

ANÁLISIS Y DISEÑO DE ALGORITMOS

VUELTA ATRÁS

Práctica 8 de laboratorio

Entrega: hasta el 19 de mayo, 23:55h.

A través de Moodle

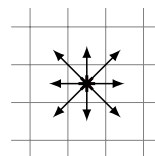
El problema del laberinto con 8 movimientos

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:

$\xrightarrow{(0,0)}$	1	1	0	1	1	0	1
	1	0	1	1	0	0	1
	1	0	1	0	0	1	0
	0	0	1	0	1	0	1
	1	1	0	0	1	0	1
	1	0	1	1	0	1	0
	1	0	0	0	0	1	0
	0	1	0	1	1	0	1
	0	1	0	1	1	1	$\xrightarrow{(8,6)}$

Un laberinto 9×7



El laberinto con ocho movimientos: Se permite visitar cualquier casilla colindante siempre que sea accesible.

Se pide aplicar el **método vuelta atrás** para encontrar el camino de longitud mínima¹ que conduzca a la salida del laberinto suponiendo, a diferencia de las prácticas anteriores, que desde una posición cualquiera (i, j) puede visitarse cualquiera de sus ocho casillas adyacentes siempre que sea accesible. El punto de partida es la casilla $(0, 0)$ y el de llegada la casilla $(n - 1, m - 1)$.

1. Nombre del programa, opciones y sintaxis de la orden.

El programa a realizar se debe llamar **maze-bt**. La orden tendrá la siguiente sintaxis:

maze-bt [-p] -f fichero_entrada

El fichero con el laberinto a resolver se suministra a través de la opción -f. Si se hace uso del parámetro opcional -p, el programa mostrará el camino de longitud mínima encontrado (no tiene por qué ser único), en dos dimensiones, superpuesto a la matriz que codifica el laberinto, escribiendo el carácter '*' en cada una de las casillas que lo componen. (véanse los ejemplos de salida).

En el caso de que se haga uso de la orden con una sintaxis distinta a la descrita se emitirá un mensaje de error advirtiéndolo y a continuación se mostrará la sintaxis correcta. No es necesario controlar

¹Se entiende por longitud del camino el número de casillas que lo componen. Puede que exista más de un camino de longitud mínima, en tal caso sólo debe mostrarse uno de ellos.

posibles errores en el contenido del fichero de entrada ya que siempre se ajustará fielmente al formato establecido (véase siguiente apartado).

2. Entrada de datos al programa. Formato del fichero de entrada.

De manera idéntica a las prácticas anteriores, el laberinto $n \times m$ se suministrará codificado en un fichero de texto cuyo nombre se recogerá a través de la opción -f. 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 lo tanto, el fichero contendrá $n + 1$ líneas que finalizarán con un salto de línea salvo, en todo caso, la última línea.

3. Salida del programa. Ejemplos de ejecución.

El programa mostrará los siguientes resultados en el caso de que exista camino de salida:

- a) La longitud del camino más corto que conduce a la salida.
- b) El tiempo de CPU empleado para resolver el problema, en milisegundos.

En el caso de que se utilice el parámetro opcional -p, se mostrará el camino obtenido tal y como se ha mencionado anteriormente.

Por otra parte, si no existe camino de salida se mostrará únicamente el literal **NO EXIT** y nada más, aunque se haya utilizado el parámetro -p.

Es imprescindible ceñirse al formato y texto de salida mostrado, incluso en lo que se refiere a los saltos de línea o carácter separador, que en todos los casos es el espacio en blanco.

A través de *Moodle* se puede descargar un archivo comprimido con varios ejemplos (0?.8maze), junto con una posible solución. Entre ellos están los ficheros utilizados a continuación para describir el formato de la salida.

<pre>\$maze-bt -f 02.8maze Shortest path length= 13 CPU elapsed time= 0 ms. \$maze-bt -f 02.8maze -p Shortest path length= 13 CPU elapsed time= 0 ms. Path 2D: **01101 10*1001 10*0010 00*0101 1*00*01 10**0*0 10000*0 010110* 010111*</pre>	<pre>\$maze-bt -p -f 07.8maze Shortest path length= 49 CPU elapsed time= 0 ms. Path 2D: *****1 00000000* 0*111010* *0*0**10* *01*10*0* *01010*0* *01010*0* *011101*1 *00000000 1*****101 000000*01 1*****101 *00000001 1*****</pre>
<pre>\$maze-bt -p -f 06.8maze NO EXIT</pre>	

Normas para la entrega.

ATENCIÓN: Estas normas son de obligado cumplimiento para que esta práctica sea evaluada.

- a)* Se debe entregar el código fuente y un *makefile* para obtener el ejecutable. No hay que entregar nada más, en ningún caso se entregarán ficheros de test.
- b)* Es imprescindible que no presente errores ni de compilación ni de interpretación (según corresponda), en los ordenadores del laboratorio asignado y en el sistema operativo *GNU/Linux*.² Se tratará de evitar también cualquier tipo de aviso (*warning*).
- c)* Todos los ficheros que se entregan deben contener el nombre del autor y su DNI (o NIE) en su primera línea (entre comentarios para que no afecte a la compilación).
- d)* Se comprimirán en un archivo **.tar.gz** cuyo nombre será el DNI del alumno, compuesto de 8 dígitos y una letra (o NIE, compuesto de una letra seguida de 7 dígitos y otra letra). Por ejemplo: **12345678A.tar.gz** o **X1234567A.tar.gz**. **Solo se admite este formato de compresión y solo es válida esta forma de nombrar el archivo.**
- e)* En el archivo comprimido **no deben existir subcarpetas**, es decir, al extraer sus archivos estos deben quedar guardados en la misma carpeta donde está el archivo que los contiene.
- f)* La práctica hay que subirla a *Moodle* respetando las fechas expuestas en el encabezado de este enunciado.

²Si trabajas con tu propio ordenador o con otro sistema operativo asegúrate de que este requisito se cumple.