ALGORITMIA

Grado en Ingeniería Informática del Software - Curso 2021-2022

GUION DE LA PRÁCTICA 5

OBJETIVO:

Programación dinámica: Cálculo de la distancia de Levenshtein

DISTANCIA DE LEVENSHTEIN

Se llama distancia de Levenshtein (Vladimir Levenshtein; Rusia; 1935-2017) entre dos strings **cad1** y **cad2** al número mínimo de operaciones que hay que hacer para que ambos strings sean iguales.

Las operaciones permitidas son: **Insertar** un carácter, **Borrar** un carácter o **Cambiar** un carácter por otro.

Es **aconsejable** ver en la red (Wikipedia y otras) información acerca de esa distancia, para ver p.e., las propiedades de esa distancia (simétrica y otras) y la importancia de la buena resolución de ese problema (y otras variantes del mismo problema), por su gran utilidad práctica en muchos campos.

Ejemplo:

cad1= "BARCAZAS"

cad2= "ABRACADABRA"

***distanciaLevenshtein** (cad1,cad2) es igual a 7*

Es 7, porque es el mínimo número de operaciones, de las permitidas, para pasar de **cad1** a **cad2**, o bien, de **cad2** a **cad1**.

SOLUCIÓN RECURSIVA

En 1965, Levenshtein formuló (**d es distanciaLevenshtein**):

d(i,j)= d(i-1,j-1) **si cad1[i] == cad2[j]**

d(i,j)= 1 + min (d[i-1,j-1],d[i,j-1],d[i-1,j]) **si cad1[i] != cad2[j]**

Se cumple que:

*el mínimo por la vía d[i-1,j-1] supone **cambiar** quitando el carácter j-ésimo y poniendo el carácter i-ésimo para llegar de **cad1** a **cad2,** o bien, **cambiar** quitando el carácter i-ésimo y poniendo el carácter j-ésimo para llegar de **cad2** a **cad1**.

*el mínimo por la vía d[i,j-1] supone **borrar** el carácter j-ésimo para llegar de **cad1** a **cad2,** o bien, **insertar** el carácter j-ésimo para llegar de **cad2** a **cad1**.

*el mínimo por la vía d[i-1,j] supone **insertar** el carácter i-ésimo para llegar de **cad1** a **cad2,** o bien, **borrar** el carácter i-ésimo para llegar de **cad2** a **cad1**.

En Programación Dinámica hay que disponer de una matriz de **n** (len(cad1)+1) filas y **m** (len(cad2)+1) columnas y a partir de la inicialización trivial de la fila 0 y la columna 0 (en color rojo) , se iría dando valores (por filas o por columnas) a los otros elementos de la matriz según las fórmulas recurrentes antes vistas (en color verde).

TRAZA CON cad1=”BARCAZAS” y cad2=”ABRACADABRA”:

			B	A	R	C	A	Z	A	S
		j=*0*	j=*1*	j=*2*	j=*3*	j=*4*	j=*5*	j=*6*	j=*7*	j=*8*
	i=*0*	0	1	2	3	4	5	6	7	8
A	i=*1*	1	1	1	2	3	4	5	6	7
B	i=*2*	2	1	2	2	3	4	5	6	7
R	i=*3*	3	2	2	2	3	4	5	6	7
A	i=*4*	4	3	2	3	3	3	4	5	6
C	i=*5*	5	4	3	3	3	4	4	5	6
A	i=*6*	6	5	4	4	4	3	4	4	5
D	i=*7*	7	6	5	5	5	4	4	5	5
A	i=*8*	8	7	6	6	6	5	5	4	5
B	i=*9*	9	8	7	7	7	6	6	5	5
R	i=*10*	10	9	8	7	8	7	7	6	6
A	i=*11*	11	10	9	8	8	8	8	7	**7 (SOL)**

SE PIDE:

A) Analizar la **Complejidad Temporal** del algoritmo.

