

leetcode 72

Minimum Edit Distance:

Operations: insert/delete/replace

word 1 to word 2

1) horse to ros op: 3

h → r; ox; ex
replace del del

2) abcdef to azced

b → z, dx, f → d
replace del replace
del

	h	r	o	s
h	0	1	2	3
r	1	1	2	3
o	2	2	1	2
s	3	2	2	2
e	4	3	3	2
c	5	4	4	3

Structure to solve:



Recurrence Relation:

$$tab(i,j) = \begin{cases} tab(i-1,j-1); & \text{if } w1[i] == w2[j] \\ 1 + \min \begin{pmatrix} tab(i-1,j) \\ tab(i,j-1) \\ tab(i,j-1) \end{pmatrix} & \text{otherwise} \end{cases}$$

- row 0 & col 0 are base cases.

where value equals to character count.

starting at 0 going to len(word) respectively

- if both letter we looking at are same no extra work req^d ∴ fetch diagonal value because src[i] == des[j]

let's understand the other condition in recurrence

+1 would be for current edit (but which one)

depends on what minimum chooses.

- Say min chooses $tab(i-1,j-1) \Rightarrow$ replace src[i] by des[j]
- if min chooses $tab(i,j-1) \Rightarrow$ insertion of des[j]
- if min chooses $tab(i-1,j) \Rightarrow$ deletion of src[i]

Basic tracking: solⁿ building

if ~~diag~~ diag op matched value value
nothing was done, go there
otherwise
case: $1 + \min(\dots, \dots)$

value came from left:
des[j] was inserted
value came from top:
del src[i]

value came from diag:
replace src[i] to des[j]

Pseudo Code:
easy & intuitive
once idea is clear

// handle base cases
// row 0 & col 0

for i ← 1 to w1.len
for j ← 1 to w2.len

if same
 $tab(i,j) \leftarrow diag$

else
 $tab(i,j) \leftarrow$
 $1 + \min \begin{pmatrix} tab(i-1,j) \\ tab(i,j-1) \\ tab(i,j-1) \end{pmatrix}$

Same as recurrence