

2017

PISLO8. Задача о расстоянии Левенштейна.

		M	A	E	S	T	R	O
	0	1	2	3	4	5	6	7
J	1	1	2	3	4	5	6	7
A	2	2	1	2	3	4	5	6
C	3	3	2	2	3	3	4	5
K	4	4	3	3	3	4	4	4
E	5	5	4	4	4	4	5	5
R	6	6	5	5	5	5	4	5
S	7	7	6	6	6	6	5	5



Кафедра экономической информатики. Бгуир., 2017

PISLO8. Задача о расстоянии Левенштейна.

➤ Общие принципы построения алгоритма

➤ Задача о расстоянии редактирования

➤ Подзадачи и рекуррентное соотношение

➤ Решение рекурсией сверху-вниз “Top-Down”

➤ Решение итерацией снизу-вверх “Down-Top”

➤ Восстановление решения

➤ Расход памяти

➤ Применение

➤ Задачи (A) (B) (C)

➤ Материалы: <http://tinyurl.com/ei-pisl>

➤ Github: <https://github.com/Khmelov/PISL2017-01-26>

Кафедра экономической информатики. Бгуир. 2017

2

PISLO8. Общие вопросы дин. прогр.

1. Какие значения мы вычисляем (что ищем)

2. Как их вычислять (какое рекуррентное соотношение)

3. Какие начальные значения (инициализация рекуррентных соотношений)

4. Направление расчета (рекурсия или итерация)

5. Где искать ответ

Для примера давайте представим задачу про числа Фибоначчи как задачу дин. прогр.

1. F(n)

2. F(n)=F(n-1)+F(n-2)

3. F(0)=0, F(1)=1

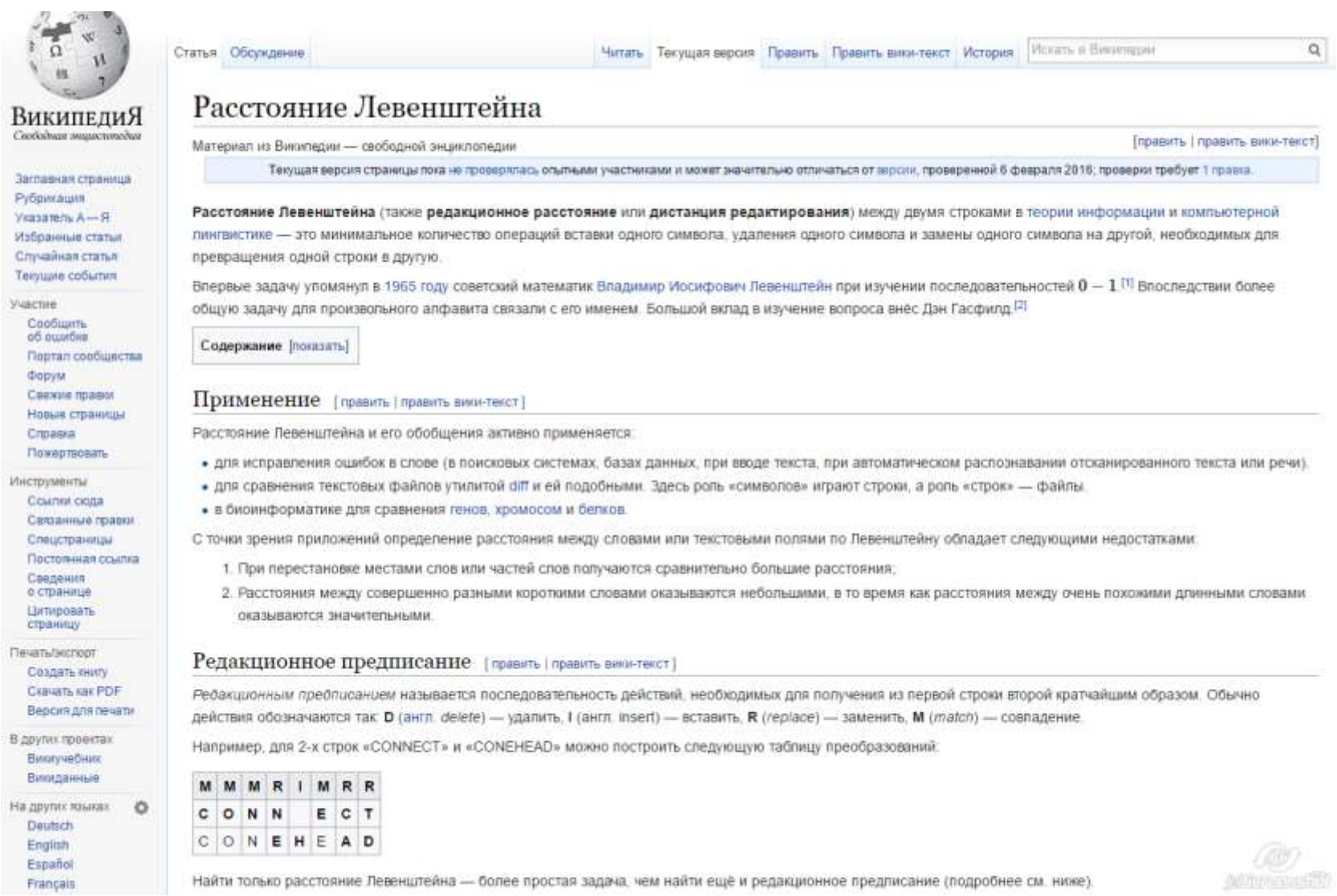
4. F(n)=F(n-1)+F(n-2) – рекурсия или for 0..n – итерация

5. result F(n) или последний элемент в массиве.

Кафедра экономической информатики. Бгуир. 2017

3

PISLO8. Задача о расстоянии Левенштейна.



Кафедра экономической информатики. Бгуир. 2017

4

PISLO8. Задача о расстоянии Левенштейна.

Расстояние редактирования

Вход: строки $A[1 \dots n]$ и $B[1 \dots m]$.

Выход: минимальное количество вставок, удалений и замен символов, необходимое для преобразования A в B . Данное число называется расстоянием редактирования и расстоянием Левенштейна.

PISLO8. Задача о расстоянии Левенштейна.

Выравнивание

Пример

E	D	I	-	T	I	N	G	-
-	D	I	S	T	A	N	C	E

стоимость: 5

PISLO8. Задача о расстоянии Левенштейна.

Выравнивание

Пример

совпадения замены/несовпадения

E	D	I	-	T	I	N	G	-
-	D	I	S	T	A	N	C	E

удаления вставки

стоимость: 5

PISLO8. Задача о расстоянии Левенштейна.

