

Práctica 4: Programación Dinámica

Manuel Vicente Bolaños Quesada
Pablo Gálvez Ortigosa
Carlos García Jiménez

1/6/2022

- ① Introducción
- ② Programación Dinámica: Algoritmo
- ③ Programación Dinámica: Dos ejemplos
- ④ Conclusión

- 1 Introducción
- 2 Programación Dinámica: Algoritmo
- 3 Programación Dinámica: Dos ejemplos
- 4 Conclusión

Introducción

- Problema a resolver mediante **Programación Dinámica**

Encontrar la mayor subsecuencia común a dos secuencias dadas (LCS)

- 1 Introducción
- 2 Programación Dinámica: Algoritmo
- 3 Programación Dinámica: Dos ejemplos
- 4 Conclusión

Propiedades Programación Dinámica

El problema verifica las propiedades para ser resuelto por PD:

- **Naturaleza n -etápica**
- **Verifica el Principio de Optimalidad de Bellman**
- **Plantea una recurrencia**
- **Calcula la solución** al terminar todas las etapas

Algoritmo

- Primeras fila y columna se rellenan con ceros.
- Cada celda está compuesta de una flecha o puntero y un número
- Si las letras de la fila y la columna coinciden
 - Se escoge el número anterior de la diagonal izquierda + 1
 - Se escribe una flecha apuntando a la diagonal izquierda
- Si las letras no coinciden
 - Se escoge el máximo elemento entre el de la izquierda y el de arriba
 - Se escribe una flecha apuntando al elemento mayor
 - Si coinciden, da igual adonde apunte la flecha
- Al terminar, nos situamos en la celda inferior derecha y se siguen las flechas para construir la subsecuencia (se añaden las que tienen flechas diagonales)

- 1 Introducción
- 2 Programación Dinámica: Algoritmo
- 3 Programación Dinámica: Dos ejemplos**
- 4 Conclusión

Problema

- Comparar dos cadenas de caracteres
- Hallar el % de similitud entre las dos cadenas

Para ello, hallaremos la **subsecuencia común más larga**.

Primera pareja

- 1 “**ab**bcdef**abcd**xzyccd”
- 2 “**ab**bcdeaf**bcd**zxyccd”

- Subsecuencia más larga: “abbcdeabcdzyccd”
- Porcentaje de similitud:

$$\frac{15}{17} \cdot 100\% = 88.2352941\%$$

Dicha secuencia más larga no tiene por qué ser única:
“abbcdefbcdzyccd” también válida

	a	b	b	c	d	e	a	f	b	c	d	z	x	y	c	c	d
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	↖ 1	← 1	← 1	← 1	← 1	↖ 1	← 1	← 1	← 1	← 1	← 1	← 1	← 1	← 1	← 1	← 1
b	0	↑ 1	↖ 2	↖ 2	← 2	← 2	← 2	← 2	↖ 2	← 2	← 2	← 2	← 2	← 2	← 2	← 2	← 2
b	0	↑ 1	↖ 2	↖ 3	← 3	← 3	← 3	← 3	↖ 3	← 3	← 3	← 3	← 3	← 3	← 3	← 3	← 3
c	0	↑ 1	↑ 2	↑ 3	↖ 4	← 4	← 4	← 4	← 4	↖ 4	← 4	← 4	← 4	← 4	↖ 4	↖ 4	← 4
d	0	↑ 1	↑ 2	↑ 3	↑ 4	↖ 5	← 5	← 5	← 5	↖ 5	← 5	← 5	← 5	← 5	← 5	← 5	↖ 5
e	0	↑ 1	↑ 2	↑ 3	↑ 4	↑ 5	↖ 6	← 6	← 6	← 6	← 6	← 6	← 6	← 6	← 6	← 6	← 6
f	0	↑ 1	↑ 2	↑ 3	↑ 4	↑ 5	↑ 6	← 6	↖ 7	← 7	← 7	← 7	← 7	← 7	← 7	← 7	← 7
a	0	↖ 1	↑ 2	↑ 3	↑ 4	↑ 5	↑ 6	↖ 7	← 7	← 7	← 7	← 7	← 7	← 7	← 7	← 7	← 7
b	0	↑ 1	↖ 2	↖ 3	↑ 4	↑ 5	↑ 6	↑ 7	← 7	↖ 8	← 8	← 8	← 8	← 8	← 8	← 8	← 8
c	0	↑ 1	↑ 2	↑ 3	↖ 4	↑ 5	↑ 6	↑ 7	← 7	↑ 8	↖ 9	← 9	← 9	← 9	↖ 9	↖ 9	← 9
d	0	↑ 1	↑ 2	↑ 3	↑ 4	↖ 5	↑ 6	↑ 7	← 7	↑ 8	↑ 9	↖ 10	← 10	← 10	← 10	← 10	↖ 10
x	0	↑ 1	↑ 2	↑ 3	↑ 4	↑ 5	↑ 6	↑ 7	← 7	↑ 8	↑ 9	↑ 10	← 10	↖ 11	← 11	← 11	← 11
z	0	↑ 1	↑ 2	↑ 3	↑ 4	↑ 5	↑ 6	↑ 7	← 7	↑ 8	↑ 9	↑ 10	↖ 11	← 11	← 11	← 11	← 11
y	0	↑ 1	↑ 2	↑ 3	↑ 4	↑ 5	↑ 6	↑ 7	← 7	↑ 8	↑ 9	↑ 10	↑ 11	← 11	↖ 12	← 12	← 12
c	0	↑ 1	↑ 2	↑ 3	↖ 4	↑ 5	↑ 6	↑ 7	← 7	↑ 8	↖ 9	↑ 10	↑ 11	← 11	↑ 12	↖ 13	↖ 13
c	0	↑ 1	↑ 2	↑ 3	↖ 4	↑ 5	↑ 6	↑ 7	← 7	↑ 8	↖ 9	↑ 10	↑ 11	← 11	↑ 12	↖ 13	↖ 14
d	0	↑ 1	↑ 2	↑ 3	↑ 4	↖ 5	↑ 6	↑ 7	← 7	↑ 8	↑ 9	↖ 10	↑ 11	← 11	↑ 12	↑ 13	↑ 14

Segunda pareja

- ① “010111000100010101010010001001001001”
- ② “110000100100101010001010010011010100”

- Subsecuencia más larga: “110000001010101001000100100100”
- Porcentaje de similitud:

$$\frac{30}{36} \cdot 100\% = 83.3333333\%$$

- 1 Introducción
- 2 Programación Dinámica: Algoritmo
- 3 Programación Dinámica: Dos ejemplos
- 4 Conclusión**

Conclusión

- Programación Dinámica muy útil si se puede construir la solución a partir de la solución de anteriores etapas.
- En el problema estudiado $\rightarrow O(n \cdot 2^n)$ a $O(n^2)$!!
- Desventaja: Grandes requerimientos de memoria
- Auge genética (Genoma Humano) y más memoria en dispositivos \rightarrow Cada vez más interés de Programación Dinámica