Teoría de Algoritmos - Problema de la transformación de cadenas
Autor: Luis Quesada Torres
Para los alumnos de la academia Dos Motivos, verano del 2010.

Enunciado:
Sean u y v dos cadenas de caracteres. Se desea transformar u en v con el mínimo número de operaciones básicas del tipo siguiente: eliminar un carácter, añadir un carácter y cambiar un carácter. Por ejemplo, podemos pasar de **abbac** a **abcbc** en tres pasos:
abbac → abac (eliminamos b en la posición 3 (contando desde 0)).
abac → ababc (añadimos b en la posición 4).
ababc → abcbc (cambiamos a en la posición 3 por c).

Sin embargo, esta transformación no es óptima. Lo que queremos en este caso es diseñar un algoritmo que calcule el número mínimo de operaciones, de estos tres tipos, necesarias para transformar u en v y cuáles son esas operaciones, estudiando su complejidad en función de las longitudes de u y v.

Ejemplo: Si queremos pasar de **abbacd** a **babacba**, tenemos la solución óptima con 4 pasos:
abbacd → bbacd (eliminamos a en la posición 1).
bbacd → babacd (insertamos a en la posición 1).
babacd → babacb (cambiamos d en la posición 6 por b).
babacb → babacba (insertamos a en la posición 7).

Solución:

En primer lugar, vamos a ver si podría resolverse aplicando un greedy: Para ello, deberían darse dos condiciones:

- El problema debería cumplir el principio de optimalidad de Bellman, que enuncia "cada trozo de la solución óptima es a su vez una solución óptima para un problema más pequeñito".
  Esto se cumple, ya que por ejemplo, si la solución óptima para pasar de **abbacd** a **babacba** pasa por:
  abbacd → bbacd → babacd → babacb → babacba
  La solución óptima para ir de **bbacd** a **babacb** (que son dos subproblemas de este problema general), siempre serán los mismos dos pasos que hemos seguido en el problema general
- Además, no debería de existir ningún tipo de compromiso entre valor y libertad a la hora de hacer la decisión. En este caso, cada opción que tomamos te da una libertad distinta (cambiar un elemento por otro en lugar de eliminarlo te deja por delante más elementos que podrías emparejar con los siguientes). Esto se observa a la hora de intentar demostrar por inducción que el algoritmo greedy da el óptimo:
  Para i=1
  Suponemos que hemos realizado la mejor decisión posible (en base a qué criterio? Cambiar? Sustituir? Insertar? No lo sabemos a priori!!)
  Si hubiera una mejor, ¿la habríamos escogido? Pues puede, porque si hemos escogido cambiar, a lo mejor la mejor era sustituir, y viceversa.

Por esto, este problema no puede resolverse por greedy.

No obstante, como hemos demostrado que se cumple el principio de optimalidad de Bellman, sí podemos aplicar programación dinámica.

Para ello, si tenemos dos cadenas, A y B, vamos a definir la función OP(m,n) que devolverá el número de operaciones necesarias para convertir la cadena A hasta el carácter m en la cadena B hasta el carácter n. Esta función actuará sobre el último carácter de A (el carácter m).

Las opciones para un problema con m y n cualesquiera serían:

- Transformar A(m-1) en B(n) (recursivamente!) y borrar el carácter am.
  Ejemplo:
  A: abca (m=4) B: abc (n=3)
  Como ya hemos transformado A(m-1) (que es **abc**) en B(n) (que es **abc**), sólo tenemos que borrar am (la **a**).

  Por tanto, una opción queda así: OP(m,n) = 1+OP(m-1,n) si eliminamos posición m de A.

  Obsérvese que sumamos uno porque estamos haciendo una operación más!

- Transformar A(m) en B(n-1) (recursivamente!) e insertar el carácter bn tras la posición m de A.
  Ejemplo:
  A: abc (m=3) B: abcd (n=4)
  Como ya hemos transformadoA(m) (que es **abc**) en B(n-1) (que es **abc**), insertamos bn (la **d**) tras m de A (al final).

  Por tanto, otra opción queda así: OP(m,n) = 1+OP(m,n-1) si insertamos bn en posición m de A.

  Obsérvese que sumamos uno porque estamos haciendo una operación más!

- Transformar A(m-1) en B(n-1) (recursivamente!) y sustituir el carácter am por bn.
  Ejemplo:
  A: abce (m=4) B: abcf (n=4)
  Como ya hemos transformadoA(m-1) (que es **abc**) en B(n-1) (que es **abc**), cambiamos am (la **e**) por bn (la **f**).

  Por tanto, otra opción queda así: OP(m,n) = 1+OP(m-1,n-1) si cambiamos am por bn

  Obsérvese que sumamos uno porque estamos haciendo una operación más!

- Transformar A(m-1) en B(n-1) (recursivamente!) y si am coincide con bn, no hacemos cambio.
  Ejemplo:
  A: abce (m=4) B: abce (n=4)
  Como ya hemos transformadoA(m-1) (que es **abc**) en B(n-1) (que es **abc**), y am=bn, no hacemos nada.

