↳ Longest Increasing Subsequence → take 2

eg: $A[\,] = [\,\boxed{3}\ 1\ 0\ 4\ 5\ 7 \dots\,]$

$0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \dots$

$x_3 \quad x_4 \quad x_5$

↳ [3 …]

⇒ find the LIS that starts with A[0].

$\boxed{1} + \underbrace{\max(x_3, x_4, x_5)}$

find the length of LIS in A[0 …. n-1] w/o any constraints.

$$\text{max} \Downarrow \text{final ans?} \begin{cases} f(0) \rightarrow \text{starts with } A(0) \\ f(1) \rightarrow \text{starts with } A(1) \\ \vdots \\ f(n-1) \rightarrow \text{starts with } A(n-1) \end{cases}$$

Let $f(i)$ denote the length of LIS in A[0…n-1] that starts with A[i] then

$$f(i) = 1 + \max_{j=i+1}^{n-1} (f(j)); \quad A[j] > A[i]$$

$$1 \qquad i == n-1$$

↳ find the length of LIS in A(0 …. n-1) w/o any constraints

↳ finds the length of LIS in A (0 … n-1) that starts with A(i)

↑ INT-MIN

$f(0) \rightarrow$ len. of LIS starts with $\boxed{\text{INT-MIN}}$

$\underbrace{\qquad}$

-1

$\underbrace{\qquad}_{\text{final ans}}$

- every rec problem is expressed using a single idx i

$0 \le i \le n-1$

∴ we'll need a 1D array of dimensions n to store results of each unique sub-problem.

$\Downarrow$

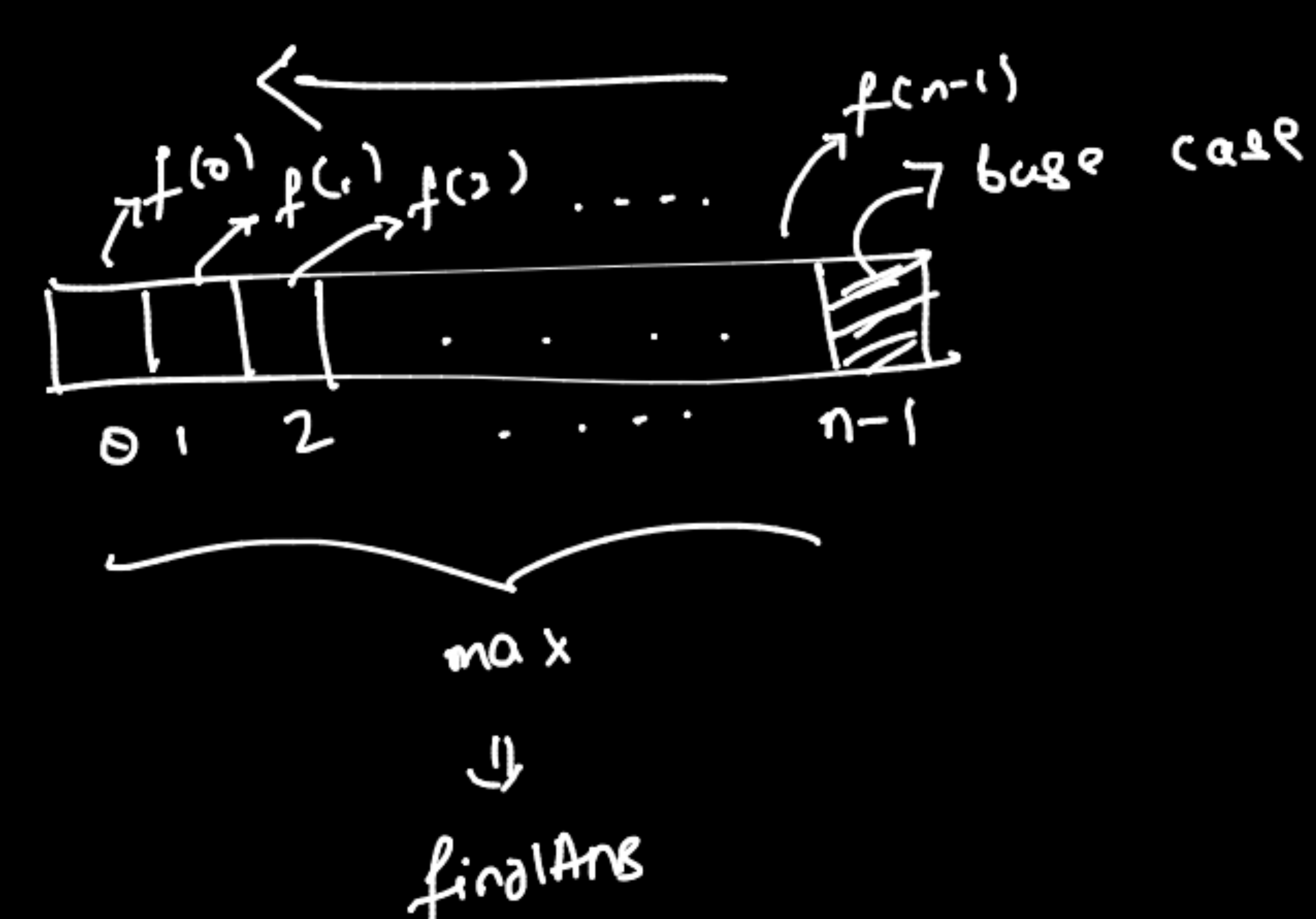
$f(i) \rightarrow$ length of LIS that ends with $A[i]$

