Bahria University
Department of Computer Science

Adv. Natural Language Processing

Lecture 4

Instructor: Dr. Muhammad Asfand-e-yar

Previous Lecture

Regular Expressions

Text Normalization

Today's Lecture

Minimum Edit Distance

Minimum Edit Distance

Definition of Minimum Edit Distance

How similar are two strings?

Spell correction

The user typed “graffe”

Which is closest?

- graf
- graft
- grail
- giraffe

Computational Biology

Align two sequences of nucleotides

AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC

Resulting alignment:

-AGG**CTATCAC**CT**GACC**T**C**CA**GG**C**CGA**-- **TGCCC** ---
T**AG**- **CTATCAC** -- **GACC**G**C**-- **GG**T**CGA**TT**TGCCC**GAC

The method is used for Machine Translation, Information Extraction, Speech Recognition

Minimum Edit Distance

The Minimum Edit Distance between two strings

Minimum Edit Distance is the minimum number of editing operations

1. Insertion
2. Deletion
3. Substitution

Which is needed to transform one string into the other

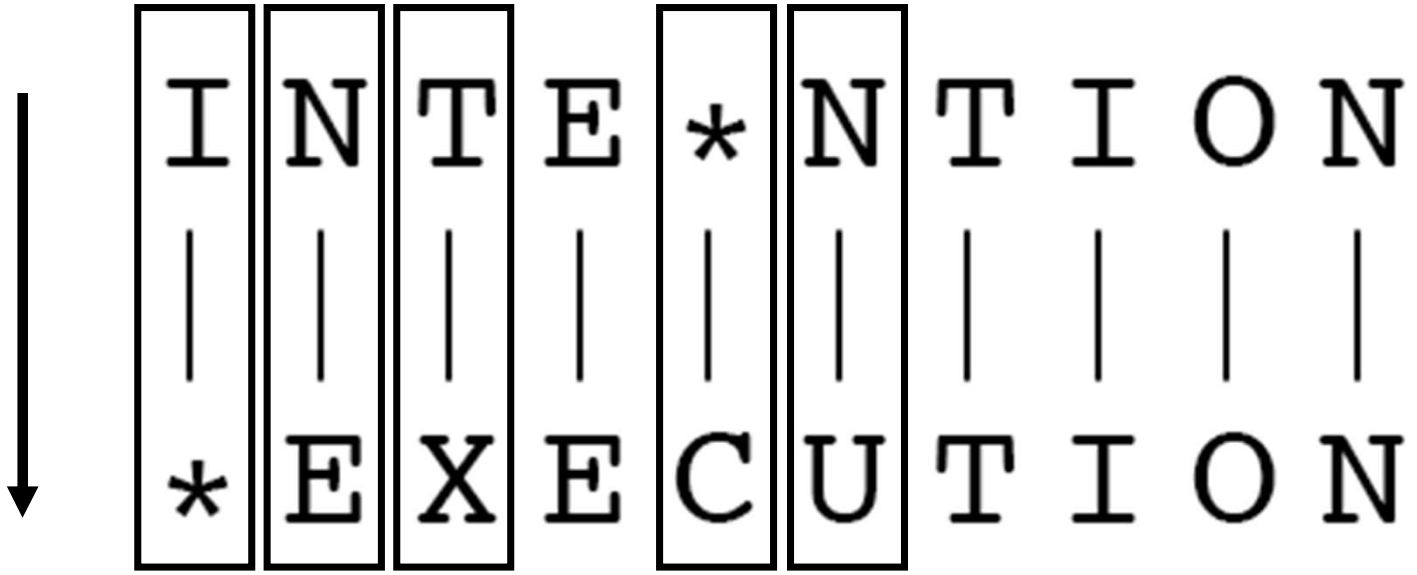
Minimum Edit Distance

Two strings and their **alignment**:

```
I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N
```

Minimum Edit Distance

Two strings and their **alignment**:



Minimum Edit Distance

```
I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N

d s s   i s
```

If each operation has cost of 1

Distance between these is 5

If substitutions cost 2 (Levenshtein)

