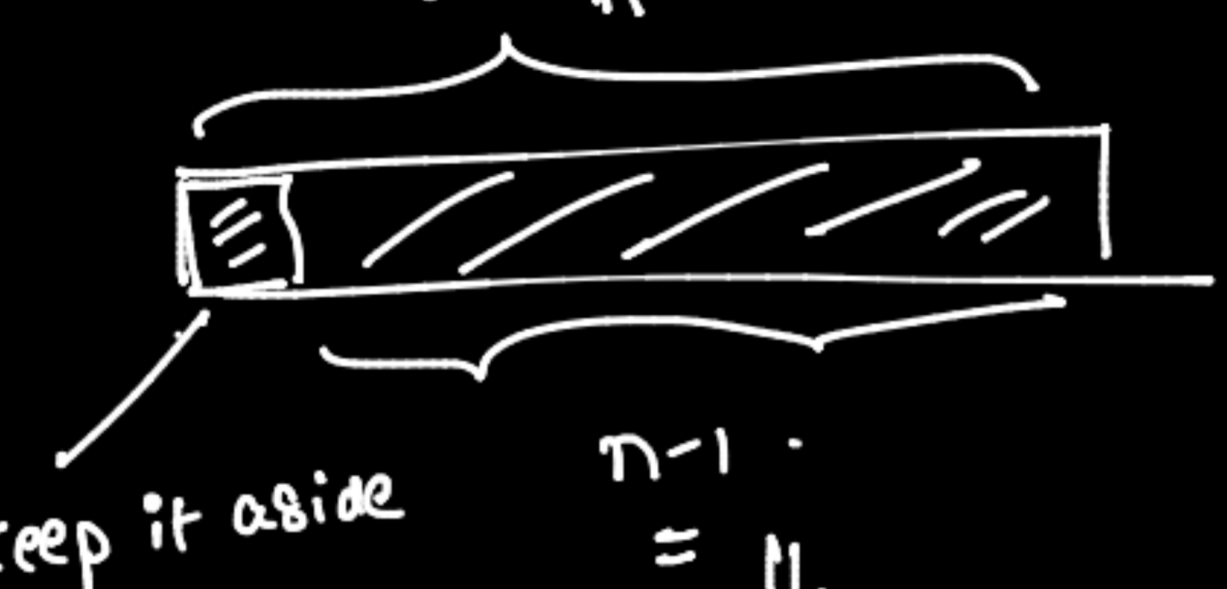
str [| | | . . . . . |] → towards

x = non-x

str " a x b c x d x f x "

⇓

a b c d f x x x x



⇓ recursion

s1 = ⓐ x b c x d x

⇓

smaller inst of the org. prob

smaller inst of same prob

recursion (assistant)