$$f(i) = \begin{cases} 1 + \max\limits_{j=0}^{i-1} (f(j)) & ; \ A(i) > A(j) \\ 1 & ; \ i = 0 \end{cases}$$

this approach

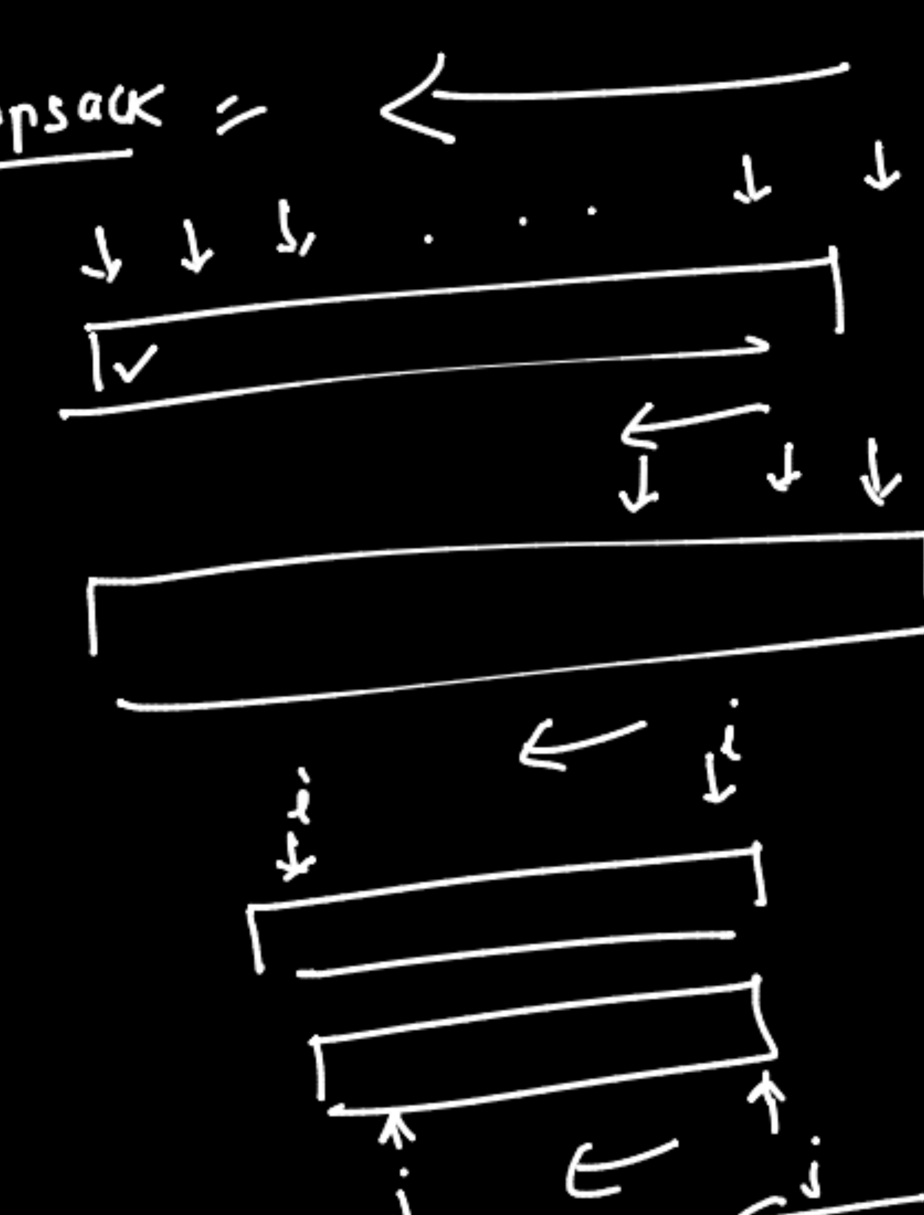
$$\left.\begin{matrix} f(0) \\ f(1) \\ \vdots \\ f(n-1) \end{matrix}\right\} \max \Rightarrow \text{INT-MAX}$$