B) **Implementar** el algoritmo de Programación Dinámica, probando su buen funcionamiento para otros ejemplos.

C) **Medir tiempos** poniendo ambas longitudes iguales (luego n=m), dando valor a las cadenas de forma aleatoria y creciendo el tamaño de problema así (n= 100, 200, 400, 800, 1600, … etc). Hay que rellenar la tabla de tiempos correspondiente y razonar si cumple o no la Complejidad hallada.

TRABAJOS OPCIONALES:

Puede realizar uno o más de los tres trabajos opcionales que siguen.
Se puede entregar más adelante, junto con alguna de las prácticas posteriores.

TRABAJO1

Calcular **cualquiera** de las formas que hay de hacer ese mínimo de operaciones para lograr una cadena de la otra y la otra de la una.

Observe como los valores subrayados indican de qué valor anterior se obtuvo cada valor[i,j] y se calculan procesándolos **hacia atrás** **(a partir de la Solución)** una vez rellenada toda la matriz. Pues bien, esa trayectoria muestra una forma posible para hacer iguales las dos cadenas.

Puede haber varias rutas posibles (en este caso las hay) y hay que coger cualquiera de ellas.

TRAZA CON cad1="BARCAZAS" y cad2="ABRACADABRA":

			B	A	R	C	A	Z	A	S
		j=*0*	j=*1*	j=*2*	j=*3*	j=*4*	j=*5*	j=*6*	j=*7*	j=*8*
	i=*0*	**0**	1	2	3	4	5	6	7	8
A	i=*1*	**1**	1	1	2	3	4	5	6	7
B	i=*2*	2	**1**	2	2	3	4	5	6	7
R	i=*3*	3	**2**	2	2	3	4	5	6	7
A	i=*4*	4	3	**2**	**3**	3	3	4	5	6
C	i=*5*	5	4	3	3	**3**	4	4	5	6
A	i=*6*	6	5	4	4	4	**3**	4	4	5
D	i=*7*	7	6	5	5	5	4	**4**	5	5
A	i=*8*	8	7	6	6	6	5	5	**4**	5
B	i=*9*	9	8	7	7	7	6	6	**5**	5
R	i=*10*	10	9	8	7	8	7	7	**6**	6
A	i=*11*	11	10	9	8	8	8	8	7	**7 (SOL)**

***Trayectoria: "BARCAZAS" a "ABRACADABRA"**:

0) = "BARCAZAS"
1)Insertar "A" = "ABARCAZAS"
1)Mantener "B" = "ABARCAZAS"
2)Insertar "R" = "ABRARCAZAS"
2)Mantener "A" = "ABRARCAZAS"
3)Borrar "R" = "ABRACAZAS"
3)Mantener "C" = "ABRACAZAS"
3)Mantener "A" = "ABRACAZAS"
4)Cambiar "Z" por "D" = "ABRACADAS"
4)Mantener "A" = "ABRACADAS"
5)Insertar "B" = "ABRACADABS"
6)Insertar "R" = "ABRACADABRS
7)Cambiar "S" por "A" = "ABRACADABRA"

***Trayectoria: "ABRACADABRA" a "BARCAZAS"**:

0) =	"ABRACADABRA"
1)Borrar "A" =	"BRACADABRA"
1)Mantener "B" =	"BRACADABRA"
2)Borrar "R" =	"BACADABRA"
2)Mantener "A" =	"BACADABRA"
3)Insertar "R" =	"BARCADABRA"
3)Mantener "C" =	"BARCADABRA"
3)Mantener "A" =	"BARCADABRA"
4)Cambiar "D" por "Z" =	"BARCAZABRA"
4)Mantener "A" =	"BARCAZABRA"
5)Borrar "B" =	"BARCAZARA"
6)Borrar "R" =	"BARCAZAA"
7)Cambiar "A" por "S" =	"BARCAZAS"

TRABAJO2

Calcular **todas** las formas que hay de hacer ese mínimo de operaciones para lograr una cadena de la otra.