Интуиция

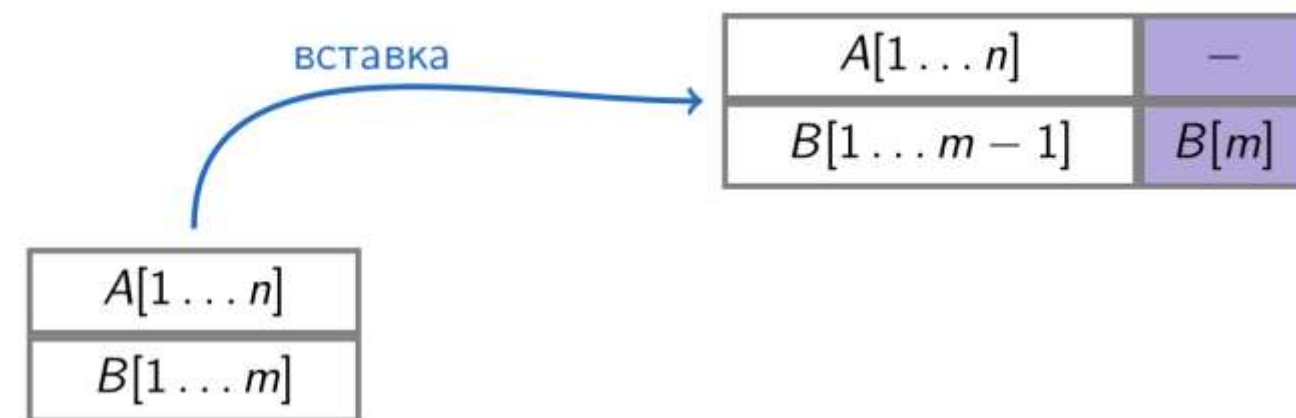
Рассмотрим последний столбец оптимального выравнивания строк $A[1 \dots n]$ и $B[1 \dots m]$:

$A[1 \dots n]$
$B[1 \dots m]$

PISLO8. Задача о расстоянии Левенштейна.

Интуиция

Рассмотрим последний столбец оптимального выравнивания строк $A[1 \dots n]$ и $B[1 \dots m]$:



PISLO8. Задача о расстоянии Левенштейна.

Интуиция

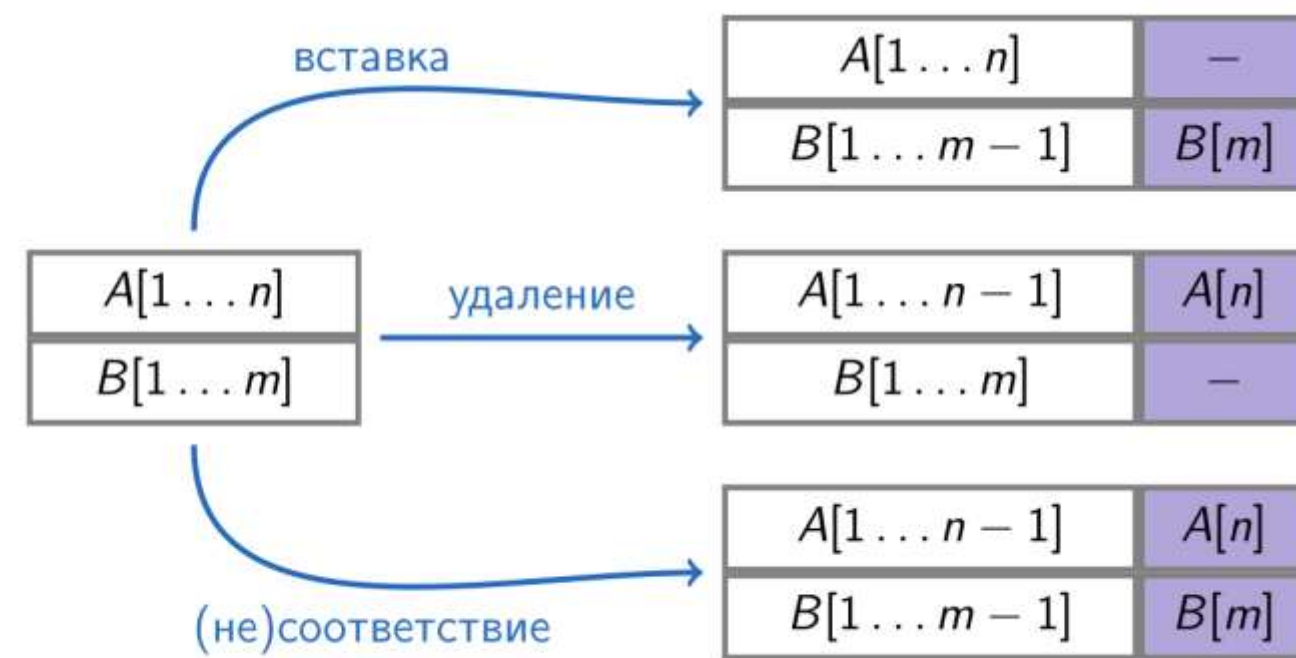
Рассмотрим последний столбец оптимального выравнивания строк $A[1 \dots n]$ и $B[1 \dots m]$:



PISLO8. Задача о расстоянии Левенштейна.

Интуиция

Рассмотрим последний столбец оптимального выравнивания строк $A[1 \dots n]$ и $B[1 \dots m]$:



PISLO8. Задача о расстоянии Левенштейна.

Подзадачи и рекуррентное соотношение

- Пусть $D[i, j]$ — расстояние редактирования строк $A[1 \dots i]$ и $B[1 \dots j]$.
- Последний столбец их оптимального выравнивания — это вставка, удаление или (не)соответствие.
- Выравнивание без последнего столбца является оптимальным выравниванием соответствующих префиксов («вырезать и вставить»).
- Поэтому

$$D[i, j] = \min \begin{cases} D[i, j-1] + 1, & \text{(вставка)} \\ D[i-1, j] + 1, & \text{(удаление)} \\ D[i-1, j-1] + \text{diff}(A[i], B[j]), & \text{((не)соотв.)} \end{cases}$$

PISLO8. Задача о расстоянии Левенштейна.

Дин. прог. сверху вниз

Инициализация

создать двумерный массив $D[0 \dots n, 0 \dots m]$
инициализировать все ячейки значением ∞

PISLO8. Задача о расстоянии Левенштейна.

Дин. прог. сверху вниз

Инициализация

создать двумерный массив $D[0 \dots n, 0 \dots m]$
инициализировать все ячейки значением ∞

Функция EDITDISTTD(i, j)

если $D[i, j] = \infty$:
 если $i = 0$: $D[i, j] \leftarrow j$
 иначе если $j = 0$: $D[i, j] \leftarrow i$

PISLO8. Задача о расстоянии Левенштейна.

Дин. прог. сверху вниз

Инициализация

создать двумерный массив $D[0 \dots n, 0 \dots m]$
инициализировать все ячейки значением ∞

Функция EDITDISTTD(i, j)

если $D[i, j] = \infty$:
 если $i = 0$: $D[i, j] \leftarrow j$
 иначе если $j = 0$: $D[i, j] \leftarrow i$
 иначе:
 $ins \leftarrow \text{EDITDISTTD}(i, j - 1) + 1$
 $del \leftarrow \text{EDITDISTTD}(i - 1, j) + 1$
 $sub \leftarrow \text{EDITDISTTD}(i - 1, j - 1) + \text{diff}(A[i], B[j])$
 $D[i, j] \leftarrow \min(ins, del, sub)$
вернуть $D[i, j]$

PISLO8. Задача о расстоянии Левенштейна.