Recursion

⇓

a + b c d x x x

= a b c d x x x

s2 = ⓧ a b c x x d

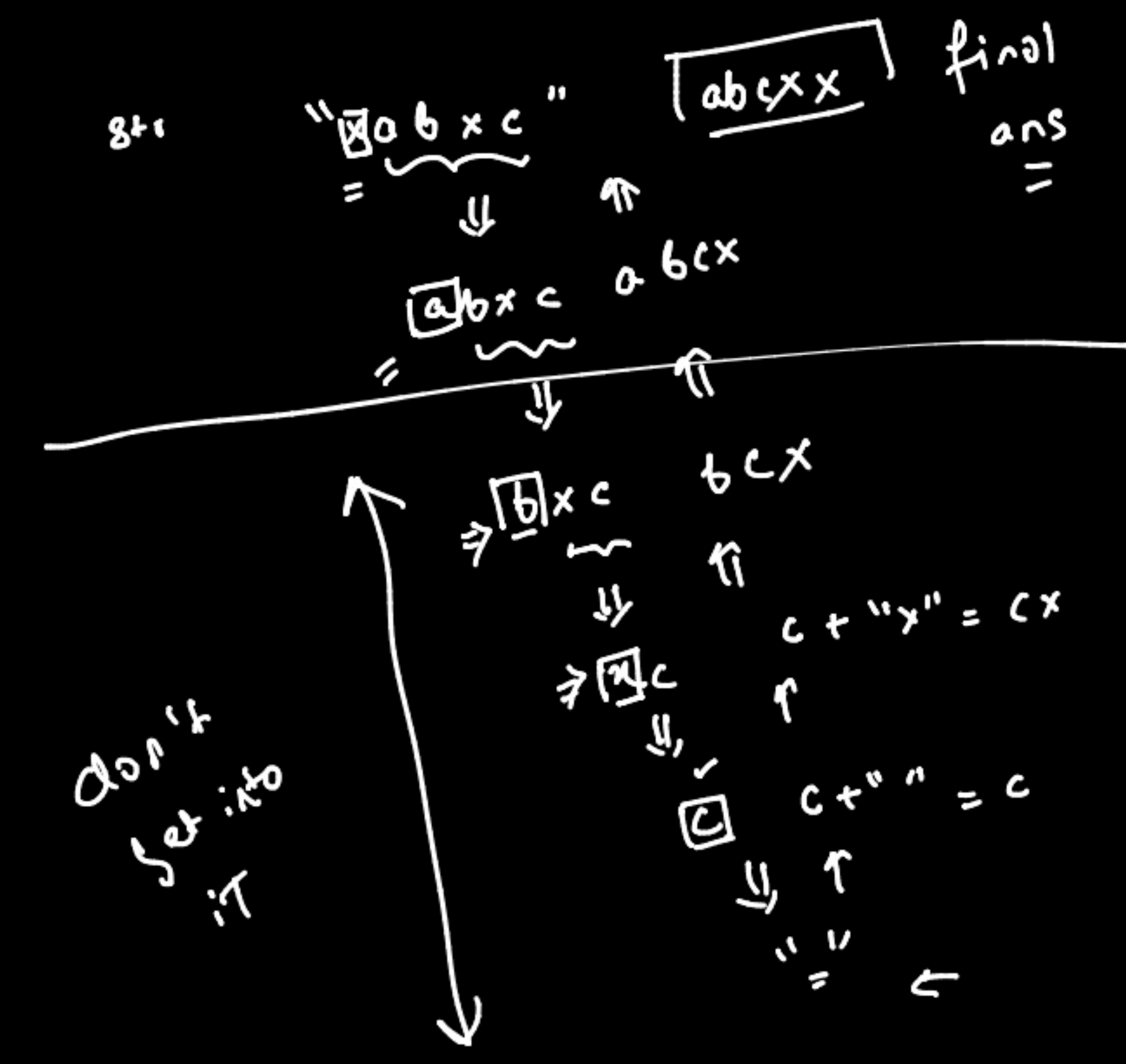
⇓

recursion

⇓

a b c d x x + x

= a b c d x x x



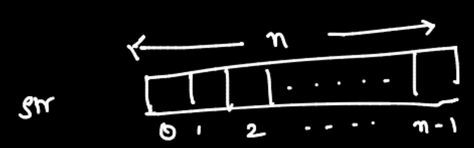
Replace pi's

str: pi p pi

o/p: 3.14 p 3.14

str: p pi p

o/p: p 3.14 p



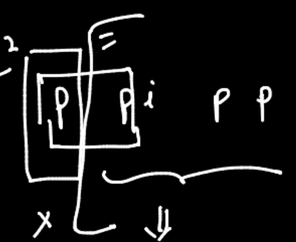
replace pi with "3.14"

ex1

⇒ 3.14 ⇓ recursion

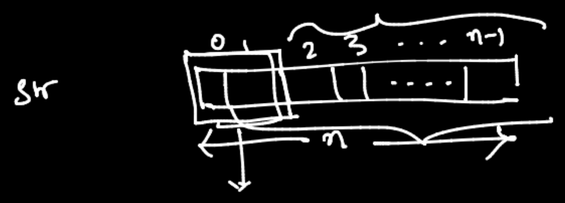
3.14 p3.14 p

ex2



⇓ recursion

"p" 3.14 p p



pi ?

X

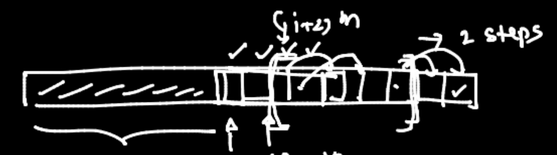
↳ replace pi with 3.14

↳ replace pi in the remaining substr of length n-2 with 3.14 ⇒ recursion

↳ do nothing

↳ replace pi into 1 as of length n-1 with 3.14 ⇒ recursion

str[ ] =



pi ?

