

# ANÁLISIS Y DISEÑO DE ALGORITMOS

## RAMIFICACIÓN Y PODA

### Práctica final

Entrega: hasta el 9 de junio, 23:55h.

A través de Moodle

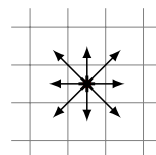
## El laberinto con 8 movimientos (II)

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

Un laberinto  $9 \times 7$



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

Se pide aplicar el esquema **ramificación y poda** para encontrar el camino de longitud mínima<sup>1</sup> que conduzca a la salida del laberinto suponiendo, al igual que en la práctica anterior, 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  $(n - 1, m - 1)$ . Además, se debe realizar un estudio del comportamiento del método según distintas estrategias de búsqueda en combinación con el uso (o no) de funciones de cota.

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

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

**maze-bb** [-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

<sup>1</sup>Se 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 (véase los ejemplos de ejecución):

- a) La longitud del camino más corto que conduce a la salida.
- b) Número de nodos que han sido explorados. Este valor a su vez se desglosará entre los siguientes:
  - 1) Número de nodos añadidos a la lista de nodos vivos.
  - 2) Número de nodos descartados por no ser prometedores.
  - 3) Número de nodos descartados por no ser factibles.
- c) Número de nodos expandidos.
- d) Número de nodos completados. De entre estos se cuantificarán además los que han actualizado la mejor solución hasta el momento.
- e) Número de nodos prometedores que finalmente fueron descartados (dejaron de ser prometedores).
- f) Número de veces que la mejor solución hasta el momento se ha actualizado a partir de una cota pesimista (este valor será 0 en el caso de que no se haga uso de cotas pesimistas en los nodos intermedios).
- g) Tiempo de CPU, expresado en milisegundos, que ha consumido el algoritmo de ramificación y poda para resolver el problema.

En el caso de que se utilice el parámetro opcional -p, se mostrará además el camino obtenido tal y como se ha mencionado anteriormente (sólo debe mostrarse el camino si se hace uso de esta opción).

Por otra parte, si no existe camino de salida se mostrará el literal **NO EXIT** en lugar de la longitud del camino más corto. Tampoco se mostrará el camino de salida aunque se haya utilizado el parámetro -p. La información relativa a los nodos generados, descartados, etc. sí debe mostrarse.

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 (25.8maze), junto con una posible solución. Entre ellos están los ficheros utilizados a continuación para describir el formato de la salida.

Ejemplos de ejecución:

```
$maze-bb -f 25.8maze
Shortest path length= 327
Explored nodes= 81665 (Added= 10408; nonpromising= 50765; nonfactible= 20492)
Expanded nodes= 10208
Completed nodes= 1 (Best solution updated= 1)
Promising but discarded nodes= 199
Best solution updated from a pessimistic bound= 0
CPU elapsed time= 30 ms.
```

```
$maze-bb -f 24.8maze
NO EXIT
Explored nodes= 1999929 (Added= 249991; nonpromising= 1743944; nonfactible= 5994)
Expanded nodes= 249991
Completed nodes= 0 (Best solution updated= 0)
Promising but discarded nodes= 0
Best solution updated from a pessimistic bound= 0
CPU elapsed time= 1830 ms.
```

```
$maze-bb -p -f 05.8maze
Shortest path length= 9
Explored nodes= 9 (Added= 1; nonpromising= 2; nonfactible= 6)
Expanded nodes= 1
Completed nodes= 0 (Best solution updated= 0)
Promising but discarded nodes= 0
Best solution updated from a pessimistic bound= 1
CPU elapsed time= 0 ms.
```

```
Path 2D:
*1111111
0*100111
1*011111
11*00111
000*1111
1111*111
11000*11
111100*0
1111100*
```

Todos los valores mostrados, salvo la longitud del camino más corto, dependen de cada implementación: estructuras de datos utilizadas, estrategias de búsqueda, cotas optimistas y pesimistas, etc. y por lo tanto, no tienen por qué coincidir con los obtenidos en cada caso particular.

