

# Distanțe de editare



# Distanțe de editare

Se dau două șiruri de caractere  $s_1$  și  $s_2$ .

**Distanțe de editare** - numărul minim de operații (inserări, modificări, ștergeri de caractere etc) necesar pentru a transforma șirul  $s_1$  în șirul  $s_2$ .

**Distanța de editare Levenshtein** - sunt permise operații de inserare, modificare și ștergere.

**Exemplu:** Distanța de la care la antet este 4.



# Distanțe de editare

- La fiecare nepotrivire a unui caracter cu cel din destinație, avem 3 operații posibile.
- Dacă analizăm, pe rând, fiecare variantă  $\Rightarrow$  backtracking  $\Rightarrow$  **ineficient**

**Soluție - Programare dinamică**

# Distanțe de editare

## Principiu de optimalitate:

Considerăm o transformare cu număr minim de operații:

$$x_1 x_2 \dots x_n$$

$$y_1 y_2 \dots y_m$$

Evidențiem ultima operație.

# Distanțe de editare

$$x_1 x_2 \dots x_n \Rightarrow y_1 y_2 \dots y_m$$

**Evidențiem ultima operație**

- $x_n = y_m$  - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_{m-1}$   
 $(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1}) + \text{păstrăm } x_n$

# Distanțe de editare

$$x_1 x_2 \dots x_n \Rightarrow y_1 y_2 \dots y_m$$

## Evidențiem ultima operație

- $x_n = y_m$  - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_{m-1}$   
 $(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1}) + \text{păstrăm } x_n$
- $x_n$  a fost șters - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_m$  (după care se șterge  $x_n$ )  
 $(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_m) + \text{ștergem } x_n$

# Distanțe de editare

$$x_1 x_2 \dots x_n \Rightarrow y_1 y_2 \dots y_m$$

## Evidențiem ultima operație

- **$x_n = y_m$**  - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_{m-1}$   
 **$(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1}) + \text{păstrăm } x_n$**
- **$x_n$  a fost șters** - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_m$  (după care se șterge  $x_n$ )  
 **$(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_m) + \text{ștergem } x_n$**
- **$x_n$  a fost modificat în  $y_m$**  - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_{m-1}$  (după care se modifică  $x_n$  în  $y_m$ )  
 **$(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1}) + \text{modificăm } x_n \leftrightarrow y_m$**

# Distanțe de editare

$$x_1 x_2 \dots x_n \Rightarrow y_1 y_2 \dots y_m$$

## Evidențiem ultima operație

- $x_n = y_m$  - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_{m-1}$   
 $(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1}) + \text{păstrăm } x_n$
- $x_n$  a fost șters - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_m$  (după care se șterge  $x_n$ )  
 $(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_m) + \text{ștergem } x_n$
- $x_n$  a fost modificat în  $y_m$  - problema se reduce la a transforma  $x_1 \dots x_{n-1}$  în  $y_1 \dots y_{m-1}$  (după care se modifică  $x_n$  în  $y_m$ )  
 $(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1}) + \text{modificăm } x_n \leftrightarrow y_m$
- a fost inserat  $y_m$  - problema se reduce la a transforma  $x_1 \dots x_n$  în  $y_1 \dots y_{m-1}$  (după care se inserează  $y_m$ )  
 $(x_1 \dots x_n \Rightarrow y_1 \dots y_{m-1}) + \text{inserăm } y_m$



# Distanțe de editare

Avem, deci, 4 cazuri. Problema se reduce la a transforma un prefix  $x_1 \dots x_i$  al primului cuvânt într-un prefix  $y_1 \dots y_j$  al celui de-al doilea cuvânt (subprobleme PD).

- $x_i = y_j$ :  $(x_1 \dots x_{i-1} \Rightarrow y_1 \dots y_{j-1})$  + păstrăm  $x_i$
- $(x_1 \dots x_{i-1} \Rightarrow y_1 \dots y_j)$  + ștergem  $x_i$
- $(x_1 \dots x_{i-1} \Rightarrow y_1 \dots y_{j-1})$  + modificăm  $x_i \leftrightarrow y_j$
- $(x_1 \dots x_i \Rightarrow y_1 \dots y_{j-1})$  + inserăm  $y_j$

# Distanțe de editare

## Subprobleme:

$c[i][j]$  = numărul minim de operații de inserare, ștergere, modificare pentru a transforma  $x_1 \dots x_i$  în  $y_1 \dots y_j$

