

Bubbles up the biggest element.



Base Case

→

$n == 1$

↪

return 1

// already sorted

$f(i, n)$ = if ($a[i] > a[i+1]$) → $f(i+1, n)$
swap($a[i], a[i+1]$)

bring the biggest
element of the
array $[i, n]$
at the $(n-1)^{th}$
index.

$\forall j \in [0, n-1]$
 $f(0, n)$

1) Remove an array recursively.

2) LL remove

data remove



$[0, 1, 2, 3, 4, 5]$
start end

$$f(s, e) = \text{sum}(a[s], a[e]) \rightarrow f(s+1, \textcircled{e-1})$$

$s \leq e$

Start Sin ++



$$f(s, e) = f(s, \underline{e+1}) \rightarrow \underline{\text{swap}(lls, lle)}$$

$s+1$ $s! = e$
 $s = s.next$

Qⁿ Given a string str, print the
subsequences of the string recursively.

"abc"

↳

"a"
↳
"b"

"c"

"ab"

"bc"

"ac"

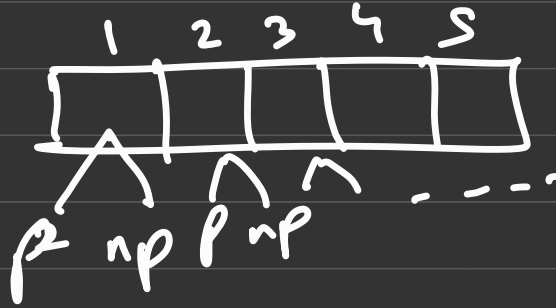
"abc"

""

