Longest Common Subsequence

→ given 2 strings → s_1 & s_2

↓

~~length of~~ ← longest common subsequence

Ex → S_1 → xyzzy → x, xz, xzy ....

S_2 → xbybzy → x, b, xby, xbz .....

xyzy

xyzy → 4

→ common subsequence → common char

↓

ordering also matters

if we go in decreasing index, we can maintain order

$lcs(abxz, dbyyz)$

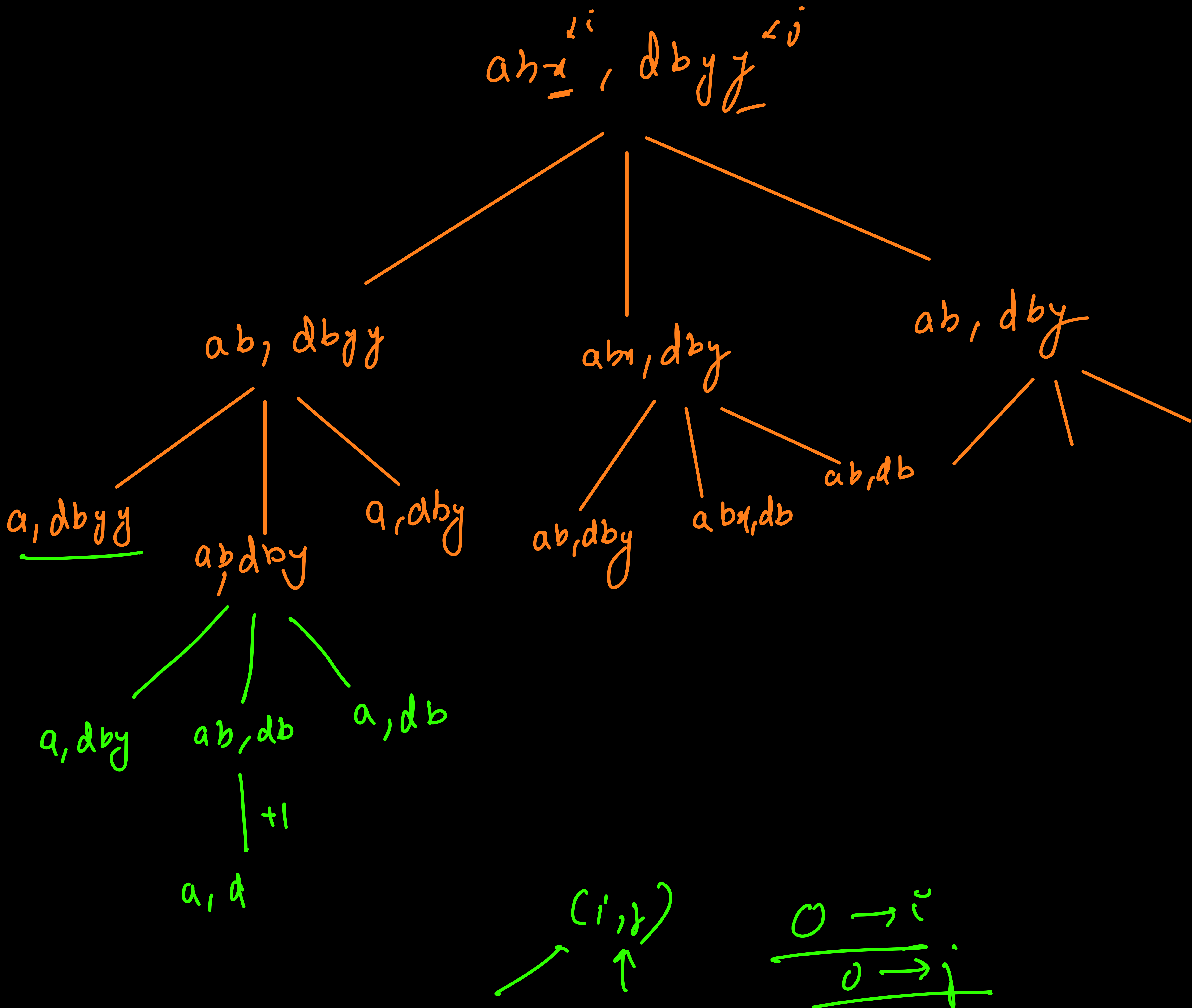