Distance between them is 8

Alignment in Computational Biology

Given a sequence of bases

AGGCTATCACCTGACCTCCAGGCCGATGCCC

TAGCTATCACGACCGCGGTCGATTTGCCCGAC

An alignment:

-AGG**CTATCAC**C**T****GACC**T**C**C**A****GG**C**CGA--TGCCC**---

T**AG-CTATCAC--GACC**G**C--GG**T**CGA**T**T****TGCCC**GAC

Given two sequences, align each letter to a letter or gap

Examples: Minimum Edit Distance in NLP

Evaluating Machine Translation and speech recognition

R	Spokesman	confirms		senior	government	adviser	was	shot	
H	Spokesman	said	the	senior		adviser	was	shot	dead
		S	I		D				I

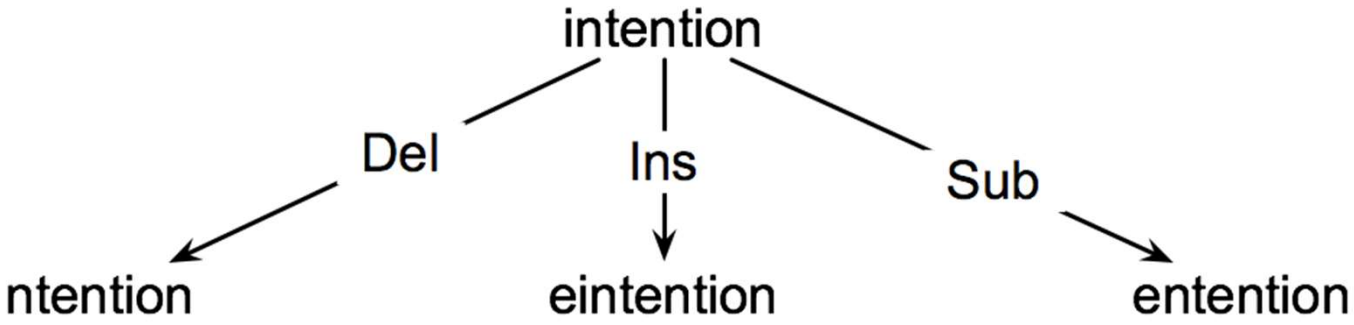
Named Entity Extraction and Entity Coreference

- . **IBM Inc**. announced today
- . **IBM** profits
- . **Stanford President John Hennessy** announced yesterday
- . for **Stanford University President John Hennessy**

How to find the Min Edit Distance?

Searching for a path (sequence of edits) from the start string to the final string:

- **Initial state**: the word we're transforming
- **Operators**: insert, delete, substitute
- **Goal state**: the word we're trying to get to
- **Path cost**: what we want to minimize: the number of edits



Minimum Edit as Search

But the space of all edit sequences is huge!

We can’t afford to navigate naïvely

Lots of distinct paths wind up at the same state.

- We don’t have to keep track of all of them
- Just the shortest path to each of those revisited states.

Defining Minimum Edit Distance

For two strings

1. X of length n
2. Y of length m

We define D(i, j)

the edit distance between X[1..i] and Y[1..j]

i.e., the first i characters of X and the first j characters of Y

The edit distance between X and Y is thus D(n, m)

D => minimum distance

Minimum Edit Distance

Computing Minimum Edit Distance

Minimum Edit Distance (Dynamic Programming)

Dynamic programming: A tabular computation of D(n, m)

Solving problems by combining solutions to subproblems.

Bottom-up

- compute D(i, j) for small i, j
- and compute larger D(i, j) based on previously computed smaller values
  - i.e., compute D(i, j) for all i (0 < i < n) and j (0 < j < m)

Minimum Edit Distance (Levenshtein Distance)

Initialization

```
D(i,0) = i
D(0,j) = j
```

Recurrence Relation:

```
For each  i = 1…M
     For each  j = 1…N
          D(i,j)= min { D(i-1,j) + 1                    (deletion in Y-axis i.e. "j")
                      { D(i,j-1) + 1                    (insert in Y-axis i.e. "j")
                      { D(i-1,j-1) +   2; {if X(i) ≠ Y(j)
                                       0; {if X(i) = Y(j)
```

(subsitute in Y-axis i.e. "j")

Termination:

```
D(N,M) is distance
```

Minimum Edit Distance (Levenshtein Distance)

Levenshtein Distance is named after the Russian scientist Vladimir Levenshtein, who devised the algorithm in 1965.

