Página



Subsecuencia en común más larga

Planteamiento del problema 1.

La Programación Dinámica se aplica en cuatro fases:

- 1. Identificar la naturaleza n-etápica del problema
- 2. Verificación del Principio de Optimalidad de Bellman
- 3. Planteamiento de una recurrencia
- 4. Cálculo de una solución.

Naturaleza n-etápica del problema

Sea la subsecuencia más larga $x_1, x_2, ..., x_n$; esta subsecuencia es resultado de una serie de sucesiones ya que tenemos que decidir los valores de x_i , con $1 \le i \le n$. Así primero tomaríamos una decisión sobre x_1 , luego sobre x_2 , y así sucesivamente.

Por lo que podemos ver que estamos ante un problema de decisión n-etápico.

Principio de Optimalidad de Bellman

Si x_i es un elemento intermedio de la subsecuencia más larga, entonces la subsecuencia $x_1, x_2, ..., x_i$ es una solución optimal, y también lo es la subsecuencia $x_i, x_{i+1}, ..., x_n$.

Planteamiento de una recurrencia

[ANADIR TEXTO]

Cálculo de una solución

[ANADIR TEXTO]

Práctica

Página

Adrián Carmona Lupiáñez
3 Ignacio Sánchez Herrera
Jacobo Casado de Gracia

2/6

Jesús José M^a Maldonado Arroyo Juan Miguel Hernández Gómez



2. Pseudocódigo

Sean S1 y S2 las secuencias de las cuales queremos hallar la subsecuencia común más larga, el algoritmo se describiría por el siguiente pseudocódigo.

```
INICIO DEL ALGORITMO
       cadena reconstruccion(matriz M, cadena a, cadena b,
2
          \hookrightarrowentero i, entero j)
       INICIO DE LA FUNCION
3
           Si i o j son 0:
4
                Devolver {}
5
           Si no, si a[i-1] es igual a b[j-1]:
6
                Devolver reconstruccion(M, a, b, i-1, j-1)
           Si no, si M[i-1][j] es mayor que M[i][j-1]:
8
                Devolver reconstruccion(M, a, b, i-1, j)
9
10
                Devolver reconstruccion(M, a, b, i, j-1)
11
       FIN DE LA FUNCION
12
13
       TAM1 = | S1 |
14
       TAM2 = | S2 |
15
       Crear una matriz M con TAM1 filas y TAM2 columnas
16
       Rellenar la primera fila y columna de M con ceros
17
       Repetir desde i=1 hasta TAM1:
           Repetir desde j=1 hasta TAM2:
19
                Si S1[i-1] es igual a S2[j-1]:
20
                    M[i][j]=M[i-1][j-1]+1
21
                Si no:
                    M[i][j]=max(M[i-1][j],M[i][j-1])
23
       Devolver reconstruccion (M, S1, S2, TAM1, TAM2)
24
  FIN DEL ALGORITMO
25
```

NOTA: El algoritmo muestra al principio una función reconstrucción que es recursiva y a la cual se la llama en primer lugar en la línea 24 del código mostrado.

Práctica

Adrián Carmona Lupiáñez
3 Ignacio Sánchez Herrera

Jacobo Casado de Gracia

Página 3/6

Jesús José M^a Maldonado Arroyo Juan Miguel Hernández Gómez



3.	Eficiencia y ecuación recursiva	

Práctica

Página

ica 3

Adrián Carmona Lupiáñez Ignacio Sánchez Herrera Jacobo Casado de Gracia Jesús José M^a Maldonado Arroyo

Juan Miguel Hernández Gómez



4. Código

Aquí se muestra el código utilizado escrito en lenguaje C++.

4/6

Hemos indicado las 2 funciones utilizadas. La primera a la que se llama desde el main es la función subsecuencia. Y esta a su vez hace uso de la función reconstrucción.

```
//Funcion de reconstruccion recursiva
   string reconstruccion(vector <vector <int>> matriz, string
2
      \hookrightarrow a, string b, int i, int j){
       if (i==0 || j==0)
3
            return "";
4
       else if (a[i-1] == b[j-1])
5
            return reconstruccion (matriz, a, b, i-1, j-1) + a
6
               \hookrightarrow [i-1];
       else if (matriz[i-1][j]>matriz[i][j-1])
7
            return reconstruccion (matriz, a, b, i-1, j);
8
       else
            return reconstruccion (matriz, a, b, i, j-1);
10
   }
11
12
13
   //Funcion para hallar la subsecuencia mas corta con
14
      →programacion dinamica
   string subsecuencia(string a, string b){
15
       int a_tam= a.size();
16
       int b_tam= b.size();
17
18
       //Creacion de la matriz con los valores y la matriz
19
          ⇔con las direcciones
       vector <vector <int>> matriz(a_tam+1, vector <int> (
20
          \hookrightarrowb_tam+1, 0));
21
       for (int i=1; i<a_tam+1; i++){</pre>
22
            for (int j=1; j < b_tam + 1; j + +) {</pre>
23
                if (a[i-1]==b[j-1])
24
                     matriz[i][j] = matriz[i-1][j-1]+1;
25
                else{
26
                     if (matriz[i-1][j]>matriz[i][j-1])
27
                          matriz[i][j] = matriz[i-1][j];
28
                     else
29
                          matriz[i][j] = matriz[i][j-1];
30
                }
31
            }
32
       }
33
34
       //Return
35
       return reconstruccion(matriz, a, b, a_tam, b_tam);
36
37
```

