Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1						
a	2						
n	3						
a	4						
d	5						
a	6						

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1					
a	2						
n	3						
a	4						
d	5						
a	6						

$b \neq c$

$d[1,1] = \min(d[0,0]+1, d[0,1]+1, d[1,0]+1)$

$d[1,1] = \min(0+1, 1+1, 1+1)$

$d[1,1] = 1$

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2				
a	2						
n	3						
a	4						
d	5						
a	6						

$a \neq c$

$d[1,2] = \min(d[0,1]+1, d[0,2]+1, d[1,1]+1)$

$d[1,2] = \min(1+1, 2+1, 1+1)$

$d[1,2] = 2$

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2						
n	3						
a	4						
d	5						
a	6						

n ≠ c
a ≠ c
n ≠ c
a ≠ c

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2					
n	3						
a	4						
d	5						
a	6						

b ≠ a

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2	1				
n	3						
a	4						
d	5						
a	6						

a==a
d[2,2] = min(d[1,1], d[1,2]+1, d[2,1]+1)
d[2,2]= d[1,1]
d[2,2]=1

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2	1	2	3		
n	3						
a	4						
d	5						
a	6						

a==a
d[2,4] = min(d[1,3], d[1,4]+1,d[2,3]+1)
d[2,4] = 3

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2	1	2	3	4	5
n	3	3	2	1	2	3	4
a	4	4	3	2	1	2	3
d	5	5	4	3	2	2	3
a	6	6	5	4	3	3	2

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2	1	2	3	4	5
n	3	3	2	1	2	3	4
a	4	4	3	2	1	2	3
d	5	5	4	3	2	2	3
a	6	6	5	4	3	3	2

Keep(a)

banana

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2	1	2	3	4	5
n	3	3	2	1	2	3	4
a	4	4	3	2	1	2	3
d	5	5	4	3	2	2	3
a	6	6	5	4	3	3	2

Keep(a)	banana
Substitute(n,d)	banada

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2	1	2	3	4	5
n	3	3	2	1	2	3	4
a	4	4	3	2	1	2	3
d	5	5	4	3	2	2	3
a	6	6	5	4	3	3	2

Keep(a)	banana
Substitute(n,d)	banada
Keep(a)	banada

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2	1	2	3	4	5
n	3	3	2	1	2	3	4
a	4	4	3	2	1	2	3
d	5	5	4	3	2	2	3
a	6	6	5	4	3	3	2

Keep(a)	banana
Substitute(n,d)	banada
Keep(a)	banada
Keep(n)	banada
Keep(a)	banada
Substitute(b,c)	canada

↑ = insert

← = delete

↖ = keep, if same values

↖ = substitue, if different values

Levenshtein Distance

	-	b	a	n	a	n	a	s
-	0	1	2	3	4	5	6	7
c	1	1	2	3	4	5	6	7
a	2	2	1	2	3	4	5	6
n	3	3	2	1	2	3	4	5
a	4	4	3	2	1	2	3	4
d	5	5	4	3	2	2	3	4
a	6	6	5	4	3	3	2	3

Delete(s)	banana~~s~~
Keep(a)	banana
Substitute(n,d)	banada
Keep(a)	banada
Keep(n)	banada
Keep(a)	banada
Substitute(b,c)	canada

↑ = insert

← = delete

↖ = keep, if same values

↖ = substitue, if different values

Levenshtein Distance

	-	b	a	n	a	n	a	s
-	0	1	2	3	4	5	6	7
c	1	1	2	3	4	5	6	7
a	2	2	1	2	3	4	5	6
n	3	3	2	1	2	3	4	5
a	4	4	3	2	1	2	3	4
d	5	5	4	3	2	2	3	4
i	6	6	5	4	3	3	3	4
a	7	7	6	5	4	4	3	4
n	8	8	7	6	5	4	4	4

Substitute(s,n)	bananan
Keep(a)	bananan
Insert(i)	bananian
Substitute(n,d)	banadian
Keep(a)	banadian
Keep(n)	banadian
Keep(a)	banadian
Substitute(b,c)	canadian

↑ = insert

← = delete

↖ = keep, if same values

↖ = substitue, if different values

Levenshtein Distance

	-	b	a	n	a	n	a
-	0	1	2	3	4	5	6
c	1	1	2	3	4	5	6
a	2	2	1	2	3	4	5
n	3	3	2	1	2	3	4
a	4	4	3	2	1	2	3
d	5	5	4	3	2	2	3
a	6	6	5	4	3	3	2

Keep(a)	banana
Substitute(n,d)	banada
Keep(a)	banada
Keep(n)	banada
Keep(a)	banada
Substitute(b,c)	canada

↑ = insert

← = delete

↖ = keep, if same values

↖ = substitue, if different values

Levenshtein Distance

	-	p	a	r	a	g	u	a	y
-	0	1	2	3	4	5	6	7	8
p	1								
a	2								
p	3								
a	4								
y	5								
a	6								

Levenshtein Distance

	-	p	a	r	a	g	u	a	y
-	0	1	2	3	4	5	6	7	8
p	1	0	1	2	3	4	5	6	7
a	2	1	0	1	2	3	4	5	6
p	3	2	1	1	2	3	4	5	6
a	4	3	2	2	1	2	3	4	5
y	5	4	3	3	2	2	3	4	4
a	6	5	4	4	3	3	3	3	4