Время работы

Лемма

Время работы алгоритма EDITDISTTD(n, m) есть $O(nm)$.

PISLO8. Задача о расстоянии Левенштейна.

Время работы

Лемма

Время работы алгоритма $\text{EDITDISTTD}(n, m)$ есть $O(nm)$.

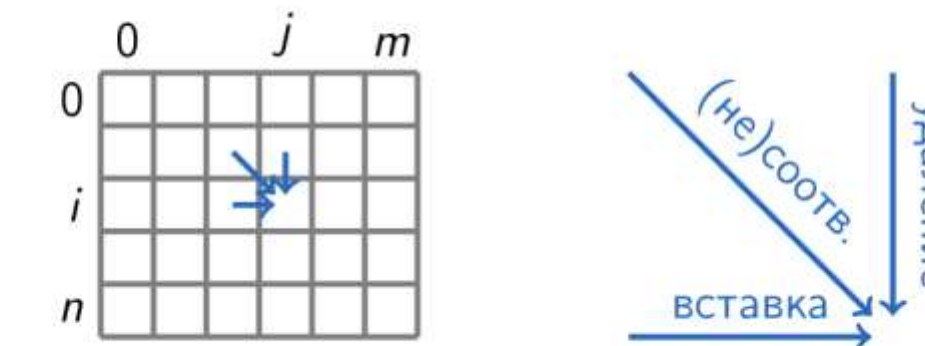
Доказательство

- Только mn рекурсивных вызовов могут быть "серьёзными" (не просто доступ к ячейке таблицы).
- Несерьёзные вызовы требуют времени $O(1)$. Это время можно учесть в вызывающей функции.
- Каждый серьёзный вызов также требует времени $O(1)$ (без учёта времени на другие рекурсивные вызовы). \square

PISLO8. Задача о расстоянии Левенштейна.

Заполнение таблицы

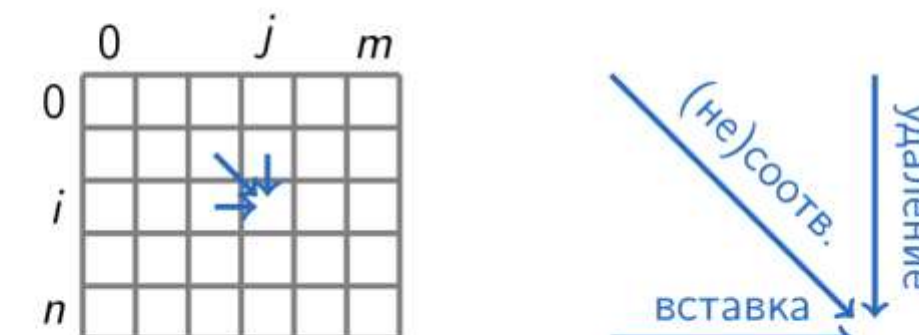
- $D[i, j]$ зависит от $D[i-1, j-1]$, $D[i-1, j]$ и $D[i, j-1]$:



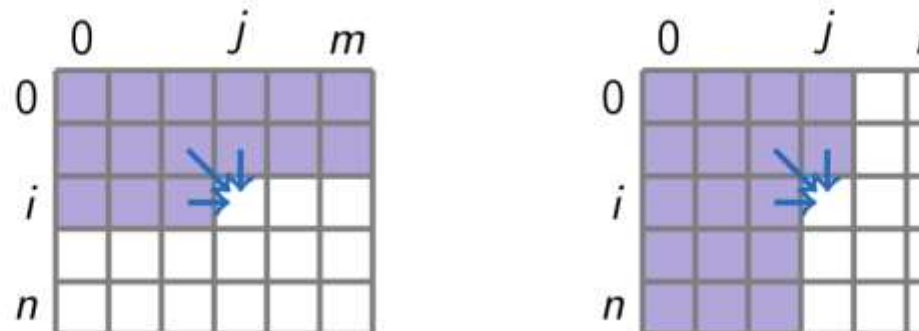
PISLO8. Задача о расстоянии Левенштейна.

Заполнение таблицы

- $D[i, j]$ зависит от $D[i-1, j-1]$, $D[i-1, j]$ и $D[i, j-1]$:



- Можно заполнять таблицу строка за строкой или столбец за столбцом:



PISLO8. Задача о расстоянии Левенштейна.

Дин. прог. снизу вверх

Функция $\text{EDITDISTBU}(A[1 \dots n], B[1 \dots m])$

```
создать массив  $D[0 \dots n, 0 \dots m]$ 
для  $i$  от 0 до  $n$ :
     $D[i, 0] \leftarrow i$ 
для  $j$  от 0 до  $m$ :
     $D[0, j] \leftarrow j$ 
для  $i$  от 1 до  $n$ :
    для  $j$  от 1 до  $m$ :
         $c \leftarrow \text{diff}(A[i], B[j])$ 
         $D[i, j] \leftarrow \min(D[i-1, j]+1, D[i, j-1]+1, D[i-1, j-1]+c)$ 
вернуть  $D[n, m]$ 
```

PISLO8. Задача о расстоянии Левенштейна.

Дин. прог. снизу вверх

Функция `EDITDISTBU(A[1...n], B[1...m])`
создать массив $D[0...n, 0...m]$
для i от 0 до n :
 $D[i, 0] \leftarrow i$
для j от 0 до m :
 $D[0, j] \leftarrow j$
для i от 1 до n :
 для j от 1 до m :
 $c \leftarrow \text{diff}(A[i], B[j])$
 $D[i, j] \leftarrow \min(D[i-1, j]+1, D[i, j-1]+1, D[i-1, j-1]+c)$
вернуть $D[n, m]$

Время работы: $O(nm)$.

Кафедра экономической информатики. Бгуир. 2017

21

PISLO8. Задача о расстоянии Левенштейна.

Пример

			E	D	I	T	I	N	G
	0	1	2	3	4	5	6	7	
0	0	1	2	3	4	5	6	7	
D	1	1							
I	2	2							
S	3	3							
T	4	4							
A	5	5							
N	6	6							
C	7	7							
E	8	8							

Кафедра экономической информатики. Бгуир. 2017

22

PISLO8. Задача о расстоянии Левенштейна.

Пример

			E	D	I	T	I	N	G
	0	1	2	3	4	5	6	7	
0	0	1	2	3	4	5	6	7	
D	1	1							
I	2	2							
S	3	3							
T	4	4							
A	5	5							
N	6	6							
C	7	7							
E	8	8							

Кафедра экономической информатики. Бгуир. 2017

23

PISLO8. Задача о расстоянии Левенштейна.

Пример

			E	D	I	T	I	N	G
	0	1	2	3	4	5	6	7	
0	0	1	2	3	4	5	6	7	
D	1	1	1						
I	2	2							
S	3	3							
T	4	4							
A	5	5							
N	6	6							
C	7	7							
E	8	8							

Кафедра экономической информатики. Бгуир. 2017

24

PISLO8. Задача о расстоянии Левенштейна.

