Se dau două șiruri de caractere s₁ și s₂.

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.

care
$$\xrightarrow{}$$
 are $\xrightarrow{}$ ane $\xrightarrow{}$ ante $\xrightarrow{}$ ante $\xrightarrow{}$ ante $\xrightarrow{}$ ante $\xrightarrow{}$ inserăm t

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

Soluție - Programare dinamică

Principiu de optimalitate:

Considerăm o transformare cu număr minim de operații:

$$X_1X_2...X_n$$

$$y_1 y_2 \dots y_m$$

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

```
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}) + păstrăm x_n
```

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

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

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

- $x_n = y_m$ problema se reduce la a transforma $x_1 \dots x_{n-1}$ în $y_1 \dots y_{m-1}$ $(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1})$ + păstrăm x_n
- x_n a fost şters problema se reduce la a transforma $x_1...x_{n-1}$ în $y_1...y_m$ (după care se şterge x_n) $x_1...x_{n-1} \Rightarrow y_1...y_m$) + ştergem 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) ($x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1}$) + modificăm $x_n \leftrightarrow y_m$

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

- $x_n = y_m$ problema se reduce la a transforma $x_1 \dots x_{n-1}$ în $y_1 \dots y_{m-1}$ $(x_1 \dots x_{n-1} \Rightarrow y_1 \dots y_{m-1})$ + păstrăm x_n
- x_n a fost şters problema se reduce la a transforma $x_1...x_{n-1}$ în $y_1...y_m$ (după care se șterge x_n) $x_1...x_{n-1} \Rightarrow y_1...y_m$) + ştergem 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 x₁...x_n în y₁...y_{m-1} (după care se inserează y_m)

$$(x_1...x_n \Rightarrow y_1...y_{m-1}) + inserăm y_m$$

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

Subprobleme:

```
\mathbf{c[i][j]} =  numărul minim de operații de inserare, ștergere, modificare pentru a transforma \mathbf{x}_1...\mathbf{x}_i în \mathbf{y}_1...\mathbf{y}_i
```

Relații de recurență - corespund cazurilor:

```
Soluția: c[n][m]
```

Ce valori din c ştim direct:

```
c[0][0] = 0 (ambele cuvinte sunt vide)

pentru i = 0 sau j = 0 (unul din cuvinte este vid):

x_1...x_{i-1} \Rightarrow \text{secven} \dagger \text{ vidă} \qquad \text{prin i ştergeri succesive}
\text{secven} \dagger \text{ vidă} \Rightarrow y_1...y_j \qquad \text{prin j inserări succesive}
c[0][0] = 0
c[i][0] = 1 + c[i-1][0] = i, \text{ pentru } i = 1, ..., n
c[0][j] = 1 + c[0][j-1] = j, \text{ pentru } j = 1, ..., m
```

Ordine de calcul a matricei: i = 0, ..., n; j = 0, ..., m

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

Exemplu: $s_1 = care \Rightarrow s_2 = antet$

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

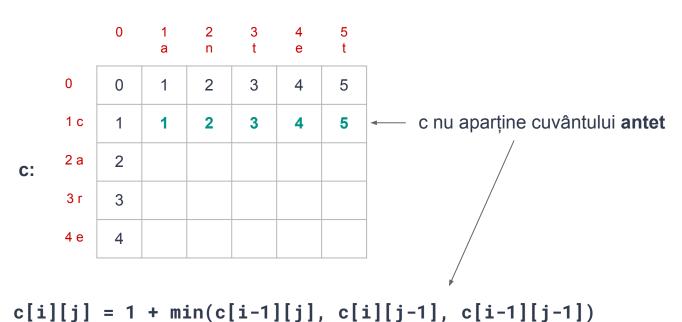
```
i = 1, j = 1:
    s1[i] = c, s2[j] = a - sunt diferite ⇒
```

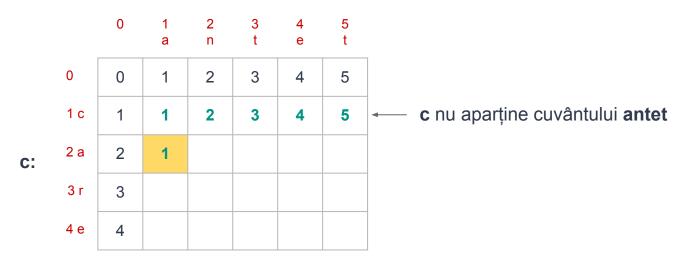
```
Exemplu: s_1 = care \Rightarrow s_2 = antet
```

```
i = 1, j = 1:

s1[i] = c, s2[j] = a - sunt diferite <math>\Rightarrow

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