dp[]



0/1 knapsack =

LCS =



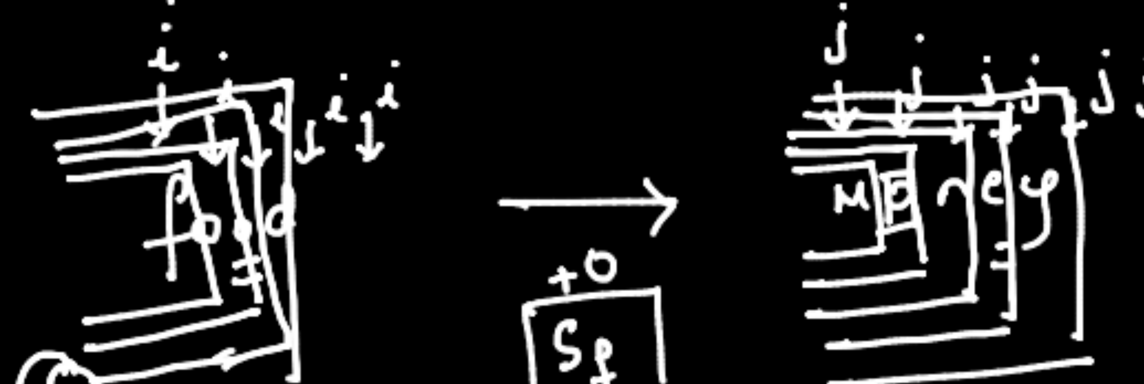
Edit Distance

s_1 and s_2

min edit oper.

- letter ins. +1 ✓
- letter del +1 ✓
- letter subs. +1 ✓

s_1 = "food" s_2 = "money"



food $\xrightarrow{S}$ mood $\xrightarrow{S_f}$ mood $\xrightarrow{I}$ monod $\xrightarrow{S}$ moned $\xrightarrow{S}$ money