X

↳ replace all pi with 3.14 in remaing str starting from (i+1)th idex ⇓ recursion

↳ replace "pi" with "3.14"

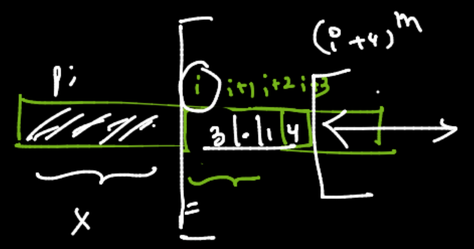
⇒

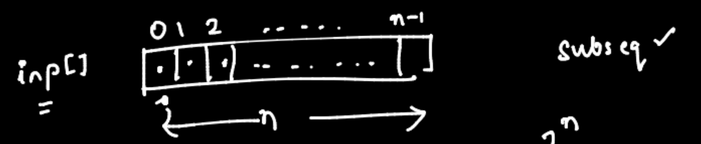
↳ shift all char. starting from (i+2)th idex 2 steps to the right

↳ str[i] = '3';
str[i+1] = '.';
str[i+2] = '1';
str[i+3] = '4';

you can nowon.

⇒ replace pi with 3.14 in the remaing str starting from the (i+4)th. ⇒ recursion





subseq ✓

2^n =

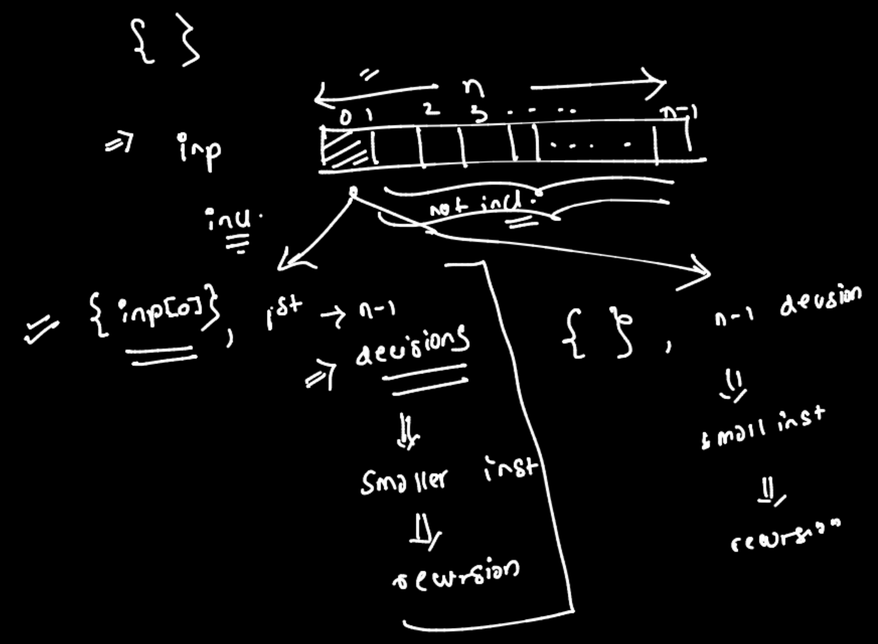
$2 \cdot 2 \cdot 2 \cdot 2 \cdots 2 \Rightarrow 2^n$

