Apuntes-Tema-4.pdf

Juandf03

Análisis y diseño de algoritmos

2º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Tema 4 "Programación Dinámica"

1. Introducción

- **Conceptos generales**

Tabla ⟹ soluciones de los subproblemas

- **Números combinatorios**

$$\begin{cases} C(n,k) = C(n-1,k-1) + C(n-1,k) & n>k>0 \\ C(n,0) = C(n,n) = 1 \end{cases}$$

Divide y Vencerás tenemos 2^n llamadas

Con una tabla (Triángulo de Pascal) ⟶ $\Theta(nk)$

2. Programación Dinámica

Método para reducir el tiempo de ejecución de un algoritmo mediante la utilización de subproblemas superpuestos y subestructuras óptimas.

- **Principio de Optimalidad** ❗

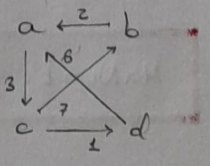
Un problema exhibe la propiedad de subestructura óptima, si una solución óptima al mismo contiene la solución óptima a subproblemas del mismo.

- **Algoritmo de Floyd**

¡ camino mínimo, grafos ponderados !
entre z nodos

¡Ej!

	a	b	c	d
a	0	∞	3	∞
b	2	0	∞	∞
c	∞	7	0	1
d	6	∞	∞	0

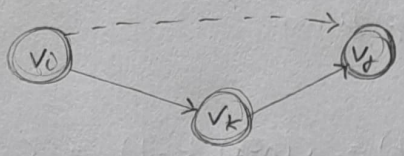


	a	b	c	d
a	0	10	3	4
b	2	0	5	6
c	7/9	7	0	1
d	6	16	9	0

• Para ir de c → a : c → b → a [T=9]

c → d → a [T=7] Camino más corto

¡ Sucesión de matrices ! $D^0 \longrightarrow \dots \longrightarrow D^n$



$d_{ij}^{k} = d_{ik}^{k-1} + d_{kj}^{k-1}$

Principio de optimalidad

$$d_{ij}^{k} = \min\left(d_{ij}^{k-1},\ d_{ik}^{k-1} + d_{kj}^{k-1}\right)$$

⟹ $T(n) \in \Theta(n^3)$, n nº de vértices del grafo

• Problema de las monedas

$C_{ij} \Longrightarrow \begin{cases} i \text{ (valor de la moneda)} \\ j \text{ (cantidad a pagar)} \end{cases}$

$C[i,j] = \min(C[i-1,j],\ 1 + C[i-1,\ j-d_i])$

¡Ej! M= 8 céntimos

	0	1	2	3	4	5	6	7	8
$d_1 = 1$	0	1	2	3	4	5	6	7	8
$d_2 = 4$	0	1	2	3	1	2	3	4	2
$d_3 = 6$	0	1	2	3	1	2	1	2	2

MinNum [i, j] | MinNum [i, j − di] + 1 | MinNum [i − 1, j]

¡ Fórmulas Ai,j !

WUOLAH

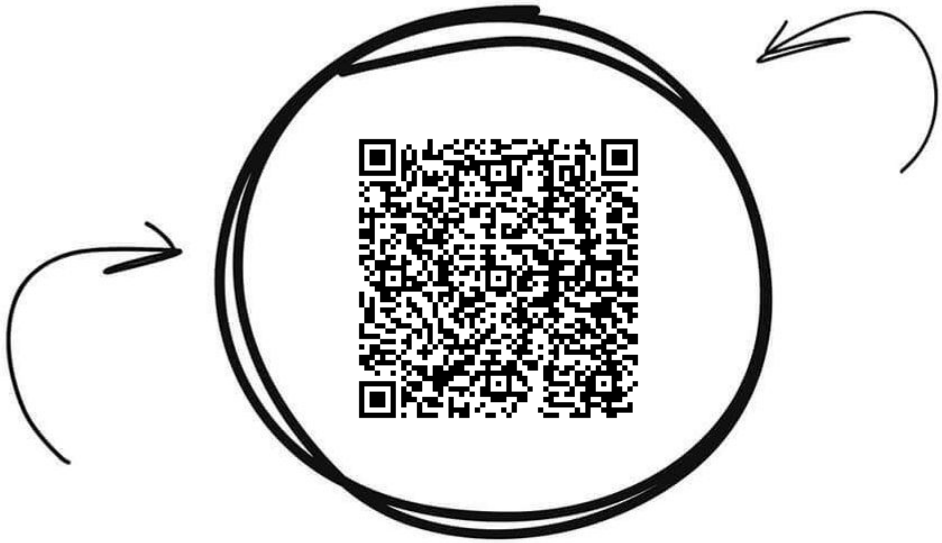
+

InfoJobs

Te ayudan a entrar en el mundo laboral

- Trabajos temporales
- Prácticas de empresa
- Tus primeros puestos de trabajo en aquello que has estudiado.

Análisis y diseño de algoritmos



Banco de apuntes de la

WUOLAH

Comparte estos flyers en tu clase y consigue más dinero y recompensas

1. Imprime esta hoja
2. Recorta por la mitad
3. Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes
4. Llévate dinero por cada descarga de los documentos descargados a través de tu QR

• Funciones con memoria

Técnica que combina recursión más una tabla de almacenamiento temporal.

NO HACE FALTA RELLENAR TODA LA TABLA ⟹ calcular los valores necesarios

• Problema de la mochila

n objetos

pesos ⟶ $w_1, w_2, \ldots, w_n$

valor ⟶ $v_1, v_2, \ldots, v_n$

W capacidad mochila

$V[i,j] \Longrightarrow \begin{cases} i \Rightarrow \text{nº de objetos} \\ j \Rightarrow \text{peso mochila} \end{cases}$

$V[i,j] = \max \{ V[i-1,j] \;,\; v_i + V[i-1, j-w_i] \}$

¡ Fórmula $V[i,j]$!

¡Ej!

item	1	2	3	4
peso	2	1	3	2
valor	12	10	20	15

W = 5

i \ j (capacidad)	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0	10	15	25	30	37

- **Subsecuencia común más larga**

$X = \langle x_1, x_2, \dots x_n \rangle$

$Y = \langle y_1, y_2, \dots, y_n \rangle$

encontrar la subsecuencia común más larga

$$C[i,j] = \begin{cases} 0 & \text{si } i=0 \ \vee \ j=0 \\ C[i-1, j-1] + 1 & \text{si } i,j>0 \ \wedge \ x_i = y_i \\ \max(C[i-1,j], C[i,j-1]) & \text{si } i,j>0 \ \wedge \ x_i \neq y_i \end{cases}$$

¡Ej!

$X = ABCBDAB$

$Y = BDCABA$

$\Longrightarrow$ $Z = BCBA$

Tema 5 "Algoritmos Voraces"

1. Introducción

Las soluciones óptimas de los subproblemas llevan a la solución óptima del problema general.

→ **Enfoque voraz**

No exploramos todas las alternativas ⇒ construir vector solución

En cada paso se decide cual de entre todas las soluciones posibles es la mejor. Condiciones:

⊕ Factible

⊕ localmente óptima

⊕ Irrevocable