order doesn't matter

→ for given n elements how many subsets are possible

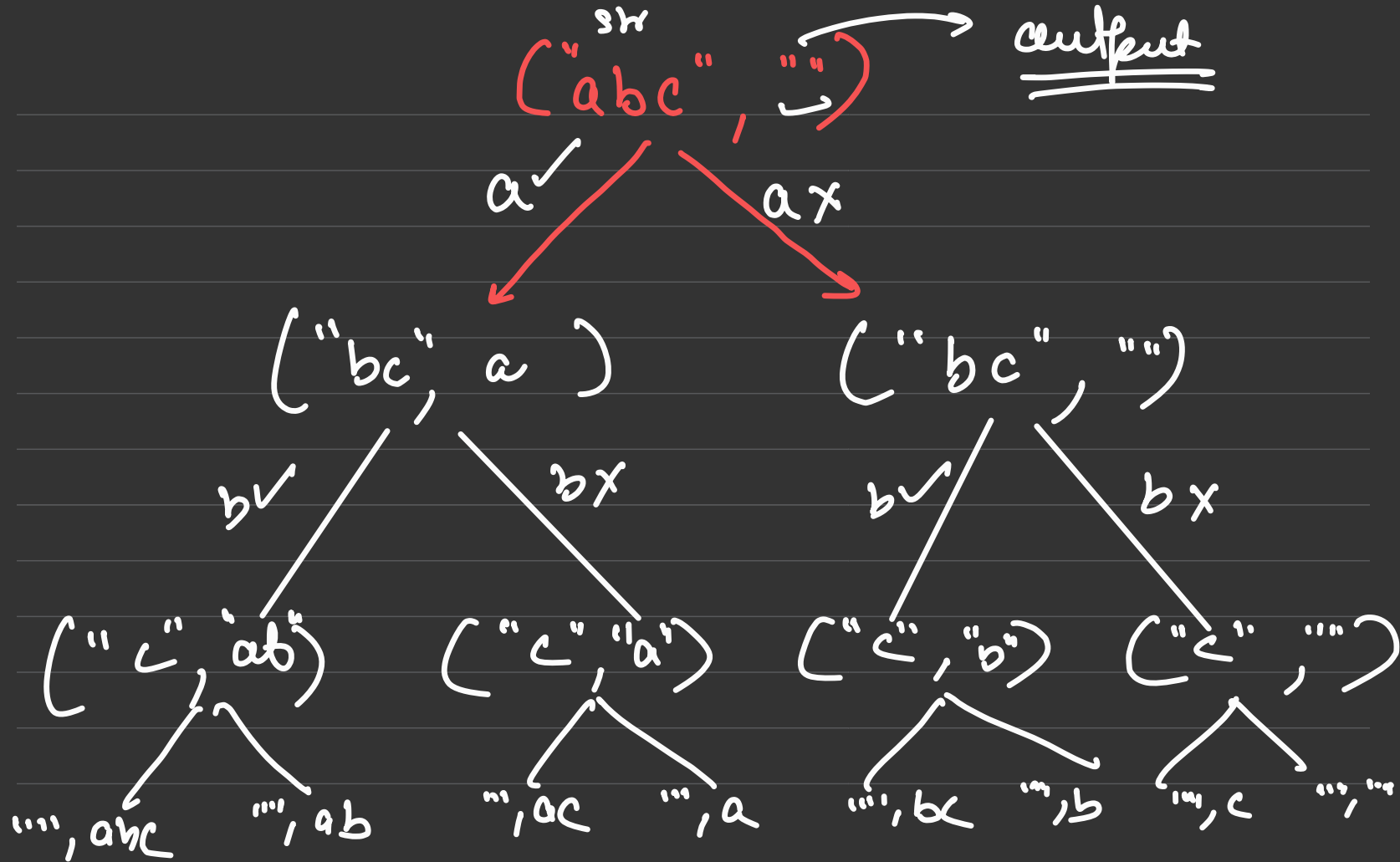
$$2^n$$

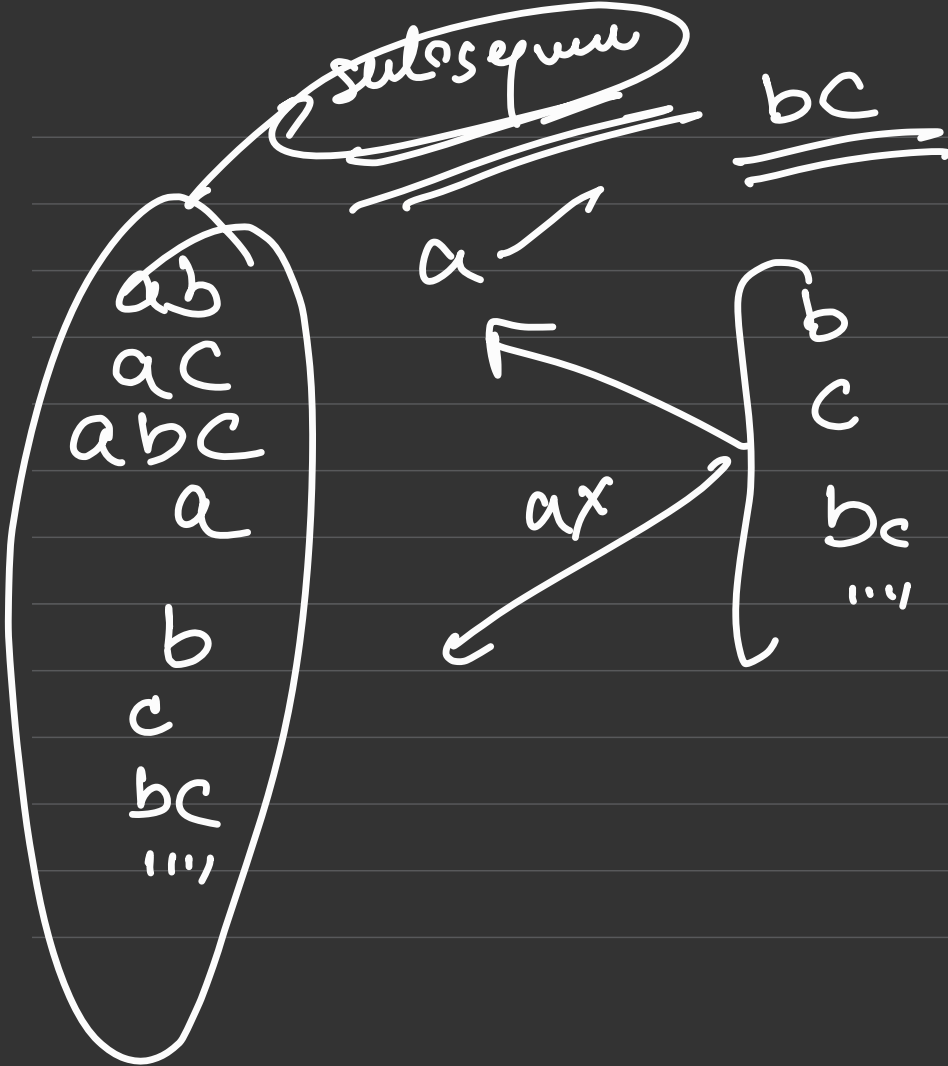


"abc"

$$f(\underline{i}) = a + (f(i+1)) , \quad f(i+1)$$

returns subsequences
of s[i, n]