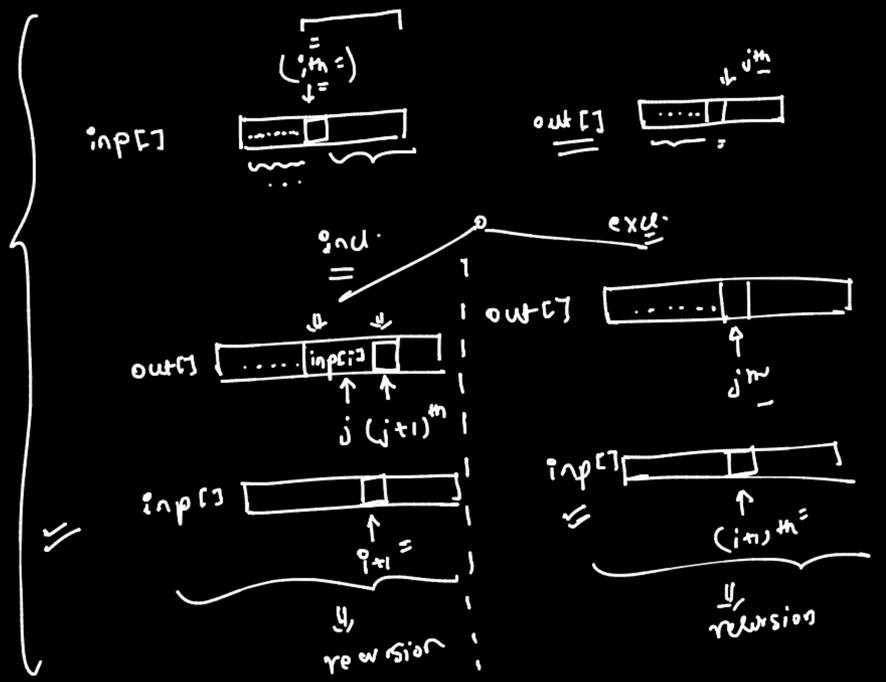
$n = 3$

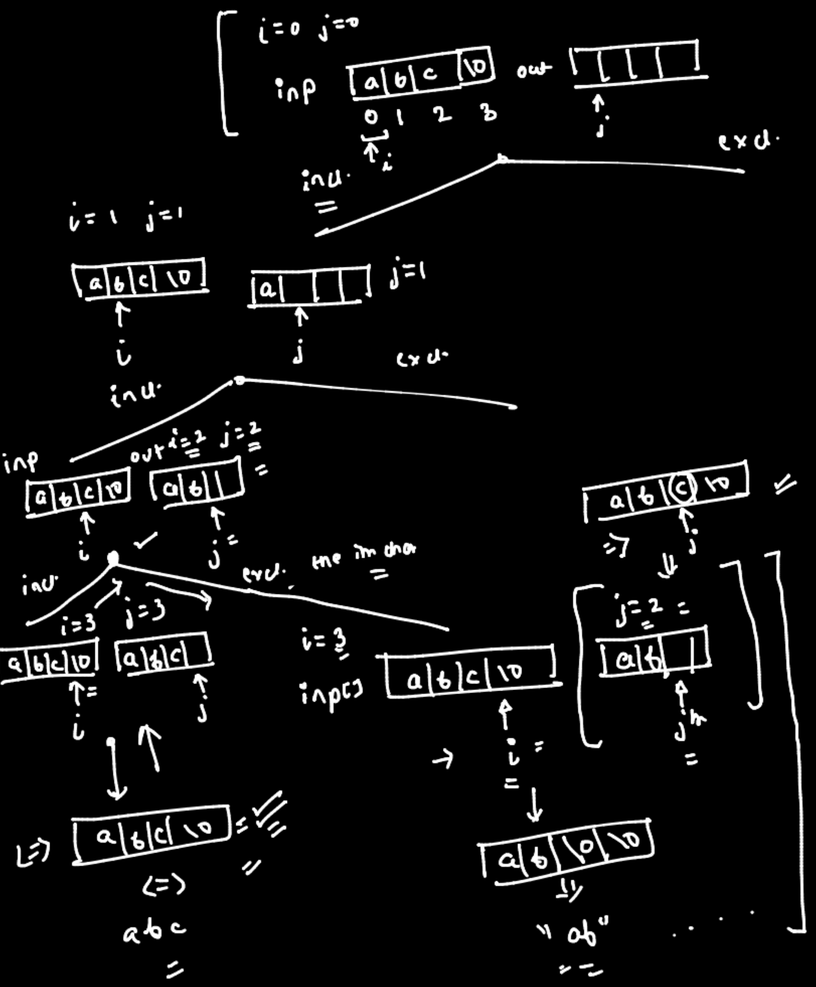
!! a b c"