If you can't spell or pronounce Levenshtein, the metric is also sometimes called Minimum Edit Distance.

The Levenshtein Distance algorithm is used in:

- Spell checking
- Speech recognition
- DNA analysis
- Plagiarism detection

Minimum Edit Distance (Levenshtein Distance)

Step	Description
1	Set n to be the length of s. Set m to be the length of t. If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..m rows and 0..n columns.
2	Initialize the first row to 0..n. Initialize the first column to 0..m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 2.
6	Set cell d[i,j] of the matrix equal to the minimum of: a: The cell immediately above plus 1: d[i-1, j] + 1. b: The cell immediately to the left plus 1: d[i, j-1] + 1. c: The cell diagonally above and to the left plus the cost: d[i-1, j-1] + cost.
7	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

Minimum Edit Distance (Levenshtein Distance)

Example to explain the previous algorithm:

The Levenshtein distance is computed when the source string is “abcdef” and the target string is “azced”.

- String s = abcdef
- String t = azced

Construct a Matrix

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1						
z	2						
c	3						
e	4						
d	5						

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1						
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Once first row (i.e. i=0) and first column (i.e. j=0) is completed then compute Dynamic Programing Algorithm

Minimum Edit Distance

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1						
z	2						
c	3						
e	4						
d	5						

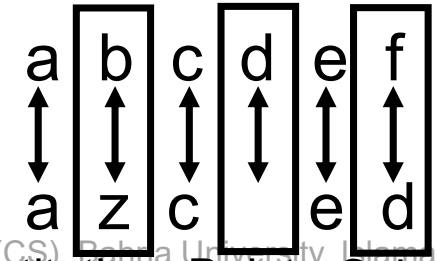
a b c d e f

a z c e d

Minimum Edit Distance

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1						
z	2						
c	3						
e	4						
d	5						



1) b substitution to z
2) d deleted
3) f substitution to d

Therefore, 3 operation are required, cost will be 5

Minimum Edit Distance

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1						
z	2						
c	3						
e	4						
d	5						

The selected cells says that if NULL is at one side and abcdef is on other side, then it requires 6 transition to convert NULL into abcdef.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1						
z	2						
c	3						
e	4						
d	5						

The selected cells says that if NULL is at one side and abzced is on other side, then it requires 5 transition to convert NULL into abzced.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0					
z	2						
c	3						
e	4						
d	5						

Start from this point; if string ***s*** has only one character *a* and string ***t*** has character *a* then how many transition would it make to convert *a* to *a*. Answer is **0** because nothing is going to change.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1				
z	2						
c	3						
e	4						
d	5						

If string ***s*** has character *ab* and string ***t*** has character *a* then how many transition would it make to convert *ab* to *a*.

Answer is **1**, because deleting *b*

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2			
z	2						
c	3						
e	4						
d	5						

If string ***s*** has character *abc* and string ***t*** has character *a* then how many transition would it make to convert *abc* to *a*.

Answer is **2**, because deleting *b*, *c*

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3		
z	2						
c	3						
e	4						
d	5						

If string ***s*** has character *abcd* and string ***t*** has character *a* then how many transition would it make to convert *abcd* to *a*.

Answer is **3**, because deleting *b*, *c*, *d*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	
z	2						
c	3						
e	4						
d	5						

If string ***s*** has character *abcde* and string ***t*** has character *a* then how many transition would it make to convert *abcde* to *a*.

Answer is **4**, because deleting *b*, *c*, *d*, *e*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2						
c	3						
e	4						
d	5						

If string ***s*** has character *abcdef* and string ***t*** has character *a* then how many transition would it make to convert *abcdef* to *a*.