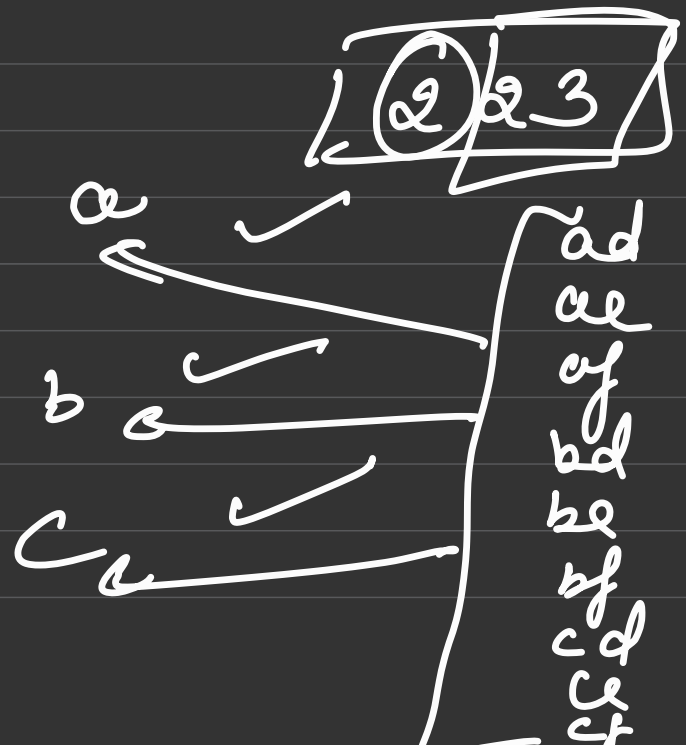
2 → "abc"
3 → "def"
:
:
:

num → 1001

← store
inp

²
↓
a b c

³
↓
d e f



Q. Given a value, print all the string
of parentheses being balanced parenthesis
of length n.

n=2

() ()

n=3

() () ()
(()) ()
() (())
((()))
(() ())

