

TRABAJO PRÁCTICO

Cátedra Programación II

Noviembre 2019

1. Presentación del ejercicio

En este Trabajo Práctico el objetivo será implementar un laberinto y una resolución para el mismo. El problema consiste en:

Dada una configuración inicial, se debe construir un laberinto en el que se asegure la posibilidad, partiendo de una posición, de llegar a un objetivo; esto se realizará evitando obstáculos. La salida generada deberá ser el laberinto y, un posible camino.

El programa realizado en *C* generará una salida, en un archivo, con el laberinto; el programa en *Python* tomará esa entrada y deberá devolver el laberinto con la salida correspondiente, es decir, encontrar el camino; en caso de no ser posible, debe volver a ejecutar el programa de *C* hasta tener un laberinto que cumpla con la característica pedida.

2. Generación del Laberinto

El programa que se debe implementar en *C* debe tomar como entrada, en un archivo, que va a contar con el siguiente formato:

```
dimension
7
obstaculos fijos
(1,2)
(1,5)
(2,4)
obstaculos aleatorios
5
posicion inicial
(1,1)
objetivo
(5,4)
```

Se asume que el laberinto es cuadrado (cantidad de filas y de columnas iguales).

Para representar el laberinto en sí, utilizaremos una matriz de caracteres, en donde la posición $[i, j]$ puede tomar los siguientes valores:

- I si es una casilla de salida;

- **0** si es una casilla libre;
- **1** si es una casilla en donde hay una pared;
- **X** si es la casilla objetivo.

La salida generada, entonces, será, por ejemplo:

```
1100100
0101000
0010000
0000000
000X000
0001000
0100100
```

En este ejemplo, los obstáculos aleatorios que salieron se encuentran en las posiciones:

$(2, 2), (3, 3), (6, 4), (7, 2), (7, 5)$

3. Resolución del Laberinto

El programa que se debe implementar en *Python* toma la salida generada por el programa anterior, debe encontrar el camino hasta la salida dando los pasos que se deben seguir. Esto se indica dando el recorrido, casilla por casilla.

Consideramos que, dada una casilla, se puede desplazar a las contiguas en cualquier dirección posible (arriba, abajo, derecha, izquierda).

Con la entrada anterior, una salida posible es:

$(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 5), (3, 5), (4, 5)$

En caso de no poder encontrarse, se debe ejecutar, nuevamente, el programa de C, con el mismo archivo de entrada.

Como se indicó, este proceso se repetirá hasta que se pueda generar un laberinto con un camino.

Observación: el archivo de entrada dado, siempre permitirá la construcción de un laberinto que cuente con un camino posible. La ubicación, aleatoria de las paredes pueden impedirlo pero, eventualmente, se podrá obtener un laberinto que cumpla con dicha propiedad.

4. Ejecutando un programa desde Python

Para ejecutar el programa de *C* desde *Python* debemos hacer lo siguiente:

```
import subprocess
...
response = subprocess.run("./a.out EntradaLaberinto.txt")
```

5. Características del Código a Entregar

Se pide que escriba un programa que cumpla con los siguientes requisitos:

- los nombres de funciones y variables deber ser significativos (estar relacionados con su uso);
- no se pueden usar variables globales, definidas fuera de funciones;
- se debe usar *Pytest* para verificar los casos de uso de cada función que retorne valores;
- se debe comentar, adecuadamente, el código.