Пример

			E	D	I	T	I	N	G	
	0	1	2	3	4	5	6	7		
0	0	1	2	3	4	5	6	7		
D 1	1	1								
I 2	2									
S 3	3									
T 4	4									
A 5	5									
N 6	6									
C 7	7									
E 8	8									

Кафедра экономической информатики. Бгуир. 2017

25

PISLO8. Задача о расстоянии Левенштейна.

Пример

				E	D	I	T	I	N	G	
	0	1	2	3	4	5	6	7			
0	0	1	2	3	4	5	6	7			
D 1	1	1	1								
I 2	2										
S 3	3										
T 4	4										
A 5	5										
N 6	6										
C 7	7										
E 8	8										

Кафедра экономической информатики. Бгуир. 2017

26

PISLO8. Задача о расстоянии Левенштейна.

Пример

				E	D	I	T	I	N	G	
	0	1	2	3	4	5	6	7			
0	0	1	2	3	4	5	6	7			
D 1	1	1	1								
I 2	2										
S 3	3										
T 4	4										
A 5	5										
N 6	6										
C 7	7										
E 8	8										

Кафедра экономической информатики. Бгуир. 2017

27

PISLO8. Задача о расстоянии Левенштейна.

Пример

					E	D	I	T	I	N	G	
	0	1	2	3	4	5	6	7				
0	0	1	2	3	4	5	6	7				
D 1	1	1	1	2								
I 2	2											
S 3	3											
T 4	4											
A 5	5											
N 6	6											
C 7	7											
E 8	8											

Кафедра экономической информатики. Бгуир. 2017

28

PISLO8. Задача о расстоянии Левенштейна.

Пример

		E	D	I	T	I	N	G	
		0	1	2	3	4	5	6	7
	0	0	1	2	3	4	5	6	7
D	1	1	1	1	2	3	4	5	6
I	2	2	2	2	1	2	3	4	5
S	3	3	3	3	2	2	3	4	5
T	4	4	4	4	3	2	3	4	5
A	5	5	5	5	4	3	3	4	5
N	6	6	6	6	5	4	4	3	4
C	7	7	7	7	6	5	5	4	4
E	8	8	7	8	7	6	6	5	5

PISLO8. Задача о расстоянии Левенштейна.

Восстановление решения

- Чтобы восстановить решение, пойдём обратно от ячейки $[n, m]$ к ячейке $[0, 0]$.
- Если $D[i, j] = D[i - 1, j] + 1$, то найдётся оптимальное выравнивание, последним столбцом которого является удаление.
- Если $D[i, j] = D[i, j - 1] + 1$, то найдётся оптимальное выравнивание, последним столбцом которого является вставка.
- Если $D[i, j] = D[i - 1, j - 1] + \text{diff}(A[i], B[j])$, то найдётся оптимальное выравнивание, последним столбцом которого является замена/несоответствие (если $A[i] \neq B[j]$) или соответствие (если $A[i] = B[j]$).

PISLO8. Задача о расстоянии Левенштейна.

Восстановление решения

- Чтобы восстановить решение, пойдём обратно от ячейки $[n, m]$ к ячейке $[0, 0]$.
- Если $D[i, j] = D[i - 1, j] + 1$, то найдётся оптимальное выравнивание, последним столбцом которого является удаление.
- Если $D[i, j] = D[i, j - 1] + 1$, то найдётся оптимальное выравнивание, последним столбцом которого является вставка.
- Если $D[i, j] = D[i - 1, j - 1] + \text{diff}(A[i], B[j])$, то найдётся оптимальное выравнивание, последним столбцом которого является замена/несоответствие (если $A[i] \neq B[j]$) или соответствие (если $A[i] = B[j]$).

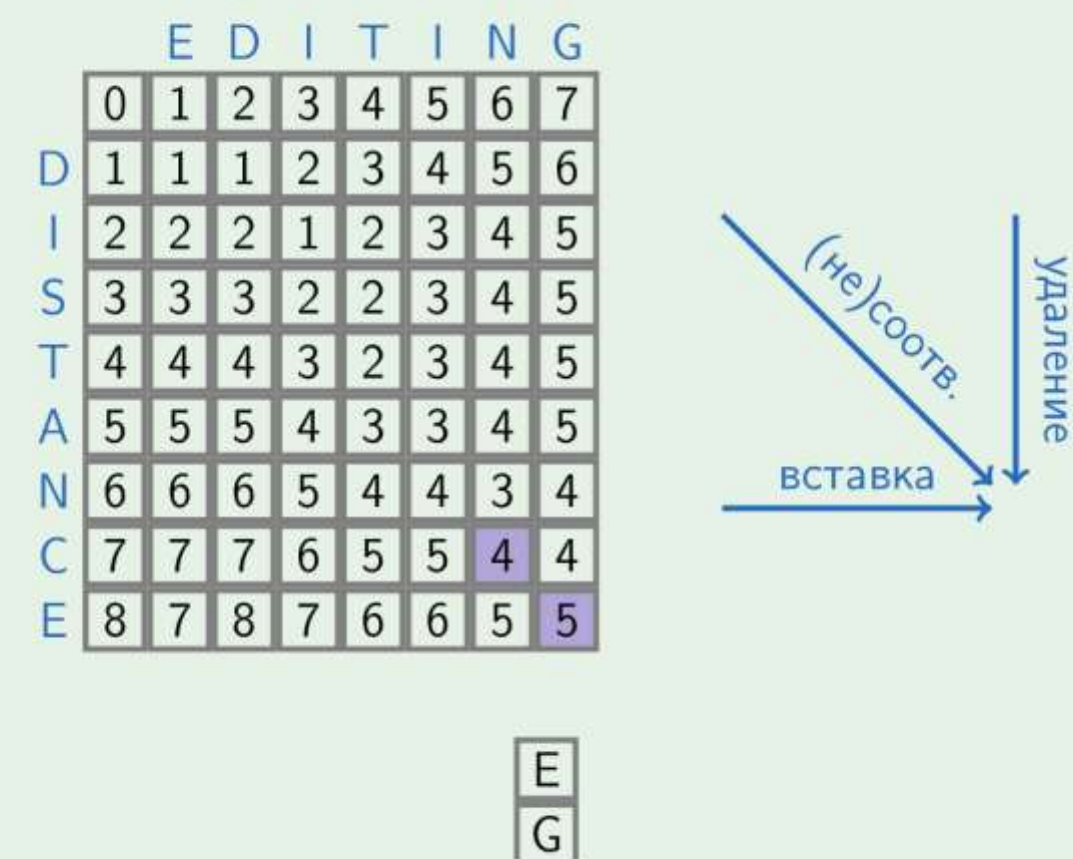
PISLO8. Задача о расстоянии Левенштейна.