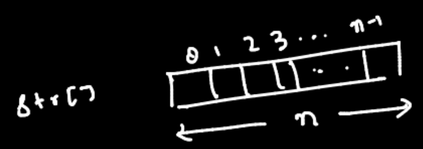
✗	✗	✗	⇒	" "
✗	✗	✓	⇒	"c"
✗	✓	✗	⇒	"b"
✗	✓	✓	⇒	"bc"
✓	✗	✗	⇒	"a"
✓	✗	✓	⇒	"ac"
✓	✓	✗	⇒	"ab"
✓	✓	✓	⇒	"abc"

↳ For a string str of length, if we want to gen. all the 2^n subseq. then for each of the n char. we've to make a choice.






i=0 j=0
inp
a b c \0
0 1 2 3
out
i
j
exd.
indu.
i=1 j=1
a b c \0
a
j=1
i
j
indu.
exd.
inp
out
i=2 j=2
a b c \0
a b
indu.
exd. the in char
i=3 j=3
a b c \0
a b c
i=3
inp[3]
a b c \0
a b c \0
(=)
abc
a b c \0
j=2
a b
a b c \0
"ab"

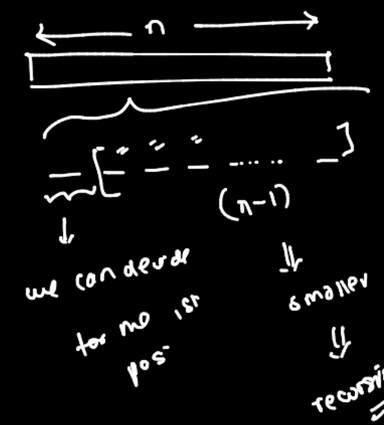


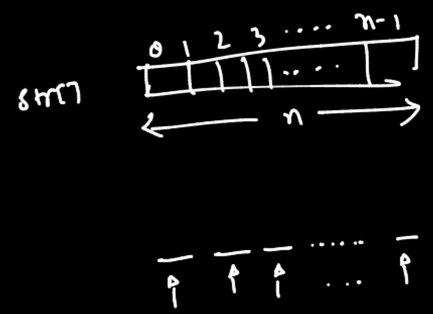
n!