Práctica

Página

3

Adrián Carmona Lupiáñez Ignacio Sánchez Herrera Jacobo Casado de Gracia Jesús José M^a Maldonado Arroyo

Juan Miguel Hernández Gómez



Escenarios de ejecución **5**.

5/6

Utilizando las secuencias jacobocasadodegracia y jesusjosemariamaldonadoarroyo, la subsecuencia común más larga es josadoaa.

Aquí podemos ver la matriz de números que se construye.

```
j e s u s j o s e m a r i a m a l d o n a d o a r r o y o
1 1 1 1 1 2 2 2 2 3 3 3 3 3 3 3 3 3 4 4 4 5 5 5 5 5 5
1 1 2 2 2 2 3 3 3 3 4 4 4 4 4 4 4 5 6 6 6 6 6 6 6 6 6 6 6
1 1 2 2 2 2 3 3 3 3 4 4 4 4 4 4 4 5 6 6 6 7 7 7 7 7
1 2 2 2 2 2 3 3 4 4 4 4 4 4 4 4 5 6 6 6 7 7 7 7 7
 2 2 2 2 2 3 3 4 4 4 4 4 4 4 4 4 5 6 6 6 7 7 7
1 2 2 2 2 2 3 3 4 4 4 5 5 5 5 5 5 5 6 6 6 7 7 7 8 8 8
1 2 2 2 2 2 3 3 4 4 5 5 5 6 6 6 6 6 6 6 7 7 7 8 8 8 8 8 8
1 2 2 2 2 2 3 3 4 4 5 5 5 6 6 6 6 6 6 6 7 7 7 8 8 8 8 8 8
1 2 2 2 2 2 3 3 4 4 5 5 6 6 6 6 6 6 6 6 7 7 7 8 8 8 8 8 8
1 2 2 2 2 2 3 3 4 4 5 5 6 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8
```

Y aquí podemos ver la matriz de direcciones que se construye.

i o gugio gomariamaldonado arrovo

		J	е	S	u	S	J	0	S	е	m	a	r	1	a	m	a	Τ	α	0	n	a	α	0	a	r	r	0	У	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i		\											_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
															١.		١					١.			١					
a	O	-	_	_	_	_	-	-	-	-	_	\	_	_	\	-	\	-	-	-	_	/	_	_	\	-	_	-	-	-
С	0	İ	-	-	-	-	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0	0	1	-	-	-	-	-	\	-	-	-	-	-	-	-	-	-	-	-	\	-	-	-	\	-	-	-	\	-	\
b	0	1	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	ĺ	-	-	-	-	-	-	-	-	-	_
0	0	1	-	-	-	-	-	\	-	-	-	-	-	-	-	-	-	-	-	\	-	-	-	\	-	-	-	\	-	\
С	0	1	-	-	-	-	-	Ì	-	-	-	-	-	-	-	-	-	-	-	Ì	-	-	-	Ì	-	-	-	_	-	_
																												-		
																												-		
																												-		
d	0	1	_	1	_	-	-	-	1	-	_	Ì	-	-	_	-	_	-	\	_	-	-	\	-	_	-	-	-	-	_
																												\		
d	0	1	-	1	-	-	-	Ì	-	-	-	1	-	-	-	-	-	-	\	Ť	-	-	\	-	-	-	-	-	-	_
е	0	1	\	-	-	-	_	1	_	\	-	-	-	-	-	-	-	-	Ť	1	-	-	Ì	-	-	-	-	-	-	_
g	0	1	Ì	_	_	-	-	1	-	Ì	_	_	-	-	-	-	-	-	1	1	-	-	1	_	-	-	-	-	-	_
																												-		
a	0	1	1	-	-	-	-	1	-	1	-	\	-	-	\	-	\	-	-	-	-	\	-	-	\	_	-	-	-	_
																												-		
																												-		
																												-		
												\			\		\					\			\					

