Minimum edit distance algorithm II

To populate the following table:

Source: play → Target: stay
Cost: insert: 1, delete: 1, replace: 2

p → s

$D[i, j] =$

$$min \begin{cases} D[i - 1, j] + del\_cost \\ D[i, j - 1] + ins\_cost \\ D[i - 1, j - 1] + \begin{cases} rep\_cost; & if\ src[i] \neq tar[j] \\ 0; & if\ src[i] = tar[j] \end{cases} \end{cases}$$

		0 #	1 s	2 t	3 a	4 y
0	#	0	1	2	3	4
1	p	1	2			
2	l	2				
3	a	3				
4	y	4				

There are three equations:

- **D[i,j] = D[i-1, j] + del_cost:** this indicates you want to populate the current cell (i,j) by using the cost in the cell found directly above.

- **D[i,j] = D[i, j-1] + ins_cost:** this indicates you want to populate the current cell (i,j) by using the cost in the cell found directly to its left.

- **D[i,j] = D[i-1, j-1] + rep_cost:** the rep cost can be 2 or 0 depending if you are going to actually replace it or not.

At every time step you check the three possible paths where you can come from and you select the least expensive one. Once you are done, you get the following:

Source: play → Target: stay
Cost: insert: 1, delete: 1, replace: 2

		0 #	1 s	2 t	3 a	4 y
0	#	0	1	2	3	4
1	p	1	2	3	4	5
2	l	2	3	4	5	6
3	a	3	4	5	4	5
4	y	4	5	6	5	4

Mark as completed

👍 Like   👎 Dislike   🚩 Report an issue