cost of transf. / distance = 4

S	S_f	I	S	S

→ edit seq ⇒ 4

= food → money

d d d d i i i i i ⇒ 9

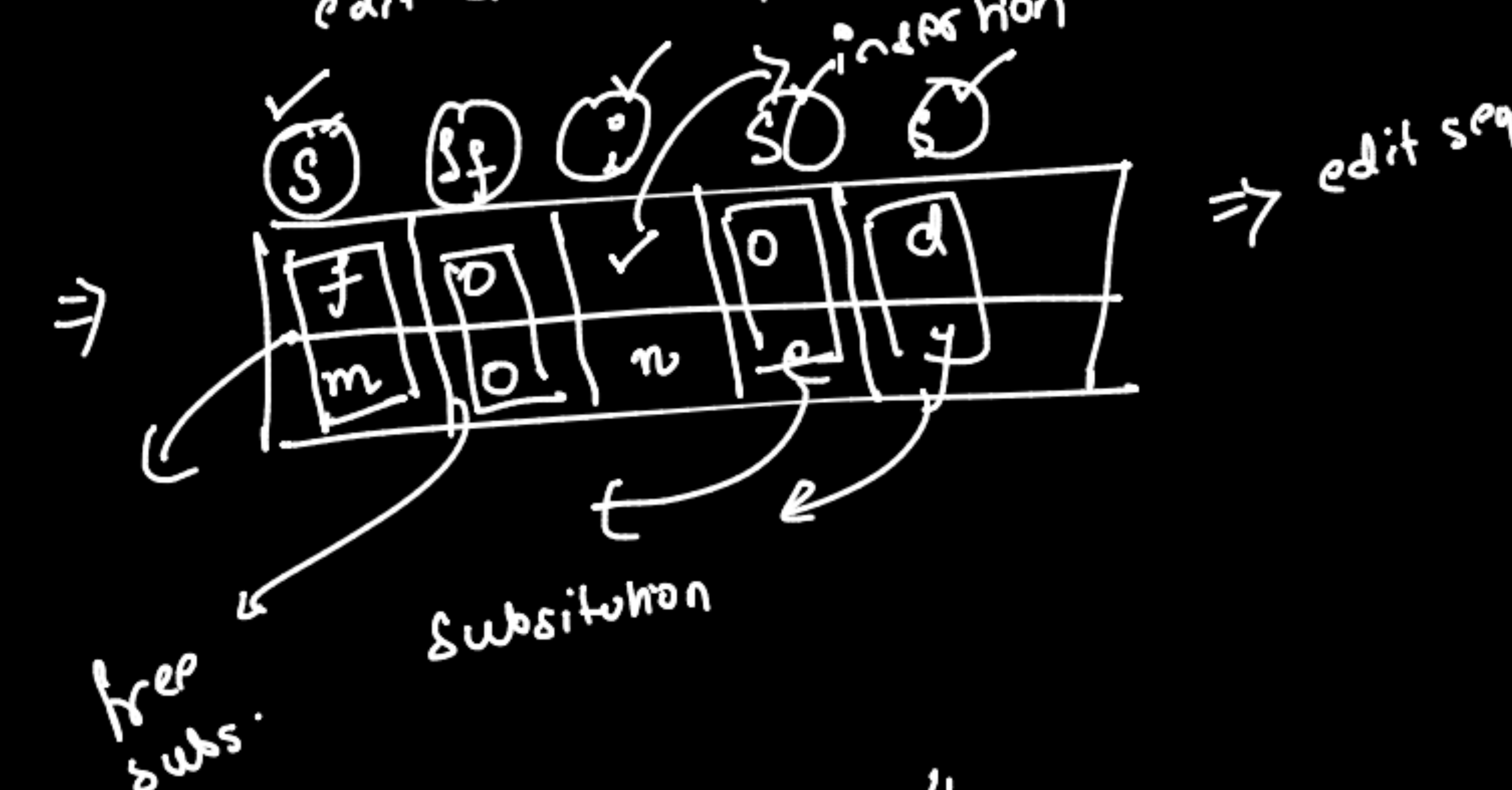
s d d d i i i i ⇒ 8

⋮

Can you transfer food into money in < 4 edits ?