**Relații de recurență** - corespund cazurilor:

- $x_i = y_j$ :  $(x_1 \dots x_{i-1} \Rightarrow y_1 \dots y_{j-1}) + \text{păstrăm } x_i$
- $(x_1 \dots x_{i-1} \Rightarrow y_1 \dots y_j) + \text{ștergem } x_i$
- $(x_1 \dots x_{i-1} \Rightarrow y_1 \dots y_{j-1}) + \text{modificăm } x_i \leftrightarrow y_j$
- $(x_1 \dots x_i \Rightarrow y_1 \dots y_{j-1}) + \text{inserăm } y_j$

$$c[i][j] = \begin{cases} c[i-1][j-1], & \text{daca } x_i = y_j \\ 1 + \min\{c[i-1][j], c[i-1][j-1], c[i][j-1]\}, & \text{altfel} \end{cases}$$

$\uparrow$                        $\uparrow$                        $\uparrow$   
ștergem  $x_i$                $x_i \leftrightarrow y_j$               inserăm  $y_j$

# Distanțe de editare

**Soluția:**  $c[n][m]$

**Ce valori din  $c$  știm direct:**

- $c[0][0] = 0$  (ambele cuvinte sunt vide)
- **pentru  $i = 0$  sau  $j = 0$**  (unul din cuvinte este vid):
  - $x_1 \dots x_{i-1} \Rightarrow$  secvență vidă      prin  $i$  ștergeri succesive
  - secvență vidă  $\Rightarrow y_1 \dots y_j$       prin  $j$  inserări succesive

$$c[0][0] = 0$$

$$c[i][0] = 1 + c[i-1][0] = i, \text{ pentru } i = 1, \dots, n$$

$$c[0][j] = 1 + c[0][j-1] = j, \text{ pentru } j = 1, \dots, m$$

**Ordine de calcul a matricei:**  $i = 0, \dots, n; j = 0, \dots, m$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1					
2 a	2					
3 r	3					
4 e	4					

$i = 0: c[0][j] = j$   
 $j = 0: c[i][0] = i$

# Distanțe de editare

Exemplu:  $s_1 = \text{care} \Rightarrow s_2 = \text{antet}$

		0	1	2	3	4	5
			a	n	t	e	t
c:	0	0	1	2	3	4	5
	1 c	1					
	2 a	2					
	3 r	3					
	4 e	4					

$i = 1, j = 1$ :

$s_1[i] = c, s_2[j] = a$  - sunt diferite  $\Rightarrow$

# Distanțe de editare

Exemplu:  $s_1 = \text{care} \Rightarrow s_2 = \text{antet}$

		0	1	2	3	4	5
			a	n	t	e	t
c:	0	0	1	2	3	4	5
	1 c	1	1				
	2 a	2					
	3 r	3					
	4 e	4					

$i = 1, j = 1$ :

$s_1[i] = c, s_2[j] = a$  - sunt diferite  $\Rightarrow$

$$c[i][j] = 1 + \min(c[i-1][j], c[i][j-1], c[i-1][j-1]) = 1 + 0 = 1$$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

**c:**

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2					
3 r	3					
4 e	4					

← c nu aparține cuvântului **antet**

$$c[i][j] = 1 + \min(c[i-1][j], c[i][j-1], c[i-1][j-1])$$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2	1				
3 r	3					
4 e	4					

← c nu aparține cuvântului **antet**

$i = 2, j = 1$ :

$s1[i] = a, s2[j] = a$  - sunt egale  $\Rightarrow$

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



# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

		0	1	2	3	4	5
			a	n	t	e	t
0	0	1	2	3	4	5	
1 c	1	1	2	3	4	5	
2 a	2	1	2	3	4	5	←
3 r	3						
4 e	4						

$$c[i][j] = 1 + \min(c[i-1][j], c[i][j-1], c[i-1][j-1])$$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

		0	1	2	3	4	5
			a	n	t	e	t
c:	0	0	1	2	3	4	5
	1 c	1	1	2	3	4	5
	2 a	2	1	2	3	4	5
	3 r	3	2				
	4 e	4					

$i = 3, j = 1$ :

$s1[i] = r, s2[j] = a$  - sunt diferite  $\Rightarrow$

$c[i][j] = 1 + \min(c[i-1][j], c[i][j-1], c[i-1][j-1]) = 1 + 1 = 2$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2	1	2	3	4	5
3 r	3	2	2	3	4	5
4 e	4					

← r nu aparține cuvântului **antet**

$$c[i][j] = 1 + \min(c[i-1][j], c[i][j-1], c[i-1][j-1])$$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2	1	2	3	4	5
3 r	3	2	2	3	4	5
4 e	4	3	3	3	3	

$i = 4, j = 4$ :

$s1[i] = e, s2[j] = e$  - sunt egale  $\Rightarrow$

$c[i][j] = c[i-1][j-1] = 3$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2	1	2	3	4	5
3 r	3	2	2	3	4	5
4 e	4	3	3	3	3	4