Пример

		E	D	I	T	I	N	G	
		0	1	2	3	4	5	6	7
D		1	1	1	2	3	4	5	6
I		2	2	2	1	2	3	4	5
S		3	3	3	2	2	3	4	5
T		4	4	4	3	2	3	4	5
A		5	5	5	4	3	3	4	5
N		6	6	6	5	4	4	3	4
C		7	7	7	6	5	5	4	4
E		8	7	8	7	6	6	5	5

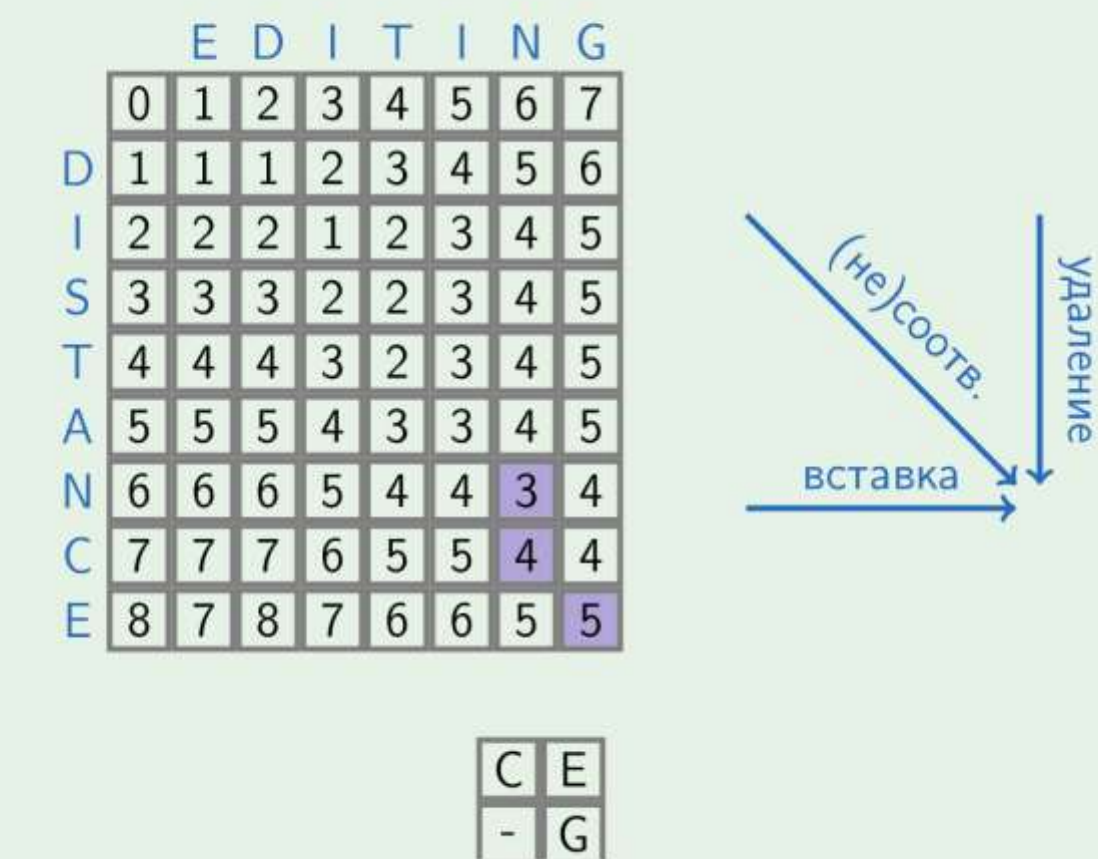
PISLO8. Задача о расстоянии Левенштейна.

Пример



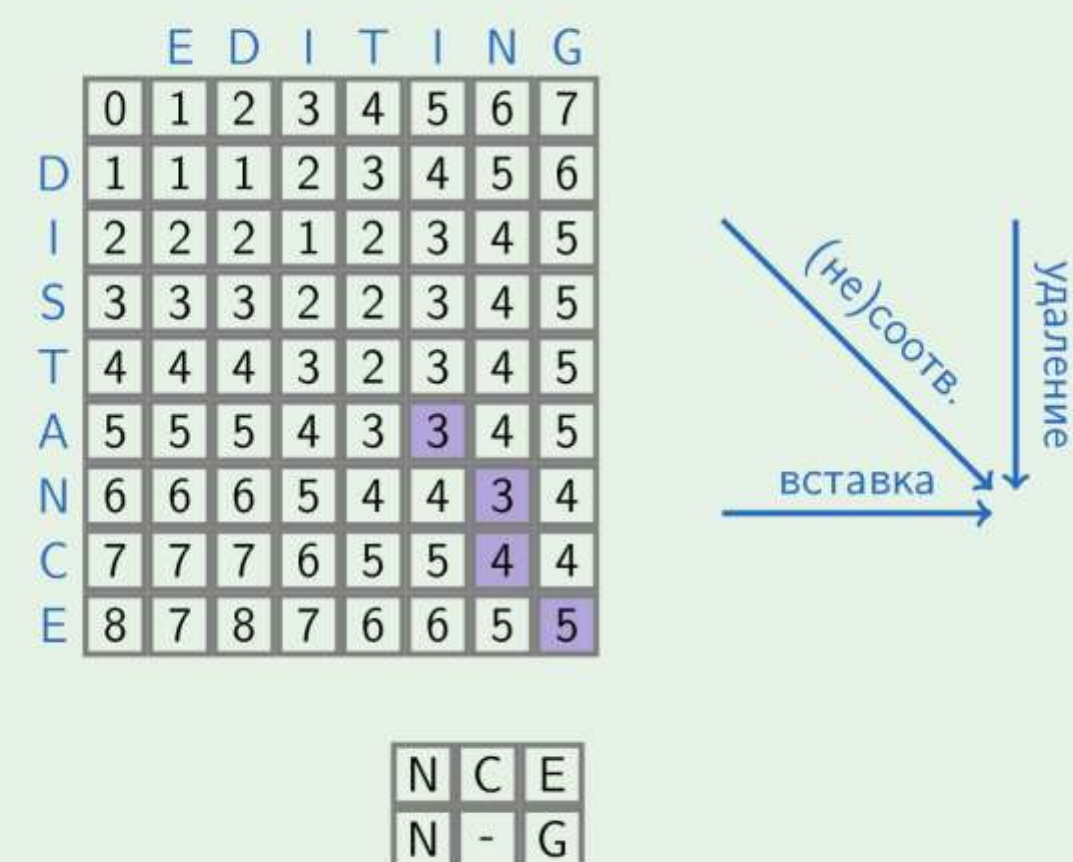
PISLO8. Задача о расстоянии Левенштейна.