food → money

edit dist = 4

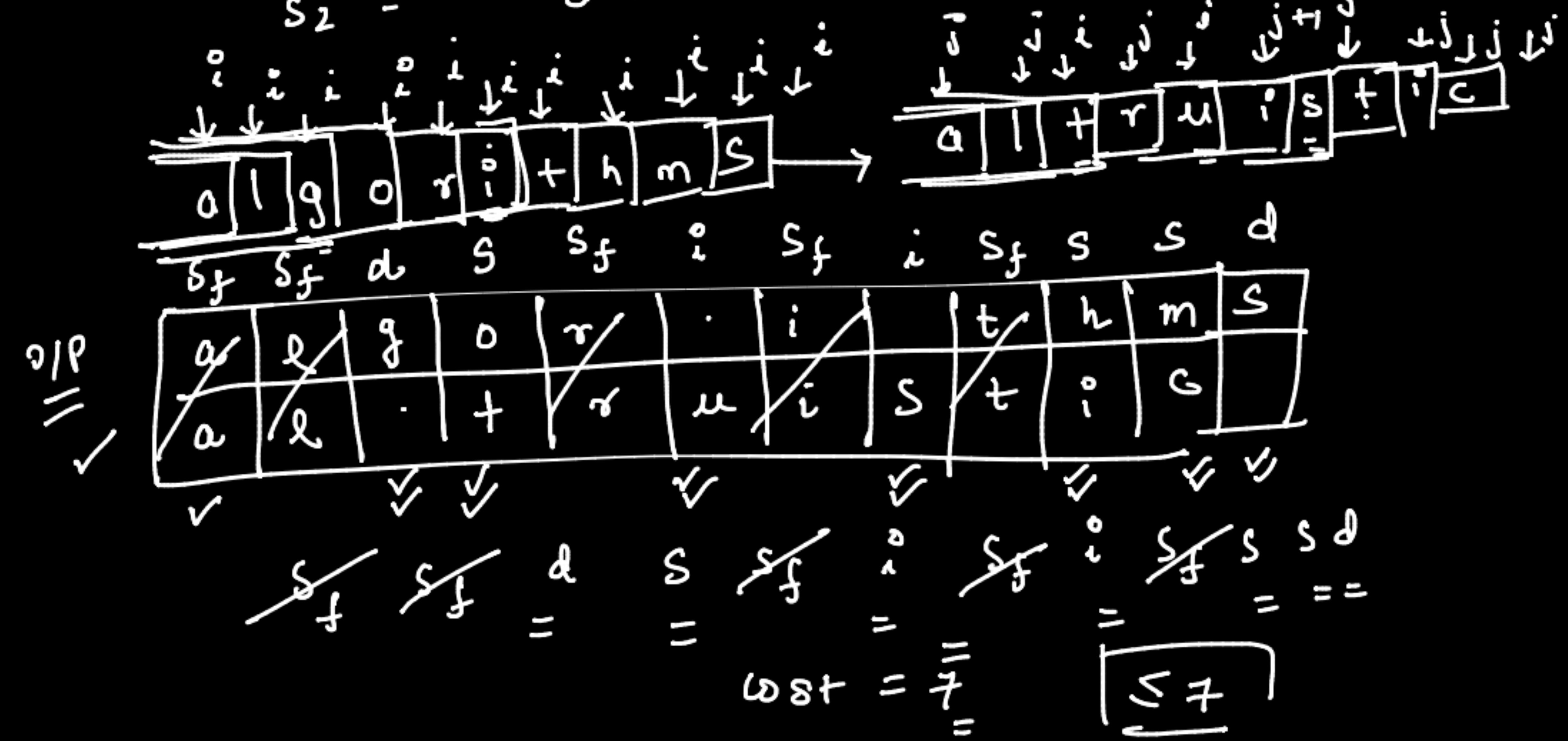


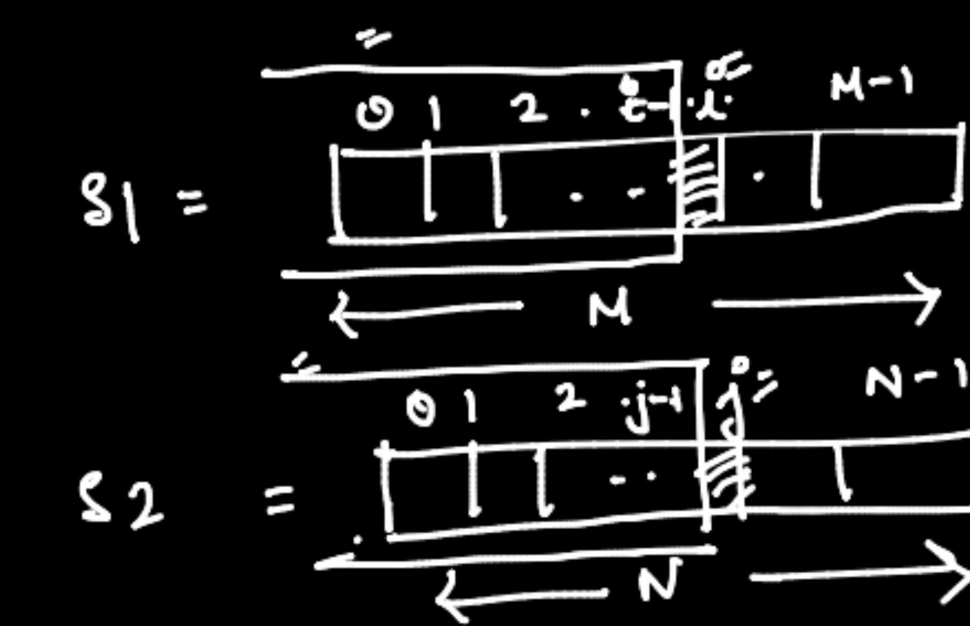
cost = 4

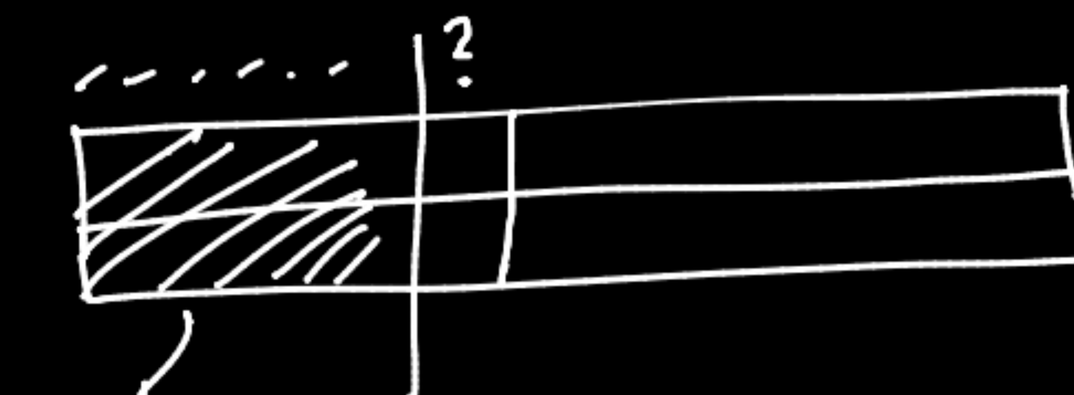
S ~~S_f~~ i S S

S_1 = "algorithms"

S_2 = "altruistic"







You've already made a decision for mago coin

Recursive Strategy

Solving edit dist prob. is equi. to coming up with the most optimal o/p alignment

$\downarrow$

shortest edit seq

suffix : $S_1[i \ldots m-1]$ to $S_2[j \ldots n-1]$?

$\min(1+X, 1+Y, Z+\underbrace{(S_1[i] \neq S_2[j])}_{1,0})$

(i, j)

insertion +1

aligned prefixes $S1[0 \ldots i-1]$ with $S_2[0 \ldots j]$

$\underbrace{(i, j+1)}$ to

$= S_1[i \ldots n-1]$ is X

$= S_2[j+1 \ldots n-1]$

deletion +1

aligned prefixes $S_1[0 \ldots i]$ with $S_2[0 \ldots j-1]$

$\underbrace{(i+1, j)}$

$S_1[i+1 \ldots n-1]$ to $S_2[j \ldots n-1]$ Y

substitution

$S_1[i] \mathrel{!=} S_2[j]$?

true +1

aligned prefix $S_1[0 \ldots i]$ with $S_2[0 \ldots j]$

$\underbrace{(i+1, j+1)}$

$S_1[i+1 \ldots m-1]$ to $S_2[j+1 \ldots n-1]$ Z

false (free)

aligned $S_1[0 \ldots i]$ with $S_2[0 \ldots j]$

$(i+1, j+1)$

Let $f(i, j)$ denote the min # edit operations req. to transform the suffix $s_1[i \ldots m-1]$ into suffix $s_2[j \ldots m-1]$

edit dist

$$f(i, j) = \min \begin{cases} 1 + f(i+1, j) & \text{// deletion} \\ 1 + f(i, j+1) & \text{// insertion} \\ f(i+1, j+1) + s_1(i) \neq s_2(j) & \text{// substitution} \end{cases}$$

$i = m$

$f(0,0)$ edit dist

→ min # edits reqd. to transform the suffix $s_1[0 \ldots m-1]$ into $s_2[0 \ldots n-1]$