→ $S_1 = abxz$ $\quad i$

$S_2 = dbyyz$ $\quad j$

→ i^{th} & j^{th} char are equal

↓

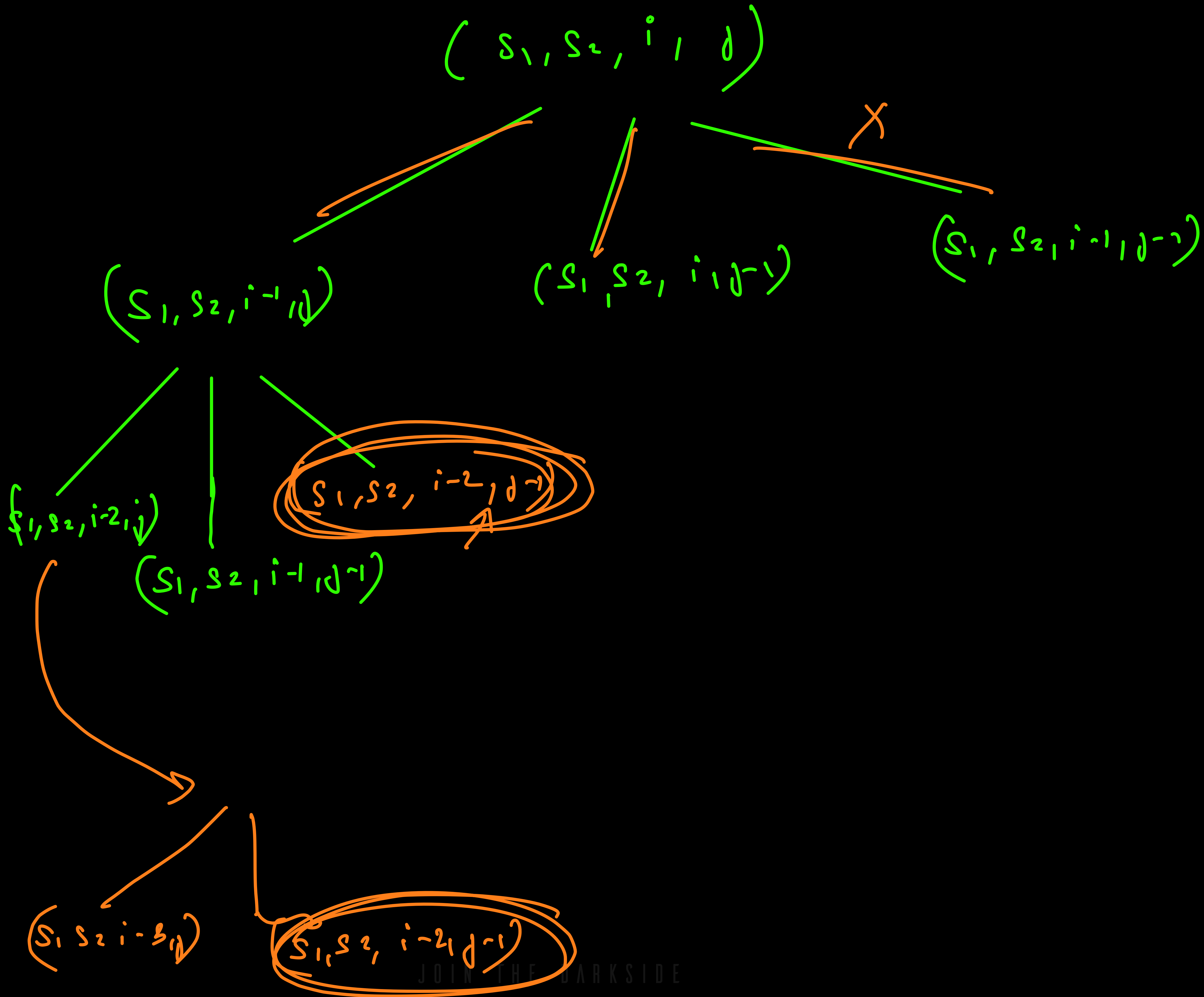
$1 + lcs(abx, dbyy)$

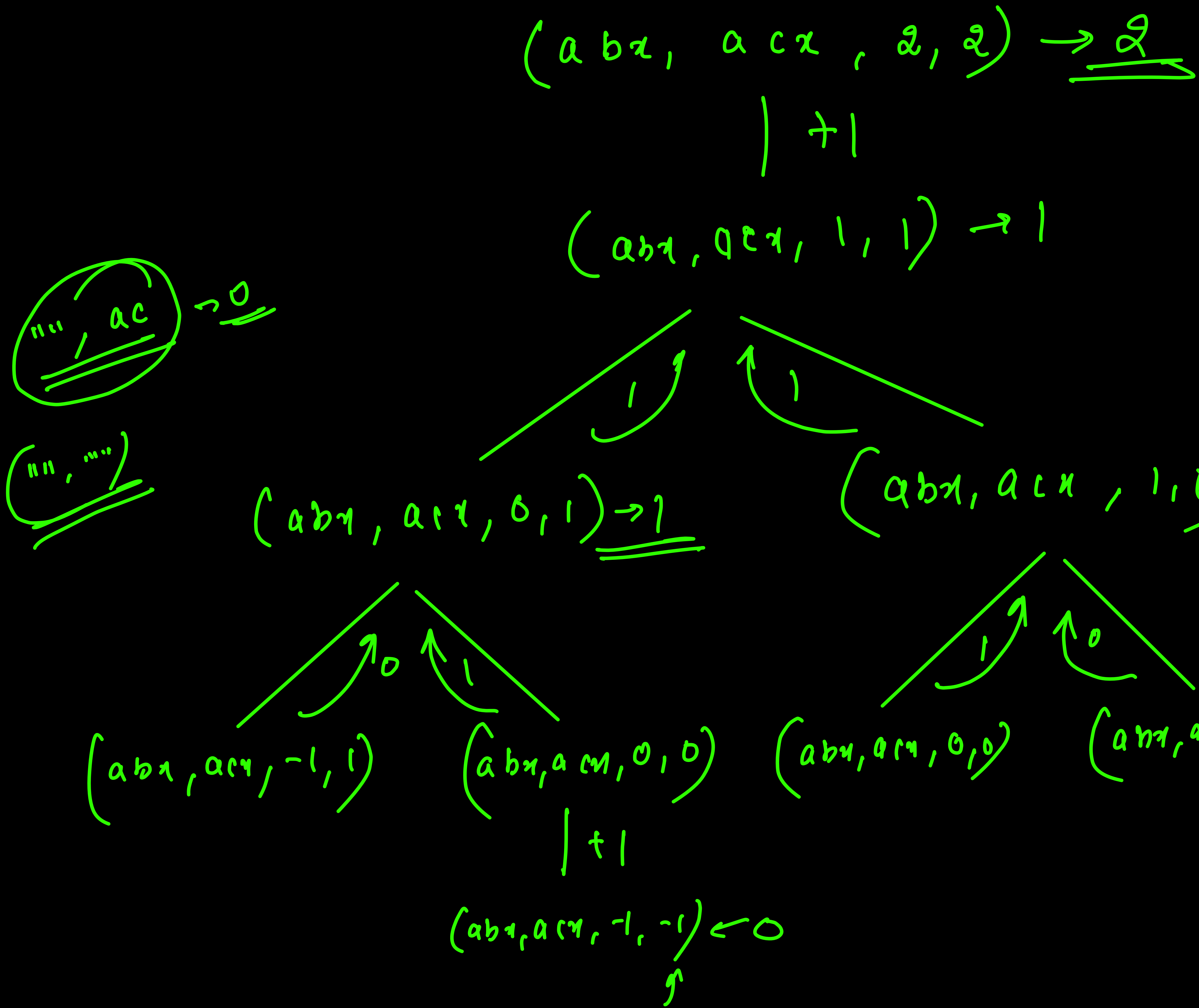

abx <i , dbyy <j
ab, dbyy
abx, dby
ab, dby
a, dbyy
ab, dby
a, dby
ab, dby
abx, db
ab, db
a, dby
ab, db
a, db
+1
a, a
(i, j)
0 → i
0 → j

$$f(s_1, s_2, i, j) = \begin{cases} 1 + f(s_1, s_2, i-1, j-1) & \text{if } s_1[i] == s_2[j] \\ \max\big(f(s_1, s_2, i-1, j),\ f(s_1, s_2, i, j-1) \big) & \text{else} \end{cases}$$

length of the lcs btw $s_1[0, i]$ and $s_2[0, j]$

max → 2d storage


(s1, s2, i, j)
(s1, s2, i-1, j)
(s1, s2, i, j-1)
(s1, s2, i-1, j-1)
(s1, s2, i-2, j)
(s1, s2, i-1, j-1)
(s1, s2, i-2, j-1)
(s1 s2 i-3, j)
(s1, s2, i-2, j-1)