```
i = 2, j = 1:

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

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

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

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

```
0 1 2 3 4 5 t e t

0 0 1 2 3 4 5

1c 1 1 2 3 4 5

c: 2a 2 1 2 3 4 5

3r 3 2 4 6
```

```
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
```

		0	1 a	2 n	3 t	4 e	5 t	
	0	0	1	2	3	4	5	
	1 c	1	1	2	3	4	5	
c:	2 a	2	1	2	3	4	5	
	3 r	3	2	2	3	4	5	r nu aparține cuvântului antet
	4 e	4						

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

		0	1 a	2 n	3 t	4 e	5 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	

```
i = 4, j = 4:

s1[i] = e, s2[j] = e - sunt egale <math>\Rightarrow

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

Exemplu: care ⇒ antet

		0	1 a	2 n	3 t	4 e	5 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

Exemplu: care ⇒ antet

		0	1 a	2 n	3 t	4 e	5 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?

Exemplu: care ⇒ antet

		0	1 a	2 n	3 t	4 e	5 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?

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

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

$$\uparrow \qquad \uparrow \qquad \uparrow$$

$$\text{ştergem } \mathbf{x_i} \leftarrow \mathbf{y_j} \quad \text{inserăm } \mathbf{y_j}$$

Exemplu: care ⇒ antet

```
0
      0
                         3
                                     5
                               4
1 c
                                      5
2 a
             1
                         3
                               4
                                      5
3 r
                               4
                                      5
4 e
       4
                                      4
```

Soluția: c[4][5] = 4

```
s1[4] != 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
```

Exemplu: care ⇒ antet

```
0
      0
                         3
                                     5
1 c
                                     5
2 a
             1
                         3
                               4
                                     5
3 r
                               4
                                     5
4 e
      4
```

inserăm t

```
s1[4] = s2[4] \Rightarrow
c[4][4] = c[3][3] și nu s-a făcut nicio operație (păstrăm e)
```

Exemplu: care ⇒ antet

```
0
       0
                          3
                                      5
                                4
1 c
                                       5
2 a
             1
                          3
                                4
                                       5
3 r
                                      5
                                4
4 e
       4
```

păstrăm e inserăm t

```
s1[3] != s2[3] ⇒ c[3][3] = 1 + min(c[2][3], c[3][2], c[2][2])= 1 + c[3][2] = 1 + c[2][2] ⇒!! soluția nu este unică ⇒ alegem una: c[3][3] = 1 + c[3][2] ⇒ (3,3) s-a obținut din (3,2) prin inserarea caracterului s2[3] = t
```

Exemplu: care ⇒ antet

```
0
                          3
                                       5
       0
1 c
                                       5
2 a
             1
                          3
                                       5
                                4
3 r
                                4
                                       5
4 e
       4
```

inserăm t, păstrăm e
inserăm t

```
s1[3] != 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 <math>\leftrightarrow s2[2] = n
```

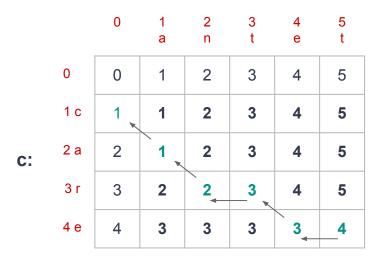
Exemplu: care ⇒ antet

```
0
      0
                         3
                                     5
1 c
                                     5
2 a
                         3
                                     5
                               4
3 r
             2
                               4
                                     5
4 e
      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)
```

Exemplu: care ⇒ antet



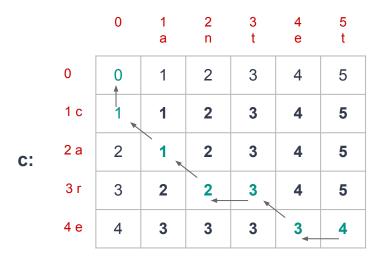
păstrăm a

modificăm r ↔ 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
```



```
ştergem c
păstrăm a
modificăm r ↔ n
inserăm t, păstrăm e
inserăm t
```

Complexitate?

Complexitate? O(nm)