Answer is **5**, because deleting *b*, *c*, *d*, *e*, *f*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1					
c	3						
e	4						
d	5						

If string ***s*** has character *a* and string ***t*** has character *az* then how many transition would it make to convert *a* to *az*.

Answer is **1**, because inserting *z*

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2				
c	3						
e	4						
d	5						

If string ***s*** has character *ab* and string ***t*** has character *az* then how many transition would it make to convert *ab* to *az*.

Answer is **2**, because substituting *b* to *z*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3			
c	3						
e	4						
d	5						

If string ***s*** has character *abc* and string ***t*** has character *az* then how many transition would it make to convert *abc* to *az*.

Answer is **3**, because substituting *b* to *z* and deleting *c*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4		
c	3						
e	4						
d	5						

If string ***s*** has character *abcd* and string ***t*** has character *az* then how many transition would it make to convert *abcd* to *az*.

Answer is **4**, because substituting *b* to *z* and deleting *d*, *c*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	
c	3						
e	4						
d	5						

If string ***s*** has character *abcde* and string ***t*** has character *az* then how many transition would it make to convert *abcde* to *az*.

Answer is **5**, because substituting *b* to *z* and deleting *d*, *c*, *e*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3						
e	4						
d	5						

If string ***s*** has character *abcdef* and string ***t*** has character *az* then how many transition would it make to convert *abcde* to *az*.

Answer is **6**, because substituting *b* to *z* and deleting *d*, *c*, *e*, *f*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2					
e	4						
d	5						

If string ***s*** has character *a* and string ***t*** has character *azc* then how many transition would it make to convert *a* to *azc*.

Answer is **2**, because inserting *z*, *c*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3				
e	4						
d	5						

If string ***s*** has character *ab* and string ***t*** has character *azc* then how many transition would it make to convert *ab* to *azc*.

Answer is **3**, because substitution *b* to *z* and insert *c*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2			
e	4						
d	5						

If string ***s*** has character *abc* and string ***t*** has character *azc* then how many transition would it make to convert *abc* to *azc*.

Answer is **2**, because substitution *b* to *z*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3		
e	4						
d	5						

If string ***s*** has character *abcd* and string ***t*** has character *azc* then how many transition would it make to convert *abcd* to *azc*.

Answer is **3**, because substitution *b* to *z* and deleting *d*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	
e	4						
d	5						

If string ***s*** has character *abcd* and string ***t*** has character *azc* then how many transition would it make to convert *abcd* to *azc*.

Answer is **4**, because substitution *b* to *z* and deleting *d*, *e*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4						
d	5						

If string ***s*** has character *abcdef* and string ***t*** has character *azc* then how many transition would it make to convert *abcdef* to *azc*.

Answer is **5**, because substitution *b* to *z* and deleting *d*, *e*, *f*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3					
d	5						

If string ***s*** has character *a* and string ***t*** has character *azce* then how many transition would it make to convert *a* to *azce*.

Answer is **3**, because inserting *z*, *c*, *e*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4				
d	5						

If string ***s*** has character *ab* and string ***t*** has character *azce* then how many transition would it make to convert *ab* to *azce*.

Answer is **4**, because substituting *b* to *z*, and inserting *c*, *e*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3			
d	5						

If string ***s*** has character *abc* and string ***t*** has character *azce* then how many transition would it make to convert *abc* to *azce*.

Answer is **3**, because substituting *b* to *z*, and inserting *e*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4		
d	5						

If string ***s*** has character *abcd* and string ***t*** has character *azce* then how many transition would it make to convert *abcd* to *azce*.

Answer is **4**, because substituting *b* to *z*, delete *d* and insert *e*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	
d	5						

If string ***s*** has character *abcde* and string ***t*** has character *azce* then how many transition would it make to convert *abcde* to *azce*.

Answer is **3**, because substituting *b* to *z*, and deleting *d*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5						

If string ***s*** has character *abcdef* and string ***t*** has character *azce* then how many transition would it make to convert *abcdef* to *azce*.

Answer is **4**, because substituting *b* to *z*, deleting *d* and *f*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4					

If string ***s*** has character *a* and string ***t*** has character *azced* then how many transition would it make to convert *a* to *azced*.

