

Edit - Distance

Minimum steps to convert
Word 1 to Word 2

Recursive \Rightarrow DP

Base Case : (i) Word1 = "abc" \Rightarrow Delete all characters
Word2 = " " \Rightarrow return len(word1)

(ii) Word1 = " " \Rightarrow insert all elements
Word2 = "abc" of word 2
return len(word2)

Hypothesis : Operations \rightarrow Insert
 \rightarrow Delete } minimum
 \rightarrow Replace

Word1 = a b d
Word2 = a c d
i j
 \rightarrow not equal
 $\text{word1}[i] \neq \text{word2}[j]$
 \Rightarrow Calculate with which
operation steps will
be minimum.
 \rightarrow equal
 $\text{word1}[i] == \text{word2}[j]$
 \Rightarrow Don't need any
operation.
 $i++ , j++$

Word1 = a b d
Word2 = a c d
n m
 \rightarrow equal $(n-1, m-1)$

Word1 = a b c d
Word2 = a c d
n m
 $\xRightarrow{\text{Replace}}$ $(n-1, m-1)$
 \rightarrow both become equal by replace
so decrease both pointers.

Word1 = a b d
Word2 = a c d
n m
 $\xRightarrow{\text{Delete}}$ $(n-1, m)$

As 'd' is already checked so check for 'c' in word1 so
 $(n-1)$

Word 1 = a ⁿ b d \Rightarrow Insert a ⁿ b c d
 Word 2 = a c d a c d
m

As for 'c' of word 2 we added 'c' in word 1
 so c is checked. so now we have to check
 for remaining i.e. left side elements of word 2

$\therefore (m-1)$ $(n, m-1)$

def solve(w1, w2, n, m):

if $n == 0$ or $m == 0$: return m or n

elif $w1[n-1] == w2[m-1]$:

return solve(w1, w2, n-1, m-1)

else:

return 1 + min(

solve(w1, w2, n-1, m-1) # replace

solve(w1, w2, n-1, m) # Delete

solve(w1, w2, n, m-1) # Insert
)

Tabulation

		$\xleftarrow{\quad w2 \quad}$			
			a	c	d
		0	1	2	3
\uparrow w1 \downarrow	a	0	0	1	(2)
	b	1	1		
	d	2	2		
		3	(3)		

if w1 is empty and w2 is at length 2 i.e. $w2 = ac$ minimum 2 steps needed

if $w2 = ""$ and $w1 = abd$ minimum 3 steps that is why this cell contain 3

Top-down

$dp = [[0] * (m+1) \text{ for } i \text{ in range } (n+1)]$

for j in range $(m+1)$: # Base case

$dp[0][j] = j$

0th row

$w1 = ""$

for i in range $(n+1)$: # Base case

$dp[i][0] = i$

0th column

$w2 = ""$

for i in range $(1, n+1)$:

for j in range $(1, m+1)$:

if $w1[i-1] == w2[j-1]$

$dp[i][j] = dp[i-1][j-1]$

else:

$dp[i][j] = 1 + \min($

$dp[i-1][j-1]$ # Replace

$dp[i-1][j]$ # Delete

$dp[i][j-1]$ # Insert

)

return $dp[-1][-1]$