0 : s_1 = empty, $i == m$ and s_2 = empty, $j == n$

$n - j$ insertions : s_1 = (empty), $i == m$ and s_2 = not empty, $j \,!= n$

$m - i$ deletions : (not empty) $i \,!= m$ and empty $j == n$

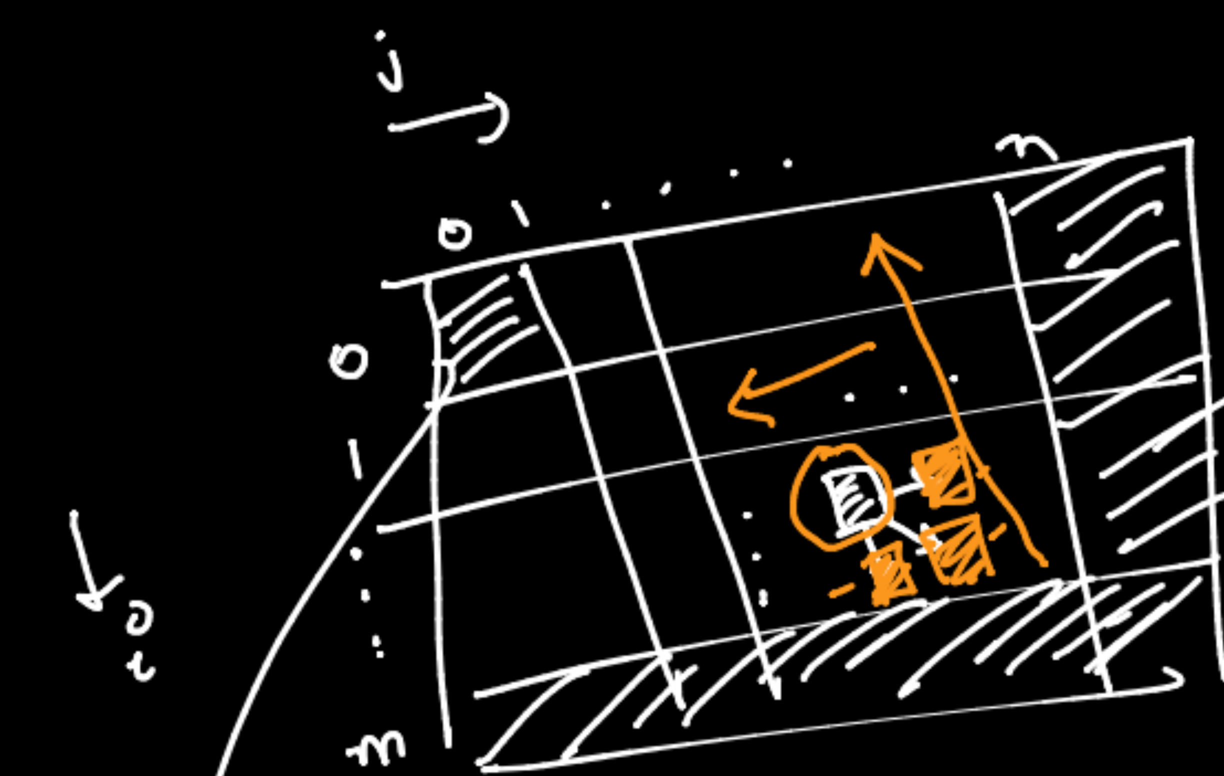
DP? ✓

Recursion tree → overlapping subproblem

Bottom up

(i, j) $0 \le i \le m$ and $0 \le j \le n$

2D $(m+1) \times (n+1)$



base case

row wise bottom to up, each row, right to left

final ans

$dp(i)(j) \to dp(i)(j+1)$, $dp(i+1)(j)$, $dp(i+1)(j+1)$

$$\boxed{\begin{aligned} S_1 &= \text{"\_"} + S_1 \\ S_2 &= \text{"\_"} + S_2 \end{aligned}}$$

filling col^n right to left

S_1 [x | 1 2 3 ← ... m] i

S_2 [x | 1 2 ← ... n] j

$$f(i,j) = \min \begin{cases} 1 + f(i, j-1) & \text{// ins} \\ 1 + f(i-1, j) & \text{// del} \\ f(i-1, j-1) + S_1(i) \;!=\; S_2(j) & \text{// sub} \end{cases}$$

⟸ follow this

denotes edit dist b/w prefix $S_1(0 \ldots i)$ and $S_2(0 \ldots j)$

$$\begin{cases} j & \boxed{i=0} \\ i & j=0 \\ 0 \text{ (not reqd)} & i=0 \text{ and } j=0 \end{cases}$$

automatically handled

$S_1 =$ [0 . . . . . . m-1] $i == m$ ✓

$S_2 =$ [0 . . . . . n-1] $j == n$ ✓

✓ $S_1 =$ [x | 1 2 . . . . . m] $i=0$

✓ $S_2 =$ [x | 1 2 . . . . . m] $j=0$

" " j