Answer is **4**, because inserting *z*, *c*, *e*, *d*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5				

If string ***s*** has character *ab* and string ***t*** has character *azced* then how many transition would it make to convert *ab* to *azced*.

Answer is **5**, because substituting *b* to *z*, inserting *c*, *e*, *d*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5	4			

If string ***s*** has character *abc* and string ***t*** has character *azced* then how many transition would it make to convert *abc* to *azced*.

Answer is **4**, because substituting *b* to *z*, inserting *e*, *d*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5	4	3		

If string ***s*** has character *abcd* and string ***t*** has character *azced* then how many transition would it make to convert *abcd* to *azced*.

Answer is **3**, because substituting *b* to *z*, inserting *e*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5	4	3	4	

If string ***s*** has character *abcde* and string ***t*** has character *azced* then how many transition would it make to convert *abcde* to *azced*.

Answer is **4**, because substituting *b* to *z*, delete *d* and insert *d*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5	4	3	4	5

If string ***s*** has character *abcde* and string ***t*** has character *azced* then how many transition would it make to convert *abcde* to *azced*.

Answer is **5**, because substituting *b* to *z*, deleting *d* and substituting *f* to *d*.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5	4	3	4	5

If string ***s*** has character *abcde* and string ***t*** has character *azced* then how many transition would it make to convert *abcde* to *azced*.

Answer is **5**, because substituting *b* to *z*, deleting *d* and substituting *f* to *d*.

Minimum Edit Distance

Back-trace for Computing Alignments

Computing alignments

Edit distance isn't sufficient

We often need to **align** each character of the two strings to each other

We do this by keeping a "back-trace"

Every time new entry in a cell, remember from where it came from

When reach the end,

Trace back the path from the lower right corner to read off the alignment

Adding Back-trace to Minimum Edit Distance

Base conditions:

`D(i,0) = i` `D(0,j) = j`

Termination:

`D(N,M) is distance`

Recurrence Relation:

```
For each  i = 1…M
      For each  j = 1…N
```

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 & \text{deletion} \\ D(i,j-1) + 1 & \text{insertion} \\ D(i-1,j-1) + \begin{cases} 2; \text{ if } X(i) \neq Y(j) \\ 0; \text{ if } X(i) = Y(j) \end{cases} & \text{substitution} \end{cases}$$

$$ptr(i,j) = \begin{cases} LEFT & \text{insertion} \\ Up & \text{deletion} \\ DIAOGNAL & \text{substitution} \end{cases}$$

Minimum Edit Distance

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1						
z	2						
c	3						
e	4						
d	5						

The selected cells says that if NULL is at one side and abcdef is on other side, then it requires 6 transition to convert NULL into abcdef.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1						
z	2						
c	3						
e	4						
d	5						

The selected cells says that if NULL is at one side and abzced is on other side, then it requires 5 transition to convert NULL into abzced.

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0					
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0					
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1				
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1				
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖↑ 2			
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2			
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖↑ 3		
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3		
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖↑ 4	
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↘↑ 5
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2						
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑←					
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1					
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	↖↑ 2				
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2				
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3			
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3			
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4		
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4		
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑	
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3						
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2					
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2					
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3				
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3				
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖↑ 2			
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2			
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖←↑ 3		
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3		
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖↑ 4	
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖↑ 5
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4						
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖↑ 3					
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3					
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4				
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4				
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖←↑ 3			
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3			
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4		
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4		
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖↑ 4	↖ 3	↖↑ 4	↖↑ 3	
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖↑ 4	↖ 3	↖↑ 4	↖ 3	
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖↑ 4	↖ 3	↖↑ 4	↖ 3	↖↑ 4
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖↑ 4	↖ 3	↖↑ 4	↖ 3	↖ 4
d	5						

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖↑ 4	↖ 3	↖↑ 4	↖ 3	↖ 4
d	5	↖↑ 4					

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4	↖ 3	↖ 4
d	5	↖ 4					

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖←↑ 5				

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖←↑ 5				

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	←↖↑ 3	←↖↑ 4	←↖↑ 5	←↖↑ 6
c	3	↖ 2	←↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	←↖↑ 4	↖ 3	←↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	←↖↑ 5	←↖↑ 4			

