FOOD → MONEY

Operations:
- edit a letter
- delete a letter
- add a letter

FOOD
↓
MOOD → MONID
↓
MONED → MONEY

→ FOO_D
MONEY
Edit operation (3).

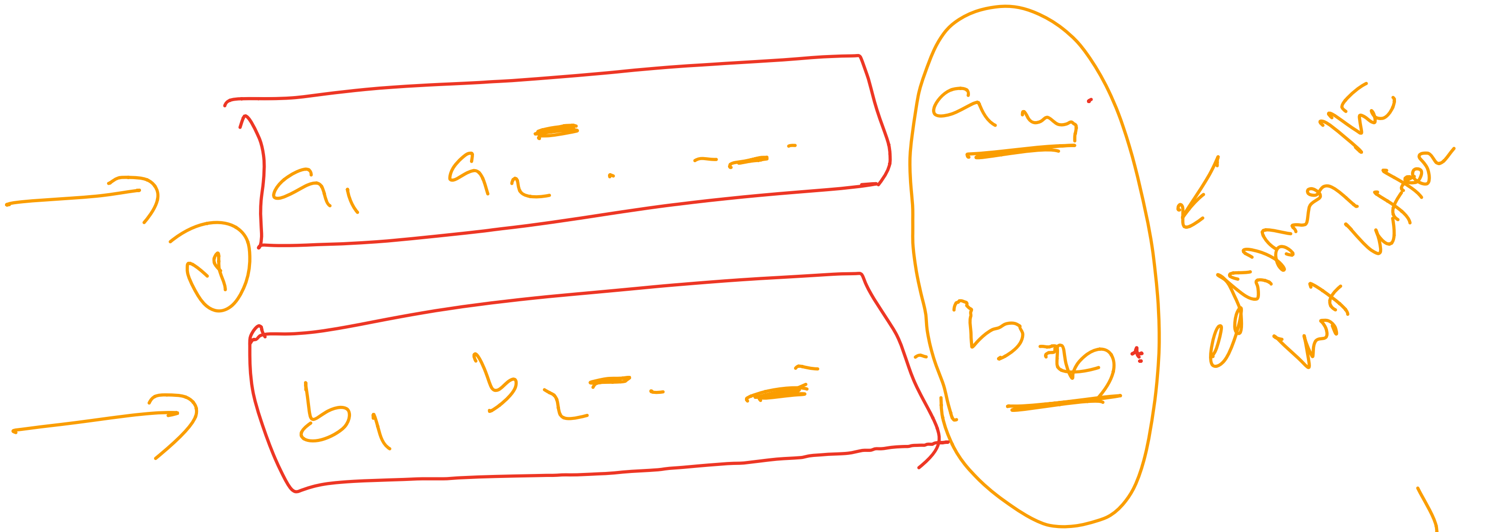
FOOD → MOOD → MOD
↓
MONEY ← MONE ← MON

FOOD
MONEY
Edit operation
(3)

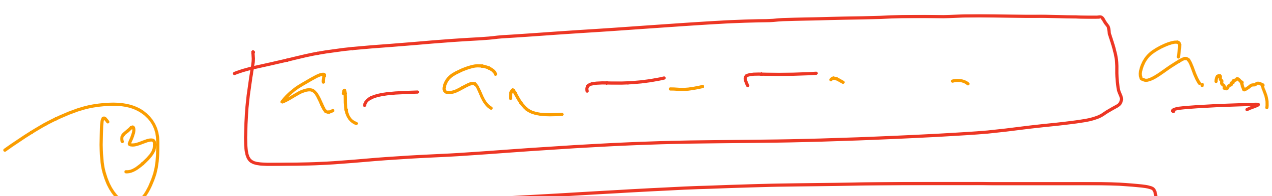
FOOD → MONEY

Input: Two sequences of letters $A[1 \ldots m]$ and $B[1 \ldots n]$

Output: The number of minimum edit operation required to convert $A[1..m]$ to $B[1..n]$



$a_m \neq b_n$



$b_1 - b_2 - \cdots - b_n$

Edit(i, j) = min number of operations required to convert A[1...i] to B[1...j].

