

Backtracking – III

Clase #28

IIC1103 – Introducción a la Programación

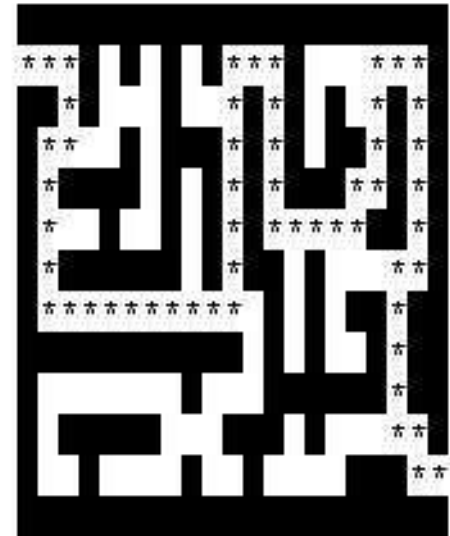
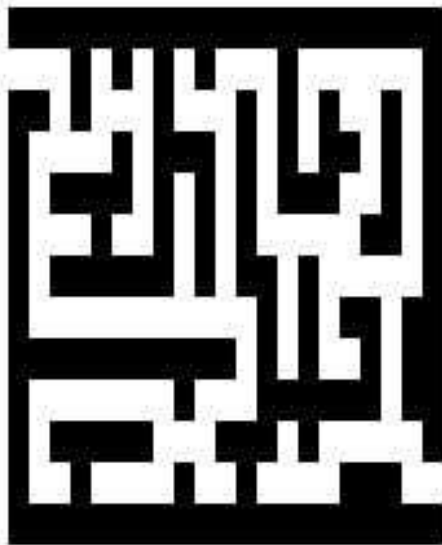
Marcos Sepúlveda (marcos@ing.puc.cl)

Veremos hoy ...

- ▶ Ejemplo Laberinto con backtracking

Laberinto

- ▶ El laberinto a resolver se representa mediante una matriz de caracteres. Su configuración está dada por la inicialización que se haga de dicha matriz.
- ▶ **Objetivo:** buscar la ruta de salida.



Laberinto – descripción general de la solución

► Solución candidata parcial:

- Una **casilla** se representa por una lista de coordenadas. Ej: [2,3]
- Una **solución parcial** corresponde a una lista de coordenadas de casillas que se visitan. Ej: [[0,1], [0,2], [1,2], [1,3], [2,3]]

► Solución inicial:

- Lista que contiene la **casilla de entrada al laberinto**.

► Soluciones candidatas:

- Una solución parcial se extiende al **agregar una casilla adicional**.
- Desde una casilla cualquiera, se puede ir: (el orden no importa) hacia arriba, hacia la derecha, hacia abajo o hacia la izquierda.

► Solución parcial sirve cuando:

- Una solución parcial sirve cuando sólo contiene casillas que: estén dentro del laberinto, no sean muros, y no hayan sido visitadas previamente.

► Solución parcial es final cuando:

- Una **solución final** se encuentra cuando las coordenadas de la última **casilla visitada** corresponde a la **salida del laberinto**.

Backtracking – Laberinto

- Resultado implementación del problema del laberinto usando Backtracking

```
>>>
LABERINTO RESUELTO:

Solución:
[[1, 0], [1, 1], [1, 2], [2, 2], [3, 2],
 [3, 1], [4, 1], [5, 1], [6, 1], [7, 1],
 [7, 2], [7, 3], [7, 4], [7, 5], [7, 6],
 [7, 7], [7, 8], [7, 9], [7, 10], [6, 10],
 [5, 10], [4, 10], [3, 10], [2, 10], [1, 10],
 [1, 11], [1, 12], [2, 12], [3, 12], [4, 12],
 [5, 12], [5, 13], [5, 14], [5, 15], [5, 16],
 [4, 16], [4, 17], [3, 17], [2, 17], [1, 17],
 [1, 18], [1, 19], [2, 19], [3, 19], [4, 19],
 [5, 19], [6, 19], [6, 18], [7, 18], [8, 18],
 [9, 18], [10, 18], [10, 19], [11, 19],
 [11, 20]]

¡tuvimos éxito!

X X X X X X X X X X X X X X X X X X X X
* * * X   X   X   X * * * X   * * * X
X X * X   X   X   * X * X   X   * X * X
X * *   X   X X X * X * X   X X * X * X
X * X X X X   X   X * X * X X X * * X * X
X *   X   X   X   X * X * * * * * X X * X
X * X X X X X X X   X * X X   X   * * X
X * * * * * * * * * *   X   X   X X * X X
X X X X X X X X X X X   X   X   X * X X
X           X           X X X X X X * X X
X   X X X X X   X   X X X   X   * * X
X       X           X           X X X * *
```