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖←↑ 5	↖ 4			

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖←↑ 5	↖ 4	↖←↑ 3		

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖←↑ 5	↖ 4	↖ 3		

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖↑ 4	↖ 3	↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖↑ 5	↖ 4	↖ 3	↖↑ 4	

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6
c	3	↖ 2	↖←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖←↑ 4	↖ 3	↖←↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖←↑ 5	↖ 4	↖ 3	↖ 4	

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	←↖↑ 3	←↖↑ 4	←↖↑ 5	←↖↑ 6
c	3	↖ 2	←↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	←↖↑ 4	↖ 3	←↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	←↖↑ 5	↖ 4	↖ 3	↖ 4	←↖↑ 5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	←↖↑ 3	←↖↑ 4	←↖↑ 5	←↖↑ 6
c	3	↖ 2	←↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	←↖↑ 4	↖ 3	←↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	←↖↑ 5	↖ 4	↖ 3	↖ 4	←↖↑ 5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	↖↑ 5	↖↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖↑ 4	↖ 3	↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖↑ 5	↖ 4	↖ 3	↖ 4	↖↑ 5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	←↑ 5	←↑ 6
c	3	↖ 2	←↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	←↑ 4	↖ 3	↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	←↑ 5	↖ 4	↖ 3	↖ 4	←↑ 5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	↖↑ 3	↖↑ 4	←↑ 5	←↑ 6
c	3	↖ 2	↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	↖↑ 4	↖ 3	↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	↖↑ 5	↖ 4	↖ 3	↖ 4	←↑ 5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	←↖↑ 3	←↖↑ 4	←↖↑ 5	←↖↑ 6
c	3	↖ 2	←↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	←↖↑ 4	↖ 3	←↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	←↖↑ 5	↖ 4	↖ 3	↖ 4	←↖↑ 5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5	4	3	4	5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5	4	3	4	5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
z	2	1	2	3	4	5	6
c	3	2	3	2	3	4	5
e	4	3	4	3	4	3	4
d	5	4	5	4	3	4	5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	←↖↑ 3	←↖↑ 4	←↖↑ 5	←↖↑ 6
c	3	↖ 2	←↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	←↖↑ 4	↖ 3	←↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	←↖↑ 5	↖ 4	↖ 3	↖ 4	←↖↑ 5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

		a	b	c	d	e	f
	0	1	2	3	4	5	6
a	1	↖ 0	↖ 1	↖ 2	↖ 3	↖ 4	↖ 5
z	2	↖↑ 1	←↑ 2	←↖↑ 3	←↖↑ 4	←↖↑ 5	←↖↑ 6
c	3	↖ 2	←↖↑ 3	↖ 2	↖ 3	↖ 4	↖ 5
e	4	↖ 3	←↖↑ 4	↖ 3	←↖↑ 4	↖ 3	↖ 4
d	5	↖ 4	←↖↑ 5	↖ 4	↖ 3	↖ 4	←↖↑ 5

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

d	**5**						
e	**4**						
c	**3**						
z	**2**						
a	**1**						
	0	**1**	**2**	**3**	**4**	**5**	**6**
		a	**b**	**c**	**d**	**e**	**f**

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

d	5	4	5	4	3	4	5
e	4	3	4	3	4	3	4
c	3	2	3	2	3	4	5
z	2	1	2	3	4	5	6
a	1	0	1	2	3	4	5
	0	1	2	3	4	5	6
		a	b	c	d	e	f

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

d	5	↙ 4	↖↓ 5	↙ 4	↙ 3	↙ 4	↖↓ 5
e	4	↙ 3	↖↓ 4	↙ 3	↖↓ 4	↙ 3	↙ 4
c	3	↙ 2	↖↓ 3	↙ 2	↙ 3	↙ 4	↙ 5
z	2	↙↓ 1	↖↓ 2	↖↓ 3	↖↓ 4	↖↓ 5	↖↓ 6
a	1	↙ 0	↙ 1	↙ 2	↙ 3	↙ 4	↙ 5
	0	1	2	3	4	5	6
		a	b	c	d	e	f