Edit(m, n) — Our answer.

$$\text{Edit}(i,j) = \begin{cases} \min \begin{cases} (+)\ \text{Edit}(i-1, j-1) \\ (+)\ \text{Edit}(i, j-1) \\ (+)\ \text{Edit}(i-1, j) \end{cases} & \text{if } a_i \neq a_j \\ \min \begin{cases} \text{Edit}(i-1, j-1) \\ \text{Edit}(i, j-1) + 1 \\ \text{Edit}(i-1, j) + 1 \end{cases} & \text{if } a_i = b_j \\ j & i = 0 \\ i & j = 0 \end{cases}$$

$a_1 \cdots a_i$

$b_1 \cdots b_j$

b_j

a_i

b_j

$a_1 \ldots a_i -$

b_j

$a\, \underline{b}\, \underline{c}\, \underline{d} \longrightarrow \underline{b}\, \underline{c}\, \underline{d}$

delet a to bcd.

① Come-up with rec. formula.
[Exc. proof].

② $i \to 1$ to m
$i \to 1$ to n
$j \to 1$

$Edit(\vec{i}, j)$

③ DS. $EDIT[m+1][n+1]$

$EDIT[i, j] \leftarrow edit(i, j)$

```
EditDistance(A[1..m], B[1..n]):
    for j ← 0 to n
        Edit[0, j] ← j

    for i ← 1 to m
        Edit[i, 0] ← i
        for j ← 1 to n
            ins ← Edit[i, j − 1] + 1
            del ← Edit[i − 1, j] + 1
            if A[i] = B[j]
                rep ← Edit[i − 1, j − 1]
            else
                rep ← Edit[i − 1, j − 1] + 1
            Edit[i, j] ← min{ins, del, rep}

    return Edit[m, n]
```

base case

base case

Time complexity $T(n) = O(n \cdot m)$

Space complexity $S(n) = O(n \cdot m)$

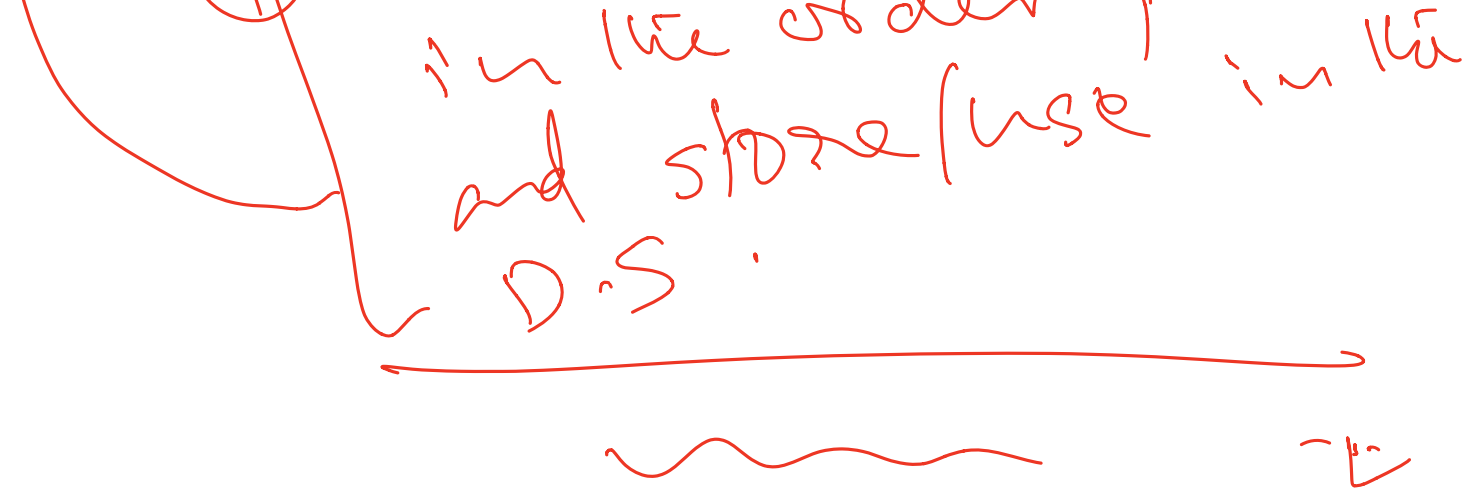
D.P

(1) Write recu. formula (proof).

(2) Find out evaluation order

(3) Find out suitable D.S

(4) Compute the values as prescribed


in the stack
and store/use in the
D.S.