  Por tanto, otra opción queda así: OP(m,n) = OP(m-1,n-1) si am=bn

  Obsérvese que en este caso tenemos que añadir como restricción que am sea igual que bn.

Además, tendremos como condiciones iniciales los casos en los que una cadena está vacía y la otra tiene elementos.

- Si A tiene m elementos y B tiene 0 elementos, tendremos que eliminar los m elementos de A (m operaciones).
  Por tanto, para OP(m,0) = m

- Si A tiene 0 elementos y B tiene m elementos, tendremos que insertar los n elementos de B en A (n operaciones).
  Por tanto, para OP(0,n) = n

Al final, la expresión recurrente y los casos base quedan así:

$$OP(m,n) = \min \begin{cases} 1+OP(m-1,n) & \text{si eliminamos posición m de A.} \\ 1+OP(m,n-1) & \text{si insertamos bn tras posición m de A.} \\ 1+OP(m-1,n-1) & \text{si cambiamos am por bn} \\ OP(m-1,n-1) & \text{si am=bn} \\ m & \text{si n=0} \\ n & \text{si m=0} \end{cases}$$

Por tanto, la tabla, para una cadena A **abbacd** (m=6) y una cadena B **babacba** (m=7), inicialmente, se vería así:

B\A	m=0	m=1	m=2	m=3	m=4	m=5
n=0	0	1	2	3	4	5
n=1	1					
n=2	2					
n=3	3					
n=4	4					
n=5	5					
n=6	6					
n=7	7					

La tabla, tras rellenarse quedaría así (por comodidad, pongo en las cabeceras de la tabla la letra de la cadena que corresponde al índice m o n en cuestión):

B\A	m=0	**a** m=1	**b** m=2	**b** m=3	**a** m=4	**c** m=5	**d** m=6
n=0	0	1	2	3	4	5	6
b n=1	1	1	1	2	3	4	5
a n=2	2	1	2	2	2	3	4
b n=3	3	2	1	2	3	3	4
a n=4	4	3	2	2	2	3	4
c n=5	5	4	3	3	3	2	3
b n=6	6	5	4	3	4	3	3
a n=7	7	6	5	4	3	4	4

Este es el procedimiento seguido para rellenarla. Empezamos haciendo un par a mano:

Para m = 1 y n = 1, tenemos am = a y bn = b.

$$OP(1,1) = \min \begin{cases} 1+OP(0,1) & \text{si eliminamos posición m de A.} & = 1+1 = 2 \\ 1+OP(1,0) & \text{si insertamos bn tras posición m de A.} & = 1+1 = 2 \\ 1+OP(0,0) & \text{si cambiamos am por bn} & = 1+0 = 1 \\ OP(0,0) & \text{si am=bn} & \text{NO CUMPLE RESTRICCIÓN} \end{cases}$$

El mínimo es 1, por lo que OP(1,1) = 1.

Para m = 2 y n = 1, tenemos am = b y bn = b.

$$OP(2,1) = \min \begin{cases} 1+OP(1,1) & \text{si eliminamos posición m de A.} & = 1+1 = 2 \\ 1+OP(2,0) & \text{si insertamos bn tras posición m de A.} & = 1+2 = 3 \\ 1+OP(1,0) & \text{si cambiamos am por bn} & = 1+1 = 2 \\ OP(1,0) & \text{si am=bn} & = 1 \end{cases}$$

El mínimo es 1, por lo que OP(2,1) = 1.

Observamos las distintas posibilidades gráficamente sobre la tabla, para rellenar la casilla siguiente escogeremos el mínimo de las distintas posibilidades:

1. Si eliminamos: Viene desde la casilla que hay justo a la izquierda, sumamos 1 a lo que hay ahí.
2. Si insertamos: Viene desde la casilla que hay justo encima, sumamos 1 a lo que hay ahí.
3. Si cambiamos: Viene desde la casilla que hay justo a la izquierda y arriba, sumamos 1 a lo que hay ahí.
4. Si mantenemos (sólo si am=bn!): Viene desde la casilla que hay justo a la izquierda y arriba, sin sumar nada.

Para OP(3,1), como la opción 4 es posible podemos escoger min{1+1,3+1,2+1,2+0}, o sea, min{2,4,3,2} = 2.

B\A	m=0	**a** m=1	**b** m=2	**b** m=3	**a** m=4
n=0	0	1	2 ca +1	3 in +1	4
b n=1	1	1	1 ma +0 (am=bn) el +1		
a n=2	2				

Así pues, el resto de la tabla la he rellenado siguiéndola de forma gráfica.