(abx, acx, 2, 2) → 2
+1
(abx, acx, 1, 1) → 1
"", ac → 0
("", "")
1
1
(abx, acx, 0, 1) → 1
(abx, acx, 1, 0) → 1
0
1
1
0
(abx, acx, -1, 1)
(abx, acx, 0, 0)
(abx, acx, 0, 0)
(abx, acx, 1, -1)
+1
(abx, acx, -1, -1) ← 0

$$f(s_1, s_2, i, j) = \begin{cases} 1 + f(s_1, s_2, i+1, j+1) & \text{if } (s_1[i] == s_2[j]) \\ \max\begin{pmatrix} f(s_1, s_2, i+1, j) \\ f(s_1, s_2, i, j+1) \end{pmatrix} & \text{else} \end{cases}$$

$$\text{lcs}(s_1[i, n-1],\ s_2[j, m-1])$$

abbcb , bcbba

bb3

lcs((acbdcda), (adcdbca))

acbca

i j

acbdcda

a cbdcd a

lps

2

ch + palindrome + ch → palindrom

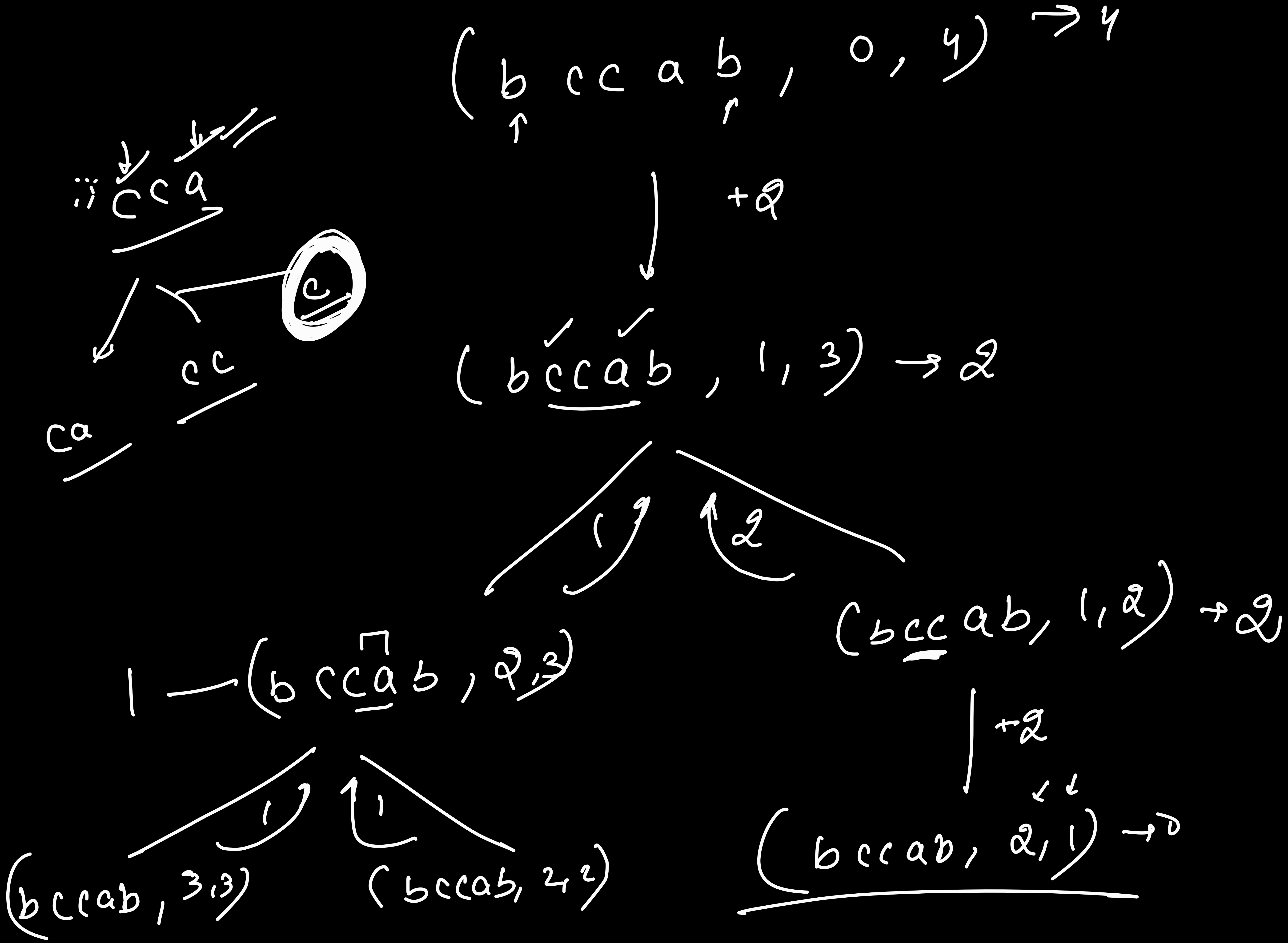
cabac

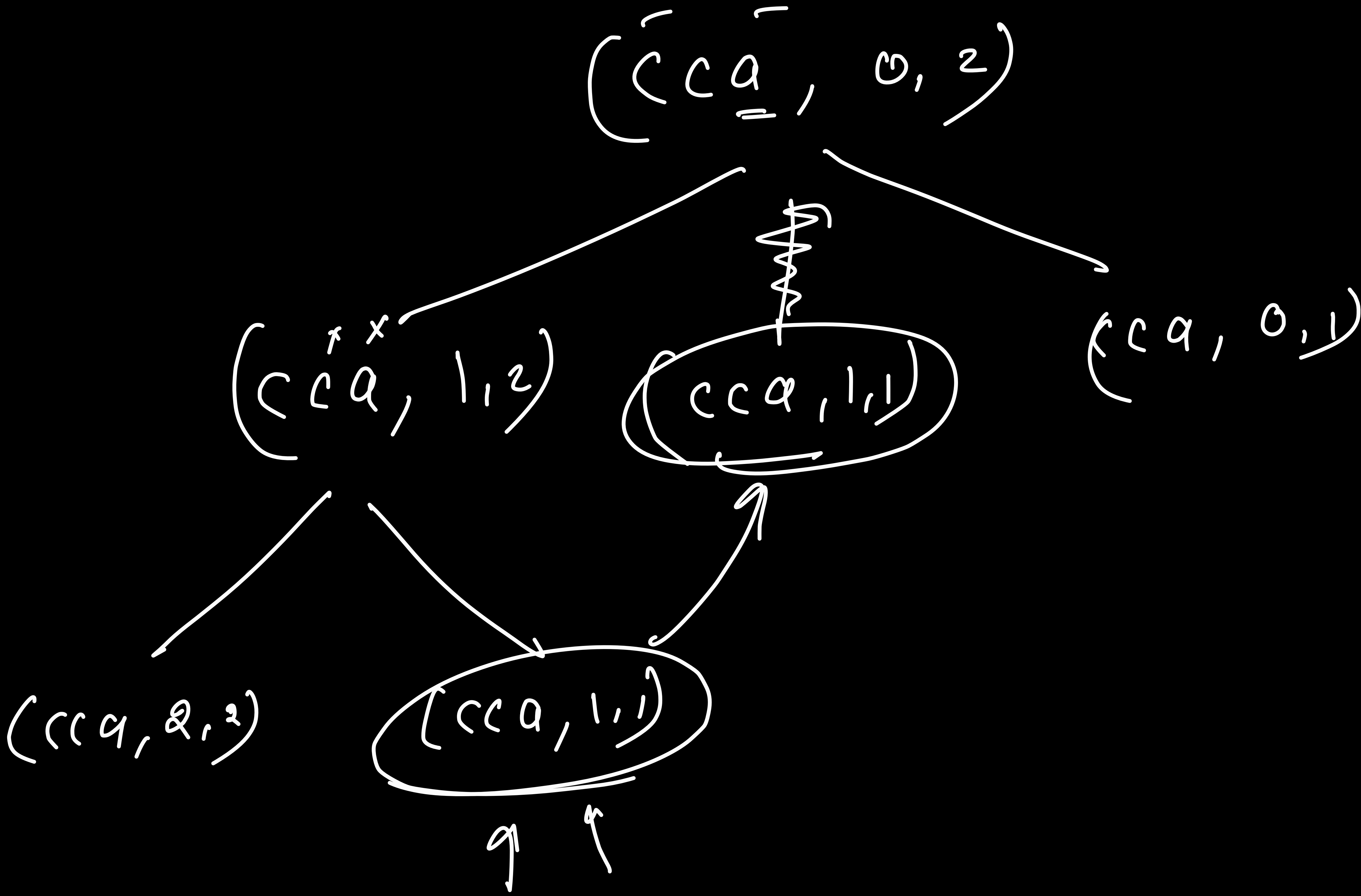
$f(s_1, i, j)$

length of lps

$\underline{\underline{s_1[i, j]}}$

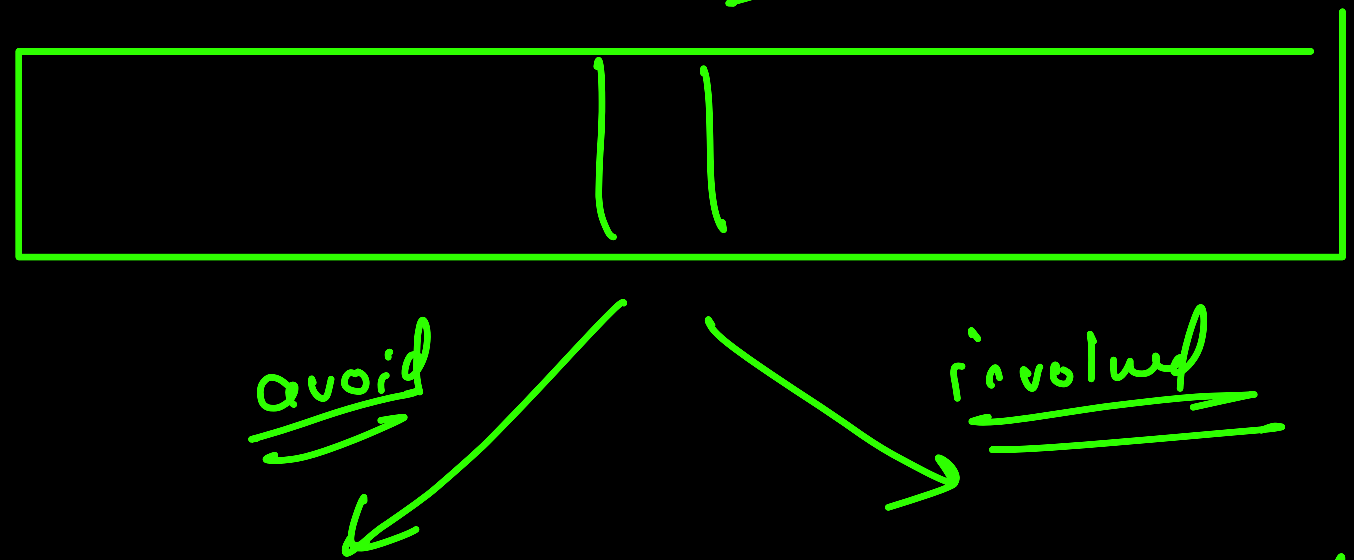
$$f(s_1, i, j) = \begin{cases} 2 + f(s_1, i+1, j-1) & \text{if } s_{1_i} == s_{1_j} \\ \max\begin{pmatrix} f(s_1, i+1, j) \\ f(s_1, i, j-1) \end{pmatrix} & \text{else} \end{cases}$$


(bccab, 0, 4) → 4
+2
(bccab, 1, 3) → 2
1
2
(bccab, 1, 2) → 2
+2
(bccab, 2, 1) → 0
1 — (bccab, 2, 3)
1
1
(bccab, 3, 3)
(bccab, 2, 2)
ii cca
ca
cc
c


(cca, 0, 2)
(cca, 1, 2)
(cca, 1, 1)
(cca, 0, 1)
(cca, 2, 2)
(cca, 1, 1)

prices k

k i



avoid

involved

if bought → sell

else → buy → if k is left

a txn is going on if you bought already but didn't

$f(i, on, k)$ → 3d dp

↓

max profit from i^{th} day to $(n-1)^{th}$ day with k txn left.