2n bytes string

open

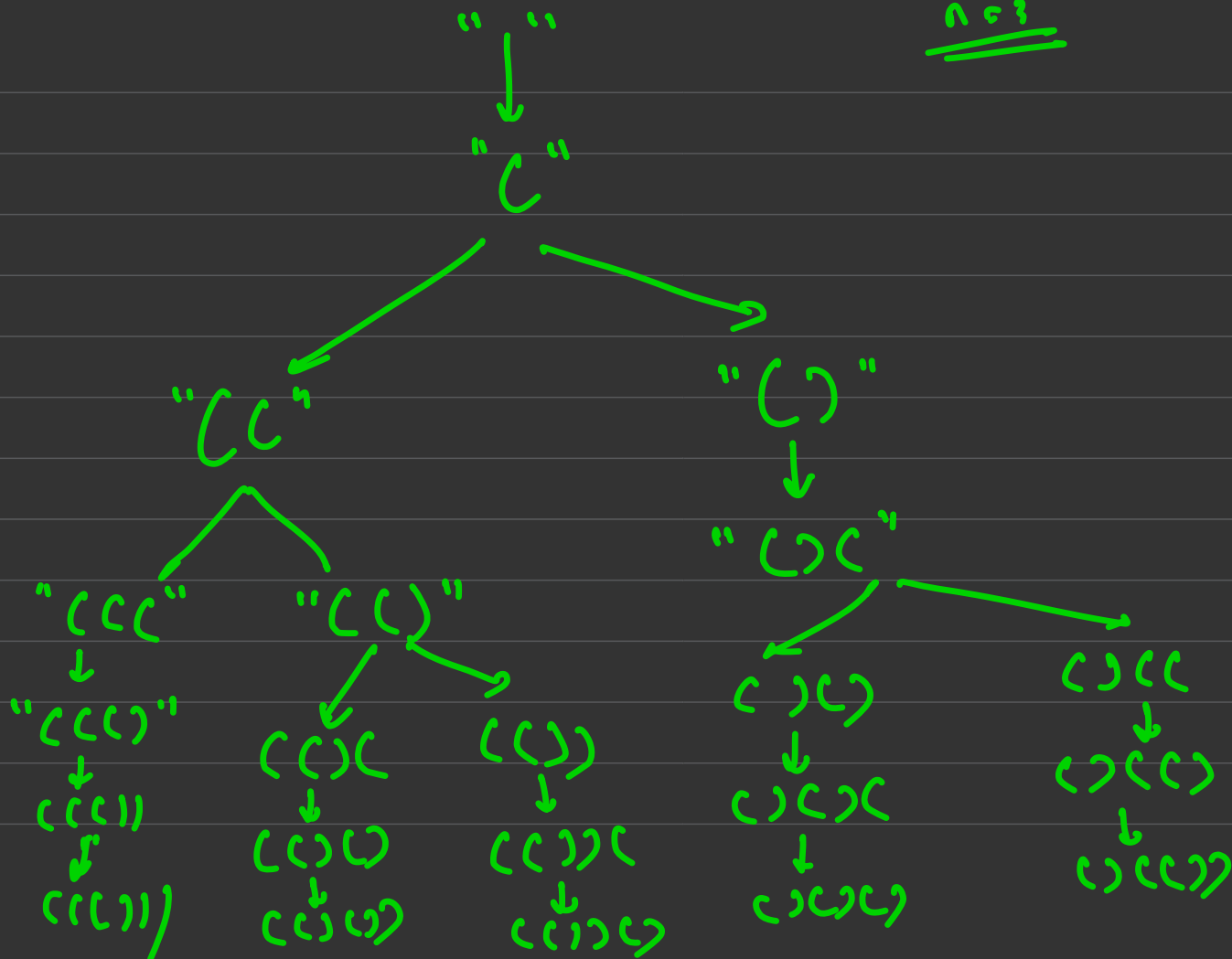
opening $\rightarrow \underline{\underline{n}}$

closing $\rightarrow \underline{\underline{(n)}}$

>>>

n=3

Ans



$f(o, c, n) \rightarrow$

$\left\{ \begin{array}{l} \text{if } (o < n) \\ \quad \hookrightarrow f(o+1, c, n) \checkmark \\ \text{if } (o > c) \\ \quad \hookrightarrow f(o, c+1, n) \checkmark \end{array} \right.$

$f(o, o, n)$ $\hookrightarrow \text{cuy}$