Hay que ir explorando **hacia atrás** todas las rutas que hay para llegar desde la solución hasta el inicio y para cada una de ellas calcular la forma de igualar las dos cadenas.

Vamos a ver otra ruta posible, de las varias que hay en este ejemplo:

TRAZA CON cad1="BARCAZAS" y cad2="ABRACADABRA":

			B	A	R	C	A	Z	A	S
		j=*0*	j=*1*	j=*2*	j=*3*	j=*4*	j=*5*	j=*6*	j=*7*	j=*8*
	i=*0*	**0**	**1**	2	3	4	5	6	7	8
A	i=*1*	1	1	**1**	2	3	4	5	6	7
B	i=*2*	2	1	**2**	2	3	4	5	6	7
R	i=*3*	3	2	2	**2**	3	4	5	6	7
A	i=*4*	4	3	2	**3**	3	3	4	5	6
C	i=*5*	5	4	3	3	**3**	4	4	5	6
A	i=*6*	6	5	4	4	4	**3**	4	4	5
D	i=*7*	7	6	5	5	5	4	**4**	5	5
A	i=*8*	8	7	6	6	6	5	5	**4**	5
B	i=*9*	9	8	7	7	7	6	6	5	**5**
R	i=*10*	10	9	8	7	8	7	7	6	**6**
A	i=*11*	11	10	9	8	8	8	8	7	**7 (SOL)**

***Trayectoria: "BARCAZAS" a "ABRACADABRA"**:

0) =	"BARCAZAS"
1)Borrar "B" =	"ARCAZAS"
1)Mantener "A" =	"ARCAZAS"
2)Insertar "B" =	"ABRCAZAS"
2)Mantener "R" =	"ABRCAZAS"
3)Insertar "A" =	"ABRACAZAS"
3)Mantener "C" =	"ABRACAZAS"
3)Mantener "A" =	"ABRACAZAS"
4)Cambiar "Z" por "D" =	"ABRACADAS"
4)Mantener "A" =	"ABRACADAS"
5)Cambiar "S" por "B" =	"ABRACADAB"
6)Insertar "R" =	"ABRACADABR"
7)Insertar "A" =	"ABRACADABRA"

***Trayectoria: "ABRACADABRA" a "BARCAZAS"**:

0) =	"ABRACADABRA"
1)Insertar "B" =	"BABRACADABRA"
1)Mantener "A" =	"BABRACADABRA"
2)Borrar "B" =	"BARACADABRA"
2)Mantener "R" =	"BARACADABRA"
3)Borrar "A" =	"BARCADABRA"
3)Mantener "C" =	"BARCADABRA"
3)Mantener "A" =	"BARCADABRA"
4)Cambiar "D" por "Z" =	"BARCAZABRA"
4)Mantener "A" =	"BARCAZABRA"
5)Cambiar "B" por "S" =	"BARCAZASRA"
6)Borrar "R" =	"BARCAZASA"
7)Borrar "A" =	"BARCAZAS"

TRABAJO 3

Diseñar e implementar otro algoritmo (alternativo a este de Programación Dinámica) para resolver este problema de la distancia mínima entre dos cadenas, medir después los tiempos para los mismos tamaños del problema que en Programación Dinámica y para concluir hacer un estudio comparativo de los tiempos de ambos algoritmos.

Las clases que programe las incluirá dentro del paquete alg<dnipropio>.p5. Si utiliza Eclipse llamar al proyecto prac05_ProgramacionDinamica<UOpropio>

Se entregará, por separado, un documento con las actividades pedidas, y los ficheros fuente de las clases que haya tenido que programar junto con una explicación coherente de los resultados.
Esto se realizará a través de la tarea que se habilitará en el campus virtual. El plazo límite es un día antes de la próxima sesión de prácticas.