Wagner-Fischer algorithm for finding string edit paths

"Mondays" ⟶ "Wednesday"

	W	e	d	n	e	s	d	a	y	ø
M	0	1	2	3	4	5	6	7	8	9
o	1	1	2	3	4	5	6	7	8	9
n	2	2	2	3	4	5	6	7	8	9
d	3	3	3	3	3	4	5	6	7	8
a	4	4	4	3	4	4	5	5	6	7
y	5	5	5	4	4	5	5	6	5	6
s	6	6	6	5	5	5	6	6	6	5
ø	7	7	7	6	6	6	5	6	7	6

M/W, o/e, +d, n, +e, +s, d, a, y, -s

First form a matrix A whose rows correspond to letters in the source S string (length m) and columns to letters in the T target (length n).

Initialise an $m \times n$ matrix as

$$A = \begin{pmatrix} 0 & 1 & \dots & n \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ m & 0 & \dots & 0 \end{pmatrix}$$

and apply the rule

$$A_{i,j} = \min\{A_{(i-1),j} + 1, A_{i,(j-1)} + 1, A_{(i-1),(j-1)} + s\}$$
$$s = \begin{cases} 0 \text{ if } S_i = T_j \\ 1 \text{ otherwise} \end{cases}$$

in order of increasing $i, j > 1$.

Then, form a path through the entries of A, starting from the $(m+1, n+1)$ position (bottom right), moving one step to the neighboring cell of minimum value until the $(1, 1)$ position (top left) is reached.

Left steps ← correspond to insertions, and upward steps ↑ correspond to deletions. Diagonal steps ↖ correspond to accepting the current character when the cell values are equal, or substituting characters otherwise.

Finally, moving along this path in the ↘ direction, you can read off the character operations which map the source string to the target.

More examples!

Generated with the above algorithm in this document's Typst source code.

"For Wednesday" ⟶ "From Monday"

	F	r	o	m		M	o	n	d	a	y	ø
F	0	1	2	3	4	5	6	7	8	9	10	11
o	1	0	1	2	3	4	5	6	7	8	9	10
r	2	1	1	1	2	3	4	5	6	7	8	9
	3	2	1	2	2	3	4	5	6	7	8	9
W	4	3	2	2	3	2	3	4	5	6	7	8
e	5	4	3	3	3	3	3	4	5	6	7	8
d	6	5	4	4	4	4	4	4	5	6	7	8
n	7	6	5	5	5	5	5	5	5	5	6	7
e	8	7	6	6	6	6	6	6	5	6	6	7
s	9	8	7	7	7	7	7	7	6	6	7	7
d	10	9	8	8	8	8	8	8	7	7	7	8
a	11	10	9	9	9	9	9	9	8	7	8	8
y	12	11	10	10	10	10	10	10	9	8	7	8
ø	13	12	11	11	11	11	11	11	10	9	8	7

F, +r, o, r/m, , W/M, e/o, -d, n, -e, -s, d, a, y

"Typst" ⟶ "Typeset"

	T	y	p	e	s	e	t	ø
T	0	1	2	3	4	5	6	7
y	1	0	1	2	3	4	5	6
p	2	1	0	1	2	3	4	5
s	3	2	1	0	1	2	3	4
t	4	3	2	1	1	1	2	3
ø	5	4	3	2	2	2	2	2

T, y, p, +e, s, +e, t

"ABC@YZ" ⟶ "AB@XYZ"

	A	B	@	X	Y	Z	ø
A	0	1	2	3	4	5	6
B	1	0	1	2	3	4	5
C	2	1	0	1	2	3	4
@	3	2	1	1	2	3	4
Y	4	3	2	1	2	3	4
Z	5	4	3	2	2	2	3
ø	6	5	4	3	3	3	2

A, B, -C, @, +X, Y, Z