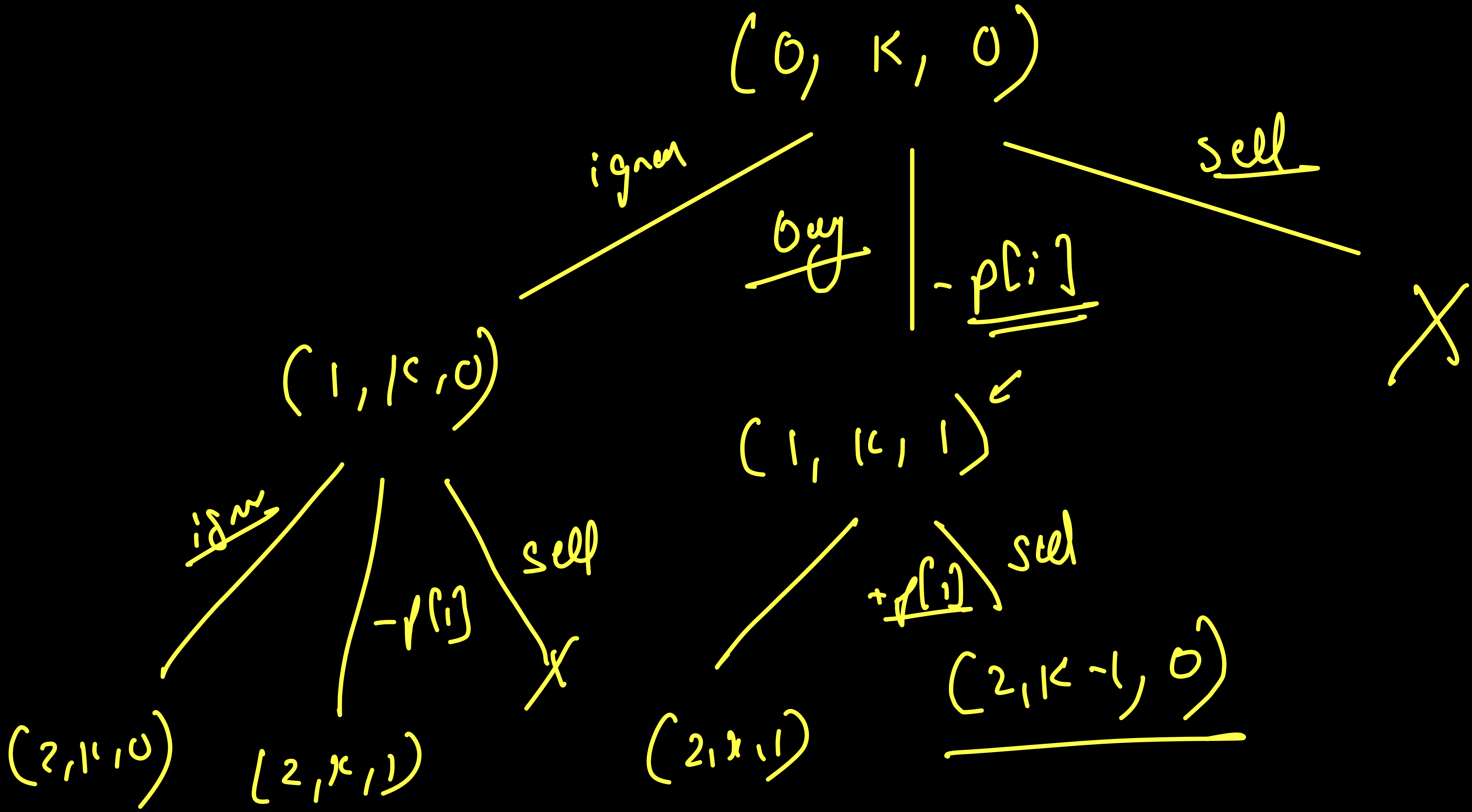
On → if a txn is going on or not

↓

$f(0, false, k)$

↳ ans

$$= \max \begin{cases} f(i+1, on, k) & \text{ignore} \\ -prices[i] + f(i+1, true, k) & \text{buy} \downarrow k>0 \text{ and } on == false \\ +prices[i] + f(i+1, False, k-1) & \text{sell} \downarrow on == true \end{cases}$$


(0, K, 0)
ignore
buy
- p[i]
sell
(1, K, 0)
(1, K, 1)
ign
-p[i]
sell
+p[i]
sell
(2, K, 0)
(2, K, 1)
(2, K, 1)
(2, K-1, 0)