Пример



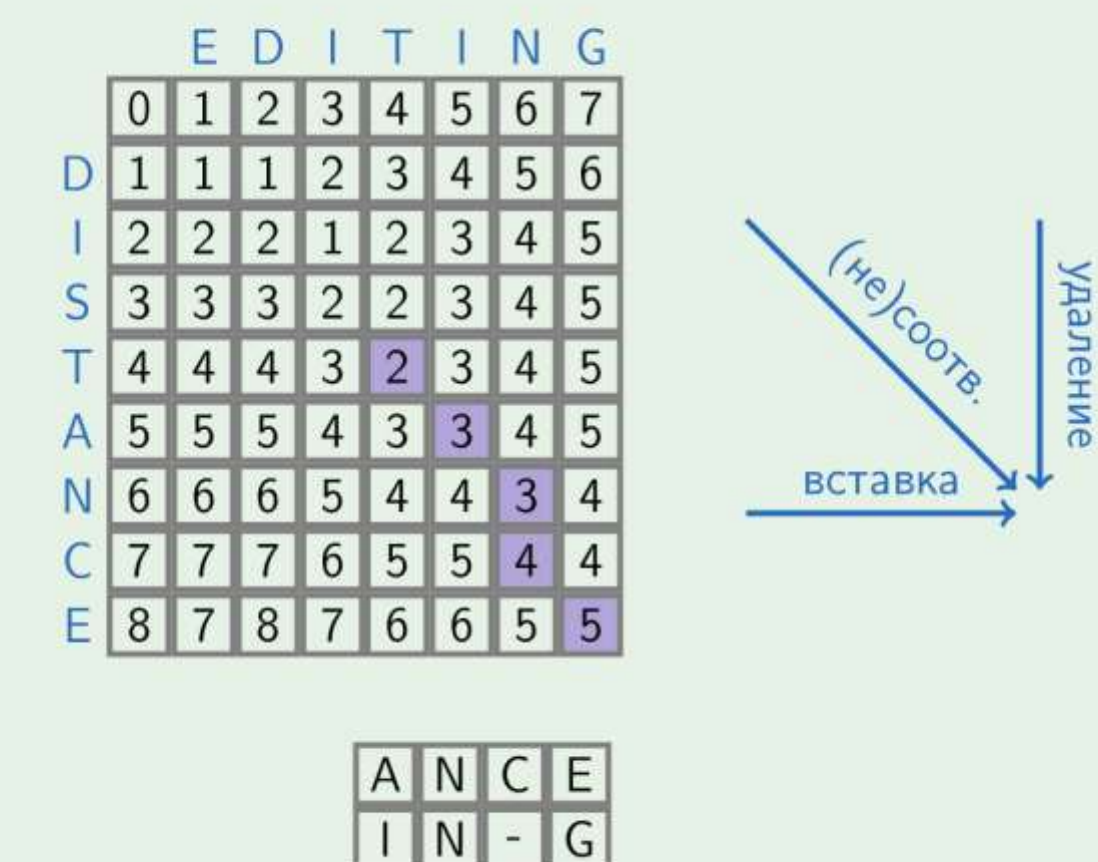
PISLO8. Задача о расстоянии Левенштейна.

Пример



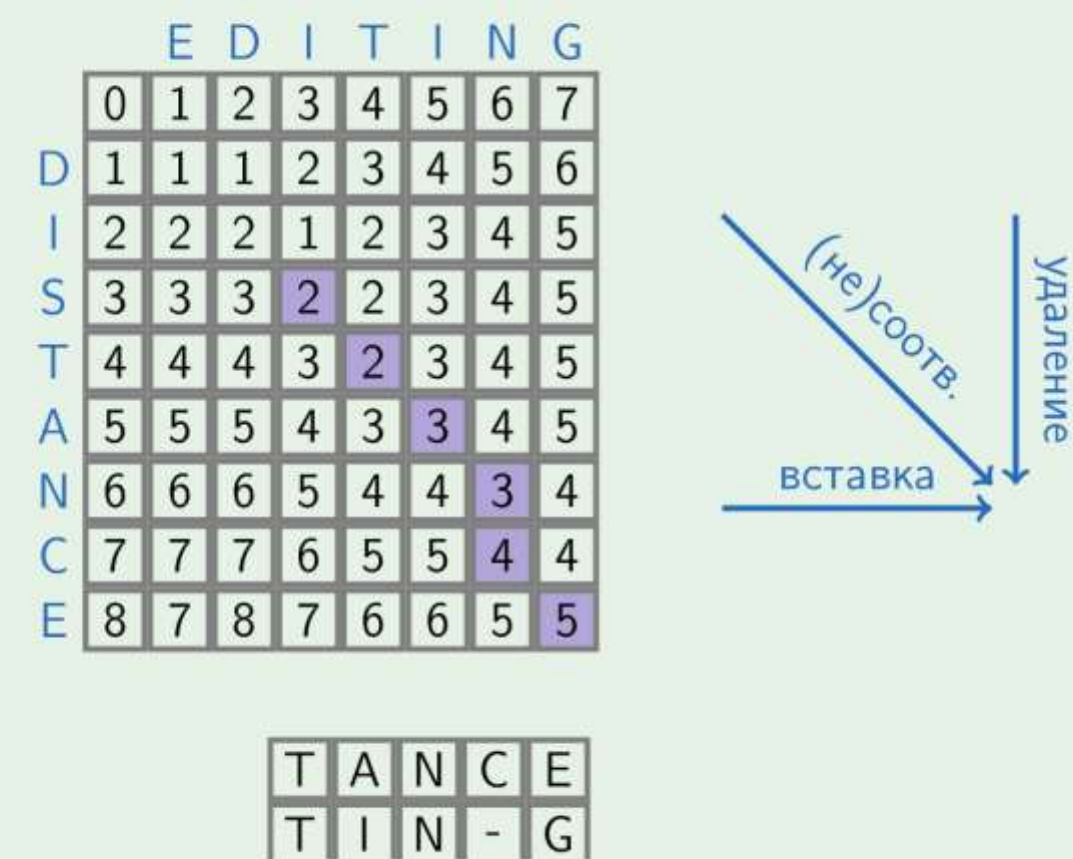
PISLO8. Задача о расстоянии Левенштейна.

Пример



PISLO8. Задача о расстоянии Левенштейна.