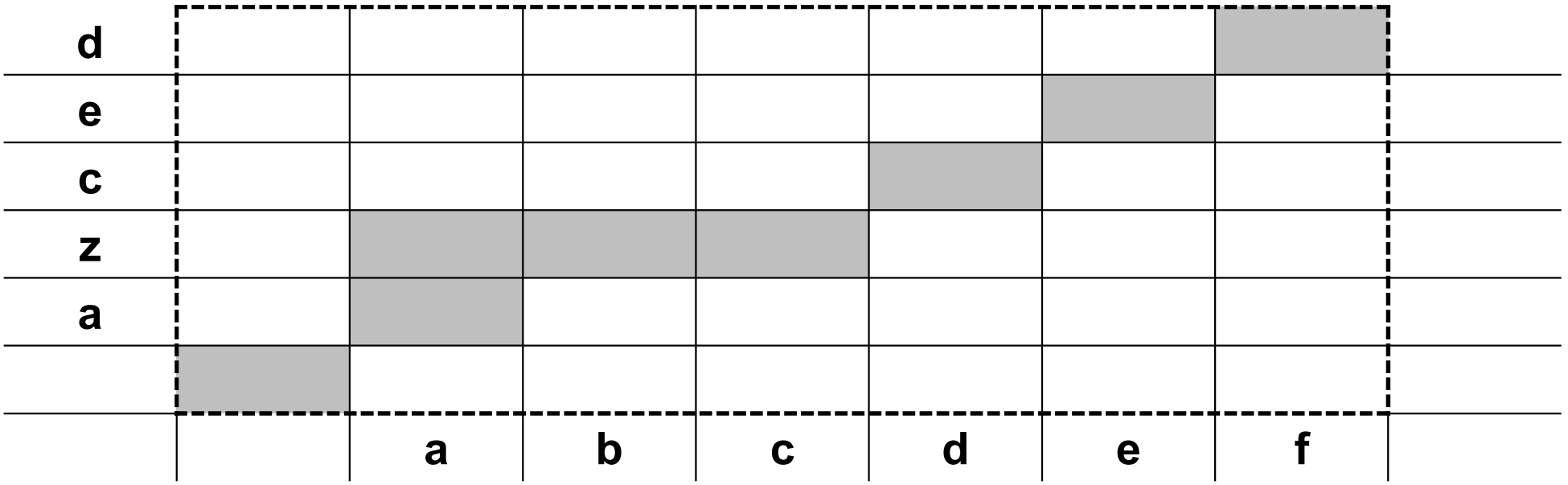
$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)

d	5	↙ 4	↖↙ 5	↙ 4	↙ 3	↙ 4	↖↙ 5
e	4	↙ 3	↖↙ 4	↙ 3	↖↙ 4	↙ 3	↙ 4
c	3	↙ 2	↖↙ 3	↙ 2	↙ 3	↙ 4	↙ 5
z	2	↙ 1	↖ 2	↖↙ 3	↖↙ 4	↖↙ 5	↖↙ 6
a	1	↙ 0	↙ 1	↙ 2	↙ 3	↙ 4	↙ 5
	0	1	2	3	4	5	6
		a	b	c	d	e	f

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit Distance (Levenshtein Distance)



$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Edit Distance

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	**0**	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

Edit Distance

N	1									
O	2									
I	3									
T	4									
N	5									
E	6									
T	7									
N	8									
I	9									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

Edit Distance

N	9	8	9	10	11	12	11	10	9	**8**
O	8	7	8	9	10	11	10	9	8	9
I	7	6	7	8	9	10	9	8	9	10
T	6	5	6	7	8	9	8	9	10	11
N	5	4	5	6	7	8	9	10	11	10
E	4	3	4	5	6	7	8	9	10	9
T	3	4	5	6	7	8	7	8	9	8
N	2	3	4	5	6	7	8	7	8	7
I	1	2	3	4	5	6	7	6	7	8
#	**0**	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

