

Bitácora 14 del Taller de Herramientas Computacionales

Elías Jiménez Cruz, 409085596

25/01/2019

En la penúltima clase, se checaron dudas sobre la tarea del día anterior, que consistió en realizar los ejercicios de la sección de evaluación del libro Python fácil. Eran ocho ejercicios, de los cuales el que presentó más dudas fue el del laberinto. Consiste en lo siguiente: "Diseñe y programe un algoritmo recursivo que encuentre la salida de un laberinto, teniendo en cuenta que el laberinto se toma como entrada y que es una matriz de valores True, False, (x,y), (a,b), donde True indica un obstáculo; False, una celda por la que se puede caminar; (x,y), el punto donde comienza a buscarse la salida y (a,b), la salida del laberinto.". Entonces, se checaron varias posibles soluciones, yo en particular realicé el algoritmo de la siguiente manera:

```
{# -*- coding: utf-8 -*-
'''Diseñe y programe un algoritmo recursivo que encuentre la salida
de un laberinto, teniendo en cuenta que el laberinto se toma como entrada y
que es una matriz de valores True, False, (x,y), (a,b), donde True
indica un obstáculo; False, una celda por la que se puede caminar;
(x,y), el punto donde comienza a buscarse la salida y (a,b), la salida del
laberinto.'''

def laberinto(matriz, e):
    try:
        if matriz[e[0]-1][e[1]] == False:
            matriz[e[0]][e[1]] = True
```

```

return(laberinto(matriz, (e[0]-1,e[1])))
except:
pass
try:
if matriz[e[0]][e[1]+1] == False:
matriz[e[0]][e[1]] = True
return(laberinto(matriz, (e[0],e[1]+1)))
except:
pass
try:
if matriz[e[0]+1][e[1]] == False:
matriz[e[0]][e[1]] = True
return(laberinto(matriz, (e[0]+1,e[1])))
except:
pass
try:
if matriz[e[0]][e[1]-1] == False:
matriz[e[0]][e[1]] = True
return(laberinto(matriz, (e[0],e[1]-1)))
except:
pass
return("La salida es: "+str(e))}

```

Esta es el archivo de implementación, el cual explica en qué consiste y como debe introducir los datos el usuario5:

```

{# -*- coding: utf-8 -*-
from Ejercicio05 import laberinto

```

```

interruptor = True
while interruptor == True:
matriz= input('')

```

Vamos a avergiuar cuál es la salida de un laberinto formado por una matriz. Para ello, dame primero una matriz booleanana, sin espacios entre los elementos, cuyas filas y columnas formarán las celdas por las que pasará el camino. Si el valor de la celda es True, se le considerará un bloque por el que no se podrá pasar, si es False, se podrá pasar por ahí. Así, las celdas que tengan el valor false deberán formar un camino entre las celdas True, pero sin que

alguna celda False toque lateralmente a más de otra celda False.
Así mismo, una vez ingresado el laberinto, presiona enter e ingresa en forma de tupla las coordenadas de filas y columnas donde se ubicará la entrada del laberinto, que será la celda donde se empezará a recorrer el camino hasta el final. Para dar mayor realismo al ejercicio, procura que la celda de entrada y la de salida sean adyacentes a los bordes de la matriz:

```
'''  
e=input()  
print laberinto(matriz,e)  
interruptor = bool(raw_input("Si desea repetir el proceso ingrese cualquier  
caracter, \nde lo contrario presione enter: "))  
}
```