Пример

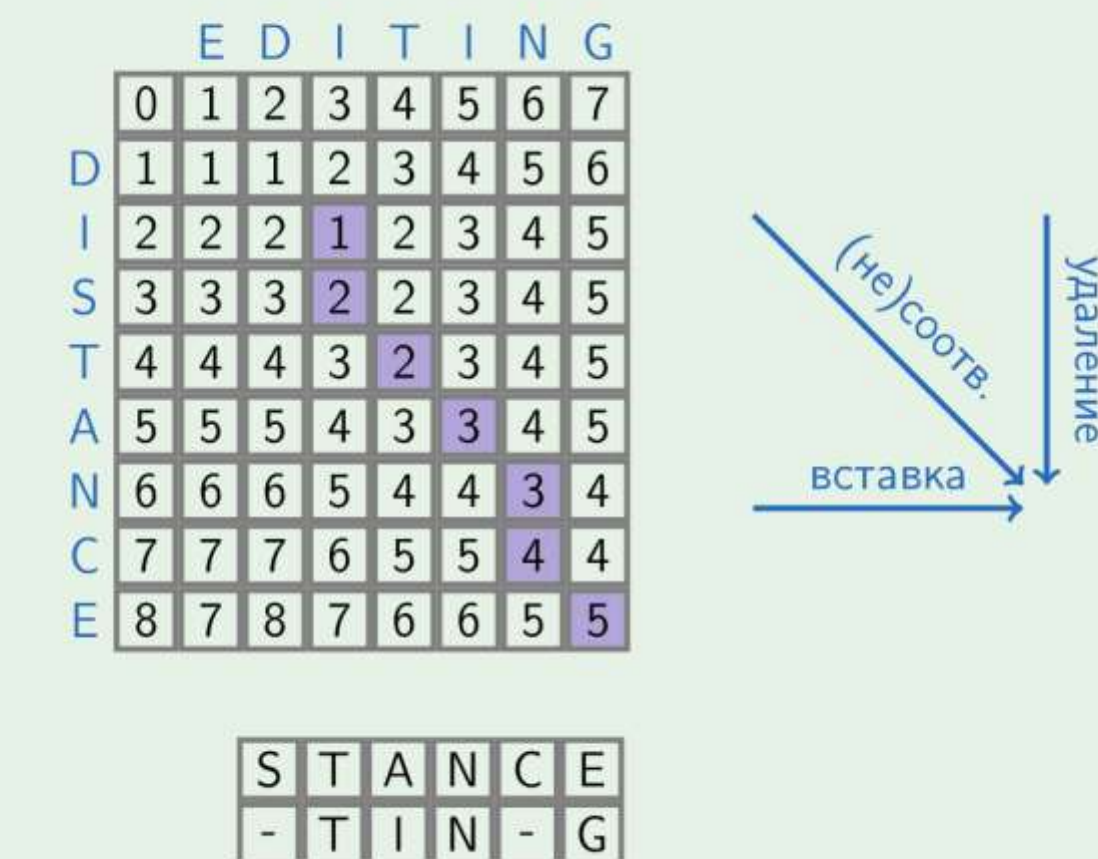


Кафедра экономической информатики. Бгуир. 2017

37

PISLO8. Задача о расстоянии Левенштейна.

Пример

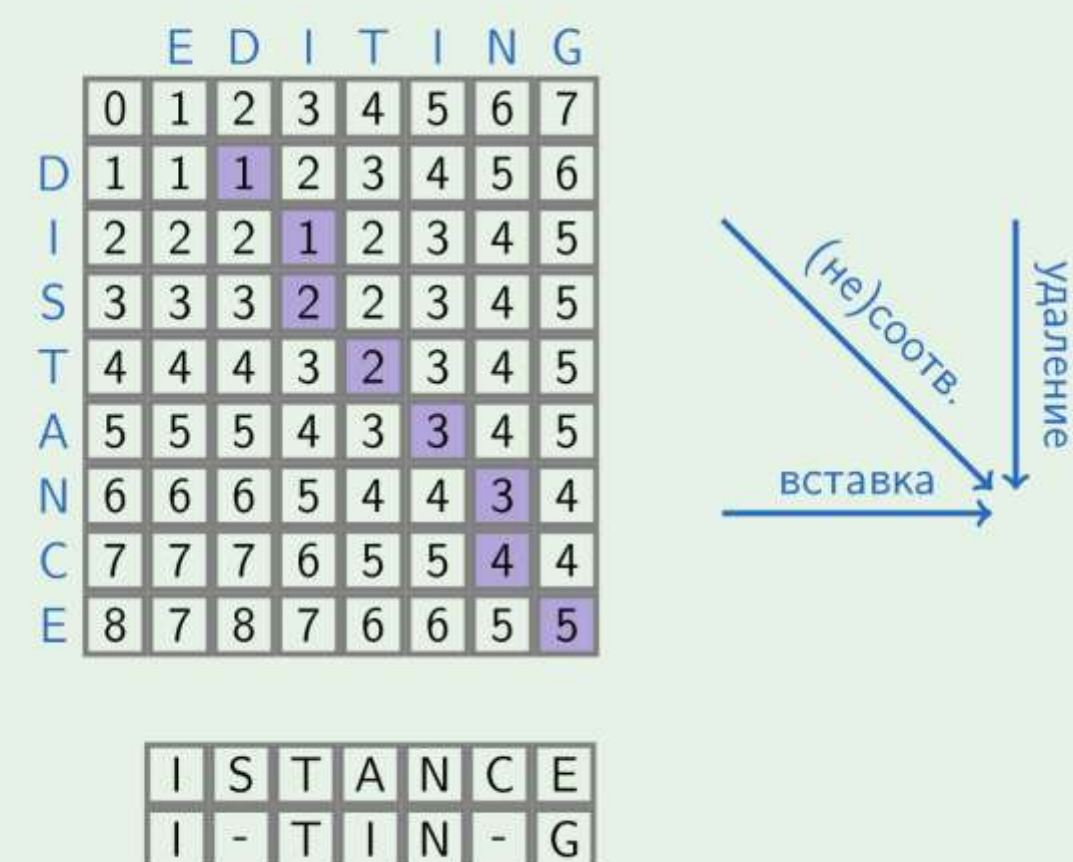


Кафедра экономической информатики. Бгуир. 2017

38

PISLO8. Задача о расстоянии Левенштейна.

Пример

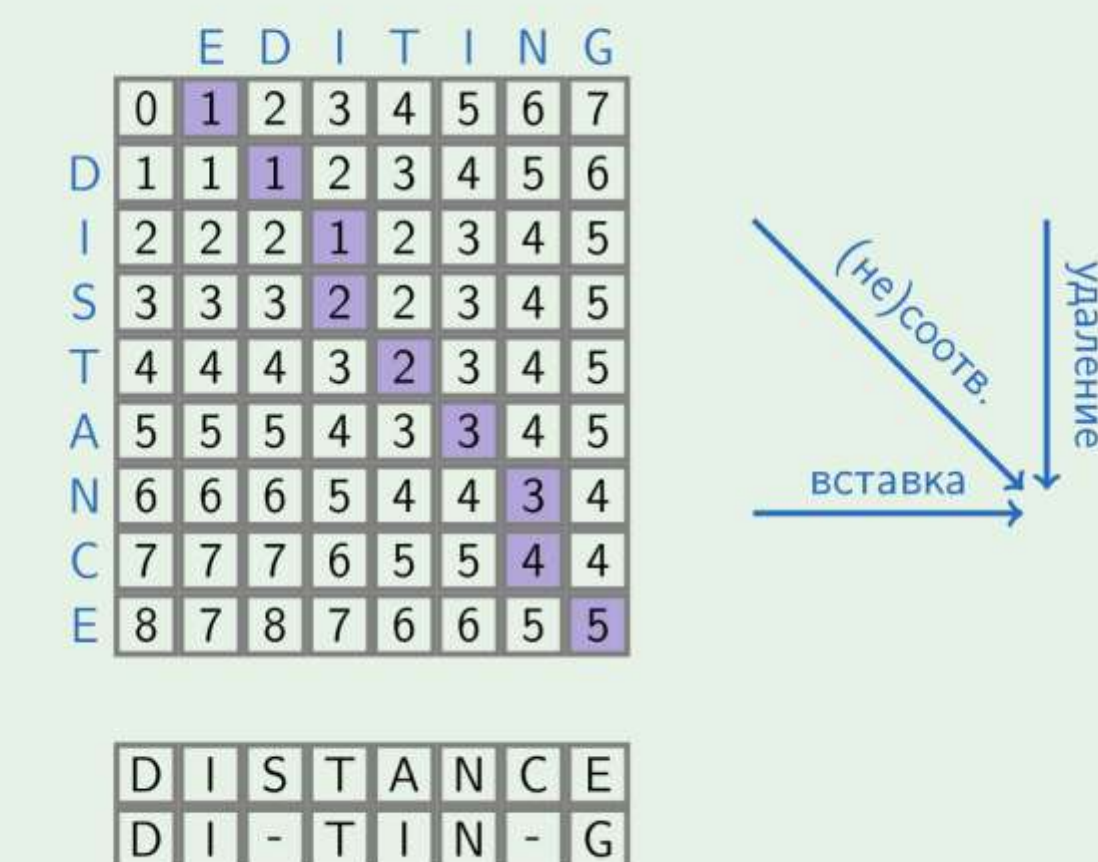


Кафедра экономической информатики. Бгуир. 2017

39

PISLO8. Задача о расстоянии Левенштейна.