Soluția:  $c[4][5] = 4$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

		0	1	2	3	4	5
			a	n	t	e	t
c:	0	0	1	2	3	4	5
	1 c	1	1	2	3	4	5
	2 a	2	1	2	3	4	5
	3 r	3	2	2	3	4	5
	4 e	4	3	3	3	3	4

**Soluția:**  $c[4][5] = 4$   
**Cum determinăm operațiile?**

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

		0	1	2	3	4	5
			a	n	t	e	t
0		0	1	2	3	4	5
1 c		1	1	2	3	4	5
2 a		2	1	2	3	4	5
3 r		3	2	2	3	4	5
4 e		4	3	3	3	3	4

**Soluția:**  $c[4][5] = 4$

**Cum determinăm operațiile?**

Mergând succesiv înapoi de la (4, 5) în celula pentru care s-a obținut egalitatea în relația de recurență:

$$c[i][j] = \begin{cases} c[i-1][j-1], & \text{daca } x_i = y_j \\ 1 + \min\{c[i-1][j], c[i-1][j-1], c[i][j-1]\}, & \text{altfel} \end{cases}$$

↑  
ștergem  $x_i$

↑  
 $x_i \leftrightarrow y_j$

↑  
inserăm  $y_j$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

		0	1	2	3	4	5
			a	n	t	e	t
0		0	1	2	3	4	5
1 c		1	1	2	3	4	5
2 a		2	1	2	3	4	5
3 r		3	2	2	3	4	5
4 e		4	3	3	3	3	4

Soluția:  $c[4][5] = 4$

$s1[4] \neq s2[5] \Rightarrow$

$c[4][5] = 1 + \min(c[4][4], c[3][5], c[3][4])$

$= 1 + c[4][4] \Rightarrow$

(4,5) s-a obținut din (4,4) prin inserarea caracterului  $s2[5] = t$



# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2	1	2	3	4	5
3 r	3	2	2	3	4	5
4 e	4	3	3	3	3	4

inserăm t

$s1[4] = s2[4] \Rightarrow$

$c[4][4] = c[3][3]$  și nu s-a făcut nicio operație (păstrăm e)

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

		0	1	2	3	4	5
			a	n	t	e	t
0		0	1	2	3	4	5
1 c		1	1	2	3	4	5
2 a		2	1	2	3	4	5
3 r		3	2	2	3	4	5
4 e		4	3	3	3	3	4

păstrăm e  
inserăm t

$s1[3] \neq s2[3] \Rightarrow$

$c[3][3] = 1 + \min(c[2][3], c[3][2], c[2][2])$

$= 1 + c[3][2] = 1 + c[2][2] \Rightarrow$

**!! soluția nu este unică**  $\Rightarrow$  alegem una:  $c[3][3] = 1 + c[3][2] \Rightarrow (3,3)$  s-a obținut din  $(3,2)$  prin inserarea caracterului  $s2[3] = t$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

**c:**

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2	1	2	3	4	5
3 r	3	2	2	3	4	5
4 e	4	3	3	3	3	4

inserăm t, păstrăm e

inserăm t

$s1[3] \neq s2[2] \Rightarrow$

$c[3][2] = 1 + \min(c[2][2], c[3][1], c[2][1])$

$= 1 + c[2][1] \Rightarrow$

(3,2) s-a obținut din (2,1) prin modificarea  $s1[3] = r \leftrightarrow s2[2] = n$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2	1	2	3	4	5
3 r	3	2	2	3	4	5
4 e	4	3	3	3	3	4

modificăm  $r \leftrightarrow n$

inserăm t, păstrăm e

inserăm t

$s1[2] = s2[1] \Rightarrow$

$c[2][1] = c[1][0]$  și nu s-a făcut nicio operație (păstrăm a)

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

		0	1	2	3	4	5
			a	n	t	e	t
0	0	1	2	3	4	5	
1 c	1	1	2	3	4	5	
2 a	2	1	2	3	4	5	
3 r	3	2	2	3	4	5	
4 e	4	3	3	3	3	4	

păstrăm a

modificăm r  $\leftrightarrow$  n

inserăm t, păstrăm e

inserăm t

**i = 1, j = 0:**

Deoarece  $j = 0$ ,  $c[1][0] = 1 + c[0][0]$  corespunzător unei ștergeri  $\rightarrow$  a caracterului  $s1[i] = c$

# Distanțe de editare

Exemplu: care  $\Rightarrow$  antet

**c:**

	0	1	2	3	4	5
		a	n	t	e	t
0	0	1	2	3	4	5
1 c	1	1	2	3	4	5
2 a	2	1	2	3	4	5
3 r	3	2	2	3	4	5
4 e	4	3	3	3	3	4

ștergem c

păstrăm a

modificăm r  $\leftrightarrow$  n

inserăm t, păstrăm e

inserăm t

# Distanțe de editare

**Complexitate?**

# Distanțe de editare

**Complexitate?**      **$O(nm)$**