Es importante hacer constar que uno de los criterios que se seguirá para corregir esta práctica es el del tiempo de respuesta. Por ello, aunque se hayan implementado distintas estrategias de búsqueda, funciones de cota, etc. el programa que se entrega debe funcionar con la configuración que se entienda más rápida. No obstante, en la memoria se deberá hacer mención a cualquier otro trabajo realizado que, por ese motivo, no se haya incluido en la versión final del programa.

#### 4. Documentación del trabajo realizado.

Se debe entregar una **memoria**, en formato PDF y con nombre **memoria.pdf**, que describa las estructuras de datos empleadas así como la/s estrategia/s de búsqueda, los mecanismos de poda y las cotas optimistas y pesimistas utilizados.

Incluirá también un análisis del comportamiento de distintas estrategias de búsqueda en conjunción con distintos mecanismos de poda. Para ello, se realizará un cuadro comparativo que contenga, para cada caso estudiado, el número de nodos que se añaden a la lista de nodos vivos y el tiempo de respuesta del algoritmo (según se ha explicado en clase de teoría).

La primera línea de la memoria contendrá el nombre del autor y su DNI. El resto se estructurará en seis apartados de la siguiente manera:

##### 1. Estructuras de datos

###### 1.1 Nodo

En este apartado se describirá brevemente el contenido del nodo, es decir, las variables que lo componen y su cometido.

###### 1.2 Lista de nodos vivos

Breve descripción de la estructura de datos utilizada y si procede, qué criterio se sigue para extraer el nodo más prometedor (si se han analizado varios criterios, se mencionará en este apartado el que se ha considerado mejor).

## 2. Mecanismos de poda

### 2.1. Poda de nodos no factibles

Se deberá indicar cómo se descartan los nodos que no son factibles; si no procede este tipo de poda debe indicarse el motivo.

### 2.2. Poda de nodos no prometedors

Se deberá indicar cómo se decide que un nodo no es prometedor; si no procede debe indicarse el motivo.

## 3. Cotas pesimistas y optimistas

Se describirá en qué consisten dichas cotas, distinguiéndose entre el nodo inicial y los demás nodos.

### 3.1 Cota pesimista y optimista del nodo inicial

### 3.2 Cota pesimista y optimista de los demás nodos

## 4. Otros medios empleados para acelerar la búsqueda

En este apartado se citará cualquier otro mecanismo utilizado con el objetivo de acelerar la búsqueda.

## 5. Estudio comparativos de distintas estrategias de búsqueda

En este apartado se incluirá el cuadro comparativo mencionado anteriormente.

## 6. Soluciones y tiempos de ejecución.

Se mostrará en este apartado el tiempo de proceso requerido para resolver cada uno de los ficheros de test publicados para esta práctica cuya numeración comienza en 2, es decir, los archivos con nombre 2?.8maze. Si alguno de los ficheros no ha podido ser resuelto por el programa realizado se escribirá '?'.<sup>2</sup>

- Fichero 20.8maze: 0 ms.
- ...
- Fichero 24.8maze: 1820 ms.
- Fichero 25.8maze: 30 ms.
- ...
- Fichero 29.8maze: ?

**IMPORTANTE:** Todos los apartados (salvo el último) se complementarán con el correspondiente fragmento de código que corrobore el trabajo descrito. Si algunos de los apartados no ha sido realizado se pondrá únicamente la frase “NO IMPLEMENTADO”.

De los distintos elementos que pueden contener un algoritmo de Ramificación y poda, sólo se evaluarán los que estén reflejados en esta memoria. Es decir, todo lo que no esté en dicho documento se entenderá que no ha sido realizado.

## 7. 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 y además el fichero `memoria.pdf`. 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*.<sup>2</sup> 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.

---

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