The Edit Distance Table

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	0	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Edit Distance

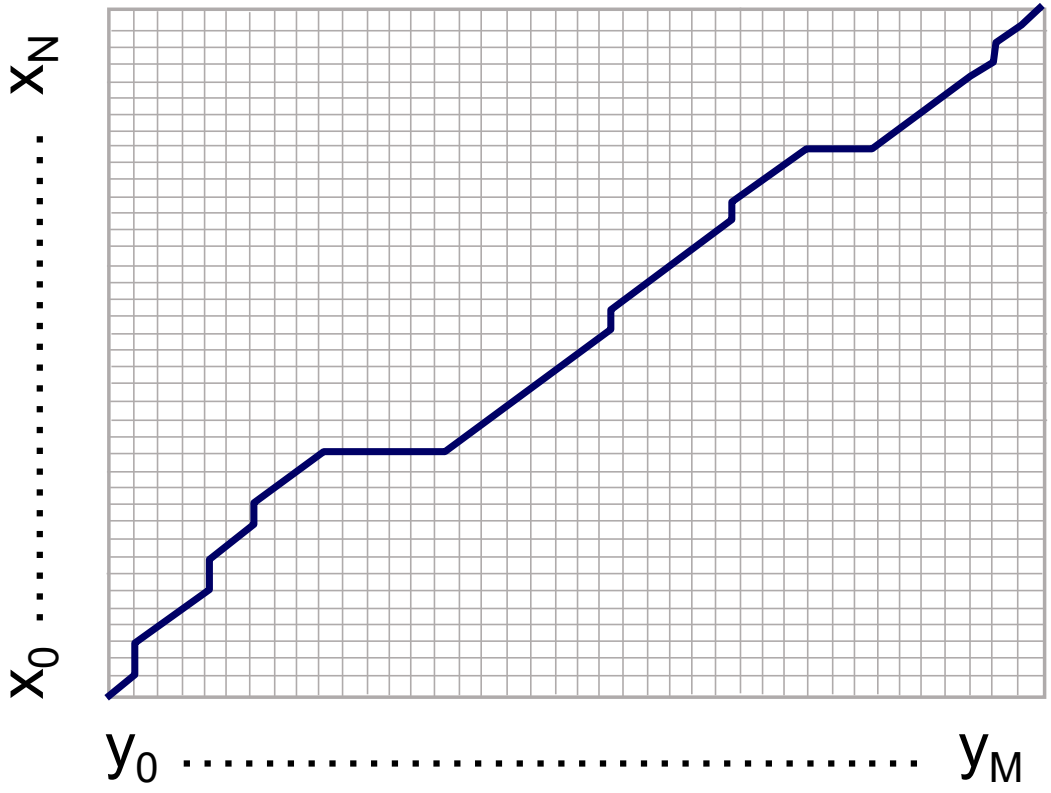
$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	**0**	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

MinEdit with Backtrace

n	9	↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙←↓ 12	↓ 11	↓ 10	↓ 9	↙ **8**	
o	8	↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↓ 10	↓ 9	↙ **8**	← 9	
i	7	↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	↙ **8**	← 9	← 10	
t	6	↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙ **8**	← 9	← 10	←↓ 11	
n	5	↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ **8**	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙↓ 10	
e	4	↙ 3	← 4	↙← **5**	← **6**	← 7	←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	
t	3	↙←↓ 4	↙←↓ **5**	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙ 7	←↓ 8	↙←↓ 9	↓ 8	
n	2	↙←↓ **3**	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↓ 7	↙←↓ 8	↙ 7	
i	**1**	↙←↓ 2	↙←↓ 3	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙ 6	← 7	← 8	
#	**0**	1	2	3	4	5	6	7	8	9	
	#	**e**	**x**	**e**	**c**	**u**	**t**	**i**	**o**	**n**	

The Distance Matrix



Every non-decreasing path

from (0,0) to (M, N)

corresponds to
an alignment
of the two sequences

An optimal alignment is composed of optimal subalignments

Slide adapted from Serafim Batzoglou

Q&A

That’s all for today’s Lecture