"abc" n=3 ⇒ 3! = 6

a b c
a c b
b a c
b c a
c b a
c a b

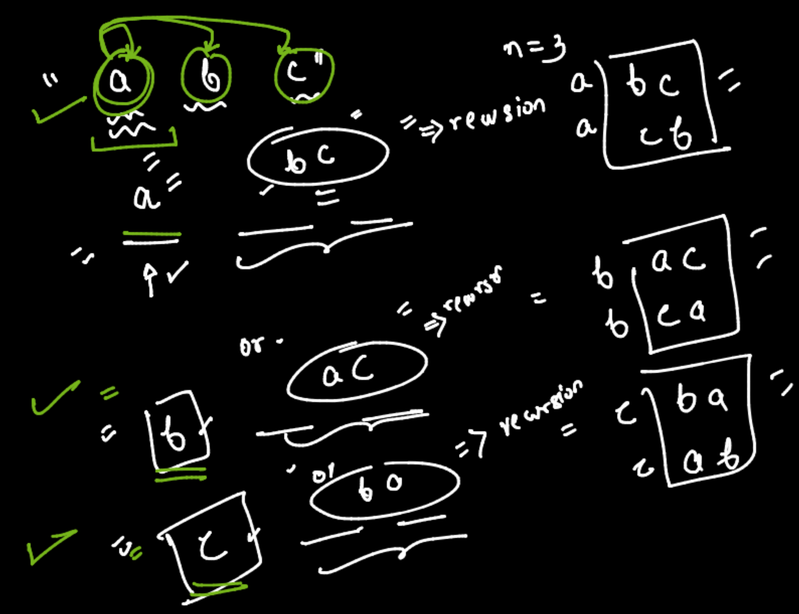
6 permutations

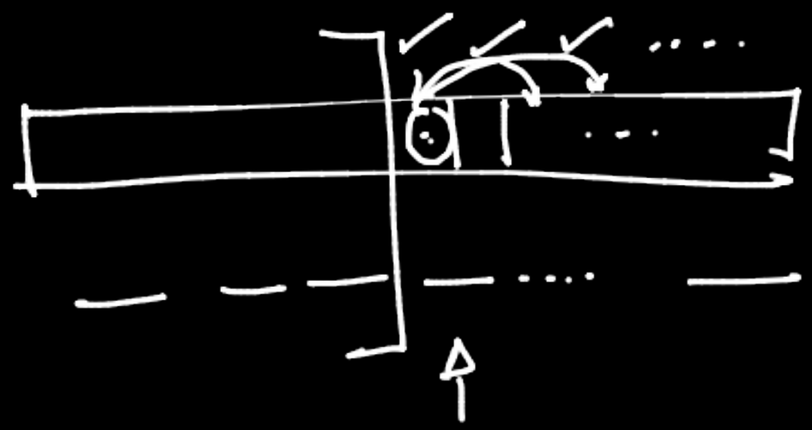




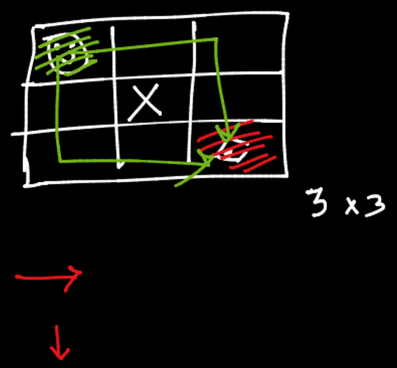
n!

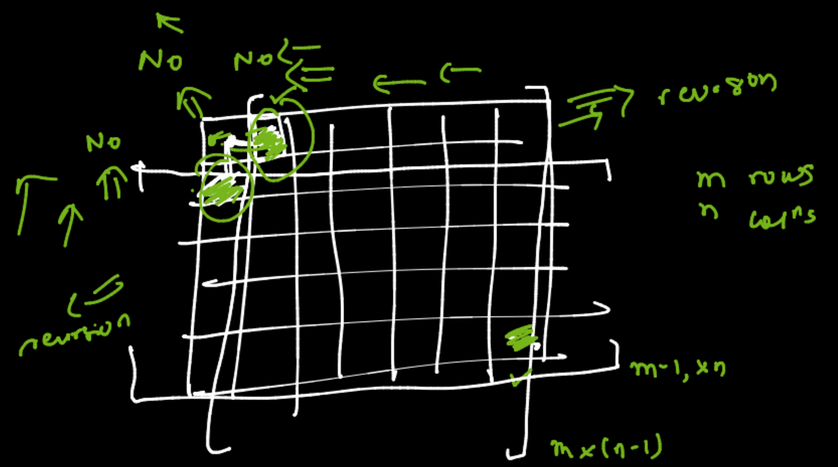
1) "a b c"

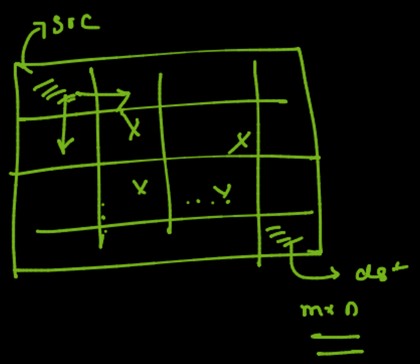


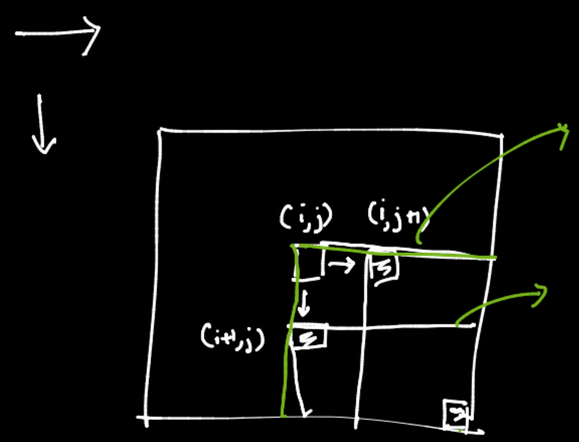


j^{th} $\quad j = i$ to $n-1$


X
3 x 3


No
No
No
return
return
m rows
n cols
m-1, n-1
m x (n-1)


src
dest
m x n


(i,j)
(i,j+1)
(i+1,j)