Práctica

3

Adrián Carmona Lupiáñez Ignacio Sánchez Herrera Jacobo Casado de Gracia Jesús José M^a Maldonado Arroyo

Juan Miguel Hernández Gómez



Página 6/6

Vamos a observar otro ejemplo utilizando las secuencias de caracteres ignaciosanchezherrera y juanmiquelhernandezqomez, cuya subsecuencia común más larga es ignaneze.

```
j u a n m i g u e l h e r n a n d e z g o m e z
 0 0 1 1 1 1 2 2 2 2 2 2 2 3 4 4 4 4 4 4 4 4 4 4
   0 0 1 1 1 1 2 2 2 2 2 2 2 3 4 4 4 4 4 4 4 4 4 4
   0 0 1 1 1 2 2 2 2 2 2 2 2 3 4 4 4 4 4 4 4 4 4 4
   0 0 1 1 1 2 2 2 2 2 2 2 2 3 4 4 4 4 4 4 5 5 5 5
   0 0 1 1 1 2 2 2 2 2 2 2 2 3 4 4 4 4 4 4 5 5 5 5
   0 0 1 1 1 2 2 2 2 2 2 2 2 3 4 4 4 4 4 5 5 5 5
   0 0 1 2 2 2 2 2 2 2 2 2 2 3 4 5 5 5 5 5 5 5 5 5
c 0 0 0 1 2 2 2 2 2 2 2 2 2 3 4 5 5 5 5 5 5 5 5 5
h 0 0 0 1 2 2 2 2 2 2 2 3 3 3 3 4 5 5 5 5 5 5 5 5 5
 0 0 0 1 2 2 2 2 2 3 3 3 4 4 4 4 5 5 6 6 6 6 6 6 6
   0 0 1 2 2 2 2 2 3 3 3 4 4 4 4 5 5 6 7 7 7 7 7 7
h 0 0 0 1 2 2 2 2 2 3 3 4 4 4 4 4 5 5 6 7 7 7 7 7 7
   0 0 1 2 2 2 2 2 3 3 4 5 5 5 5 5 5 6 7 7 7 7 8 8
   0 0 1 2 2 2 2 2 3 3 4 5 6 6 6 6 6 6 7 7
   0 0 1 2 2 2 2 2 3 3 4 5 6 6 6 6 6 6 7 7 7
e 0 0 0 1 2 2 2 2 2 3 3 4 5 6 6 6 6 6 7 7 7 7 7
r 0 0 0 1 2 2 2 2 2 3 3 4 5 6 6 6 6 6 7 7 7 7 7 8 8
a 0 0 0 1 2 2 2 2 2 3 3 4 5 6 6 7 7 7 7 7 7 7 7 8 8
```

Y aquí podemos ver la matriz de direcciones que se construye.

		j	u	a	n	m	i	g	u	е	1	h	е	r	n	a	n	d	е	z	g	0	m	е	z
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	0	-	-	-	-	-	\	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
g	0	-	-	-	_	-	Ì	\	-	-	-	-	-	-	-	-	-	-	-	-	\	_	-	-	-
n	0	_	_	-	\	-	-	Ť	-	-	_	-	-	-	\	-	\	-	-	-	_	_	-	-	-
						-																		-	-
				٠,		_										١,					_			_	_
i	0	_	_	İ	_	-	\	_	_	_	_	_	_	_	İ	İ	_	_	_	_	_	_	_	_	_
						-									İ						-	\	_	_	_
s	0	_	_	i	_	_	i	_	_	_	_	_	_	_	i						_	٠,	_	_	_
a	0	_	_	\	_	_	Τ	_	_	_	_	_	_	_	i	į	_	_	_	_	_	i	_	_	_
n	0	_	_	ì	\	-	_	_	_	_	_	_	_	_	į	ì	\	_	_	_	_	_	_	_	_
C	0	_	_	i	ì	_	_	_	_	_	_	_	_	_	ì	i	ì	_	_	_	_	_	_	_	_
						-															_	_	_	_	_
	0																				_	_	_	\	_
_	0									,			٠,						•					_	
_	0					_																		_	_
	0																							\	_
r		_		i	i	-	_	_	_	\ 	_	i	\ 	\	_	_	_	_	_	¦	_	_	_	\ 	_
r		_				-	_	_	_					/	_	_	_				_	_	_	÷	
_	0																							\	
_	-																					_	_	\	_
						-								٠,								_	_	1	_
a	U	-	-	/	1	-	_	_	_	1	_	١	١	1	-	/	_	_	_	_	_	_	_	1	-