Ahora, a partir de la tabla rellena queremos obtener la solución:

B\A	m=0	**a** m=1	**b** m=2	**b** m=3	**a** m=4	**c** m=5	**d** m=6
n=0	0	1	2	3	4	5	6
b n=1	1	1	1	2	3	4	5
a n=2	2	1	2	2	2	3	4
b n=3	3	2	1	2	3	3	4
a n=4	4	3	2	2	2	3	4
c n=5	5	4	3	3	3	2	3
b n=6	6	5	4	3	4	3	3
a n=7	7	6	5	4	3	4	4

Para ello, desde el final (OP(7,6)) aplicamos la recurrencia y vemos qué opción se tomó:

Para m = 7 y n = 6, tenemos am = d y bn = a.

$$OP(7,6) = \min \begin{cases} 1+OP(6,6) & \text{si eliminamos posición m de A.} & = 1+3 = 4 \\ 1+OP(7,5) & \text{si insertamos tras en posición m de A.} & = 1+4 = 5 \\ 1+OP(6,5) & \text{si cambiamos am por bn} & = 1+3 = 4 \\ OP(6,5) & \text{si am=bn} & \text{NO CUMPLE RESTRICCIÓN} \end{cases}$$

El mínimo conocido, es decir, el valor en la tabla para OP(7,6) es 4, por lo que nos valen cualquiera de las dos opciones que dan un 4 (obsérvese que llegamos a secuencias de decisiones distintas que transforman en el mismo número de pasos a la cadena A en la cadena B). Bien, pues si escogemos la decisión de eliminar, la última decisión tomada sería que hemos eliminado la posición m de A, es decir, la posición 6 de A.

Posteriormente, seguiríamos resolviendo hacia atrás la recurrencia a partir de la llamada de esa opción: OP(6,6):

Para m = 6 y n = 6, tenemos am = d y bn = b.

$$OP(6,6) = \min \begin{cases} 1+OP(5,6) & \text{si eliminamos posición m de A.} & = 1+3 = 4 \\ 1+OP(6,5) & \text{si insertamos bn tras posición m de A.} & = 1+3 = 4 \\ 1+OP(5,5) & \text{si cambiamos am por bn} & = 1+2 = 3 \\ OP(5,5) & \text{si am=bn} & \text{NO CUMPLE RESTRICCIÓN} \end{cases}$$

El mínimo conocido, es decir, el valor en la tabla para OP(6,6) es 3, por lo que nos quedamos con la opción que vale 3, es decir, la penúltima decisión sería cambiar am (que es una **b**) por bn (que es una **d**), es decir, cambiar la posición m por una **d**. Seguiríamos resolviendo a partir de OP(5,5), que es la expresión que aparece en esa opción.

Podemos observar gráficamente en la tabla, análogamente a la observación anterior:

B\A	m=0	**a** m=1	**b** m=2	**b** m=3	**a** m=4
n=0	0	1	2 ca +1	3 in +1	4
b n=1	1	1	1 ma +0 (am=bn) el +1	2	3
a n=2	2	1	2	2	2

En este caso buscamos que nos coincida el valor de la tabla con el de una de esas opciones que es la que seguiríamos.

Obsérvese que tenemos que tener en cuenta en el caso de seguir la opción de mantener (sin sumar) que am=bn.

Para extraer la solución, uno de los recorridos hacia atrás completo en la tabla nos quedaría así:

B\A	M=0	**a** m=1	**b** m=2	**b** m=3	**a** m=4	**c** m=5	**d** m=6
n=0	0	1	2	3	4	5	6
b n=1	1	1	1	2	3	4	5
a n=2	2	1	2	2	2	3	4
b n=3	3	2	1	2	3	3	4
a n=4	4	3	2	2	2	3	4
c n=5	5	4	3	3	3	2	3
b n=6	6	5	4	3	4	3	3
a n=7	7	6	5	4	3	4	4

(Aunque se podrían sacar otros recorridos escogiendo otras opciones cuando pueden escogerse de entre varias)

Desde final a principio (en el orden en el que las sacamos) las decisiones serían (no muestro las de mantener):
Cambiar a6 por “a” (que es b7).
Insertar “b” tras a5.
Cambiar a2 por “a” (que es b2)
Cambiar a1 por “b” (que es b1)

Comprobamos:

abbacd	→ abbaca	(Cambiar a6 por “a” (que es b7)).
abbaca	→ abbacba	(Insertar “b” tras a5).
abbacba	→ aabacba	(Cambiar a2 por “a” (que es b2))
aabacba	→ babacba	(Cambiar a1 por “b” (que es b1))

Hemos llegado a la cadena objetivo.

¿Hace falta ordenar las decisiones de principio a fin? En este caso tendríamos que tener cuidado si lo hacemos. Mirad:
Si quito las decisiones de mantener y ordeno de principio a fin:
Cambiar a1 por “b” (que es b1)
Cambiar a2 por “a” (que es b2)
Insertar “b” tras a5.
Cambiar a6 por “a” (que es b7).

Comprobamos:

abbacd	→ bbbacd	(Cambiar a1 por “b” (que es b1))
bbbacd	→ babacd	(Cambiar a2 por “a” (que es b2))
babacd	→ babacbd	(Insertar “b” tras a5).
babacbd	→ babacba	(Cambiar a6 por “a” (que es b7)). **(ojo, el a6 de ahora está avanzado uno por la inserción, así que realmente es a7!! Habría que tener cuidado!!).**

Hemos llegado a la cadena objetivo.