

ANÁLISIS Y DISEÑO DE ALGORITMOS

Práctica 6 de laboratorio

Realización y entrega (esta práctica ocupará dos sesiones):

Sesión 1: Versiones recursivas y tabla de memoización, del 12 al 17 de marzo de 2015
Sesión 2: Práctica completa, del 23 al 26 de marzo de 2015

(en la sesión de laboratorio que corresponde a cada alumno)

El problema del laberinto

Se dispone de una cuadrícula $n \times m$ de valores $\{0, 1\}$ que representa un laberinto. Un valor 0 en una casilla cualquiera de la cuadrícula indica una posición inaccesible; por el contrario, con el valor 1 se simbolizan las casillas accesibles. Por ejemplo:

$\xrightarrow{(1,1)}$	1	1	0	0	1
	1	1	1	1	1
	0	1	1	0	0
	1	1	0	1	1
	1	1	1	0	0
	0	0	0	1	1 $\xrightarrow{(6,5)}$

Se pide, aplicar el método *programación dinámica* para calcular el número de caminos distintos¹ que hay desde la casilla origen $(1, 1)$ hasta la casilla destino (n, m) asumiendo que sólo son validos tres tipos de movimientos desde una casilla cualquiera (i, j) :

1. derecha: $(i, j + 1)$,
2. abajo: $(i + 1, j)$,
3. abajo y derecha (diagonal): $(i + 1, j + 1)$.

Como es evidente, tampoco son válidos los movimientos que llevan al exterior del laberinto ni los que conducen a casillas inaccesibles.

Para resolver este ejercicio se debe implementar cuatro algoritmos:

1. Recursivo sin almacén (ineficiente)
2. Recursivo con almacén (memoización)
3. Iterativo con almacén (tabla)
4. Iterativo con almacén (vector -versión con complejidad espacial mejorada-)

Salida del programa: Deberá mostrarse por pantalla y en este orden:

1. La solución obtenida mediante dichos algoritmos (como es lógico, las cuatro soluciones coincidirán).
2. La matriz de cálculos intermedios obtenida en la versión recursiva con almacén (memoización). Las casillas que no han sido utilizadas deberán contener el valor -1.
3. La matriz de cálculos intermedios obtenida en la versión iterativa con almacén (tabla).
4. La secuencia de casillas (i, j) que componen un camino cualquiera (caso de existir) comenzando en $(1, 1)$ y terminando en (n, m)

¹Dos caminos, con el mismo origen y destino, son distintos si varían en al menos una de las casillas intermedias que lo componen

Entrada al programa:

El laberinto se suministrará codificado en un fichero de texto cuyo nombre se recogerá a través de la línea de comando, por ejemplo: **laberinto entrada.txt** donde entrada.txt es el nombre del fichero que contiene el laberinto a resolver. Su formato y contenido será:

- Línea 1 del fichero: valores n y m separados mediante un único espacio en blanco.
- Línea 2 (y siguientes): m valores $\{0,1\}$ que componen la primera fila (y siguientes) del laberinto, separados mediante un único espacio en blanco

por tanto, el fichero contendrá $n + 1$ líneas, todas ellas (incluso la última) finalizarán con un salto de línea.

Ejemplo:

Sea el siguiente contenido de un fichero de texto. Se trata de un laberinto (6×5):

6	5			
1	1	0	0	1
1	1	1	1	1
0	1	1	0	0
1	1	0	1	1
1	1	1	0	0
0	0	0	1	1

La salida será:

Recursivo sin almacén (ineficiente): 8
 Recursivo con almacén (memoización): 8
 Iterativo con almacén (tabla): 8
 Iterativo con almacén (vector): 8
 Tabla resultados parciales (recursivo):

1	1	-1	-1	-1
1	3	-1	-1	-1
-1	4	-1	-1	-1
0	4	-1	-1	-1
0	4	8	-1	-1
-1	-1	-1	8	8

Tabla de resultados parciales (iterativo):

0	0	0	0	0	0
0	1	1	0	0	0
0	1	3	4	4	4
0	0	4	11	0	0
0	0	4	0	11	11
0	0	4	8	0	0
0	0	0	0	8	8

Un posible camino:
 (1,1) (2,1) (3,2) (4,2) (5,3) 6,4) (6,5)

Otros ejemplos:

[illegible]