Пример



Кафедра экономической информатики. Бгуир. 2017

40

PISLO8. Задача о расстоянии Левенштейна.

Пример

		E	D	I	T	I	N	G
D	0	1	2	3	4	5	6	7
I	1	1	1	2	3	4	5	6
S	2	2	2	1	2	3	4	5
T	3	3	3	2	2	3	4	5
A	4	4	4	3	2	3	4	5
N	5	5	5	4	3	3	4	5
C	6	6	6	5	4	4	3	4
E	7	7	7	6	5	5	4	4
	8	7	8	7	6	6	5	5

(не)соотв.

удаление

вставка

-	D	I	S	T	A	N	C	E
E	D	I	-	T	I	N	-	G

Кафедра экономической информатики. Бгуир. 2017

41

PISLO8. Задача о расстоянии Левенштейна.

Уменьшение используемой памяти

- При заполнении матрицы достаточно хранить только текущую и предыдущую строки (или столбцы).

Кафедра экономической информатики. Бгуир. 2017

42

PISLO8. Задача о расстоянии Левенштейна.

Уменьшение используемой памяти

- При заполнении матрицы достаточно хранить только текущую и предыдущую строки (или столбцы).
- Поэтому расстояние редактирования строк $A[1 \dots n]$ и $B[1 \dots m]$ можно вычислить за время $O(nm)$ с памятью $O(\min\{n, m\})$.

Кафедра экономической информатики. Бгуир. 2017

43

PISLO8. Задача о расстоянии Левенштейна.

Уменьшение используемой памяти

- При заполнении матрицы достаточно хранить только текущую и предыдущую строки (или столбцы).
- Поэтому расстояние редактирования строк $A[1 \dots n]$ и $B[1 \dots m]$ можно вычислить за время $O(nm)$ с памятью $O(\min\{n, m\})$.
- Однако для восстановления оптимального выравнивания нужна вся таблица D .

Кафедра экономической информатики. Бгуир. 2017

44

PISLO8. Задача о расстоянии Левенштейна.

Уменьшение используемой памяти

- При заполнении матрицы достаточно хранить только текущую и предыдущую строки (или столбцы).
- Поэтому расстояние редактирования строк $A[1 \dots n]$ и $B[1 \dots m]$ можно вычислить за время $O(nm)$ с памятью $O(\min\{n, m\})$.
- Однако для восстановления оптимального выравнивания нужна вся таблица D .
- Алгоритм Хиршберга находит оптимальное выравнивание за время $O(nm)$ с памятью $O(\min\{n, m\})$.

PISLO8. Задача о расстоянии Левенштейна.

Взвешенное расстояние редактирования

- Стоимости вставок, удалений и замен могут и различаться.
- Проверка правописания: некоторые замены символов более вероятны, чем другие.
- Биология: некоторые мутации более вероятны, чем другие.

PISLO8. Задача о расстоянии Левенштейна.

Обобщённое рекуррентное соотношение

$$D[i, j] = \min \{ D[i, j-1] + \text{inscost}(B[j]), \\ D[i-1, j] + \text{delcost}(A[i]), \\ D[i-1, j-1] + \text{substcost}(A[i], B[j]) \}$$

PISLO8. Задача о расстоянии Левенштейна.

Заключение

- Проанализировали структуру оптимального решения, чтобы определить подзадачи и рекуррентное соотношение на них.
- Записали рекурсивный алгоритм (сверху вниз) по данному соотношению.
- Доказали верхнюю оценку на время работы, проанализировав суммарное число рекурсивных вызовов.
- Переделали рекурсивный алгоритм в итеративный (снизу вверх), заполняющий таблицу непосредственно.
- Проанализировали структуру таблицы, чтобы сэкономить память.

Задание А.

Задача на программирование: расстояние Левенштейна
https://ru.wikipedia.org/wiki/Расстояние_Левенштейна
<http://planetcalc.ru/1721/>

Дано:
 Две данных непустые строки длины не более 100, содержащие строчные буквы латинского алфавита.

Необходимо:
 Решить задачу МЕТОДАМИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ
 Рекурсивно вычислить расстояние редактирования двух данных непустых строк

Sample Input 1:
 ab
 ab
 Sample Output 1:
 0

Sample Input 2:
 short
 ports
 Sample Output 2:
 3

Sample Input 3:
 distance
 editing
 Sample Output 2:
 5

Кафедра экономической информатики. Бгуир. 2017

49

Задание Б.

Задача на программирование: расстояние Левенштейна
https://ru.wikipedia.org/wiki/Расстояние_Левенштейна
<http://planetcalc.ru/1721/>

Дано:
 Две данных непустые строки длины не более 100, содержащие строчные буквы латинского алфавита.

Необходимо:
 Решить задачу МЕТОДАМИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ
 Итерационно вычислить расстояние редактирования двух данных непустых строк

Sample Input 1:
 ab
 ab
 Sample Output 1:
 0

Sample Input 2:
 short
 ports
 Sample Output 2:
 3

Sample Input 3:
 distance
 editing
 Sample Output 2:
 5

Кафедра экономической информатики. Бгуир. 2017

50

Задание С.

Задача на программирование: расстояние Левенштейна
https://ru.wikipedia.org/wiki/Расстояние_Левенштейна
<http://planetcalc.ru/1721/>

Дано:
 Две данных непустые строки длины не более 100, содержащие строчные буквы латинского алфавита.

Необходимо:
 Решить задачу МЕТОДАМИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ
 Итерационно вычислить алгоритм преобразования двух данных непустых строк
 Вывести через запятую редакционное предписание в формате:
 операция ("+" вставка, "-" удаление, "~" замена, "#" копирование)
 символ замены или вставки

Sample Input 1:
 ab
 ab
 Sample Output 1:
 #, #,

Sample Input 2:
 short
 ports
 Sample Output 2:
 ~p, -h, #, #, 5+s,

Sample Input 3:
 distance
 editing
 Sample Output 2:
 +e, #, #, -, #, ~i, #, -c, ~g,

P.S. В литературе обычно действия редакционных предписаний обозначаются так:
 - D (англ. delete) — удалить,
 + I (англ. insert) — вставить,
 ~ R (replace) — заменить,
 # M (match) — совпадение.

Кафедра экономической информатики. Бгуир. 2017

51