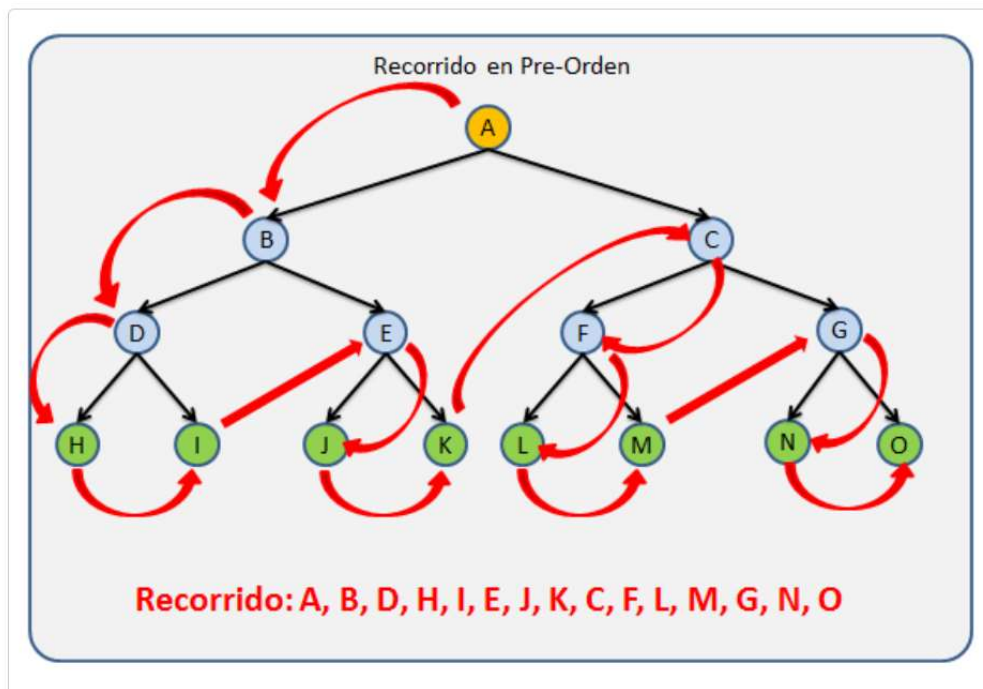


Se desarrollará un programa en Python que recorra un laberinto (generado por el algoritmo backtracking) usando el método recursivo de recorrido en profundidad.

Búsqueda en profundidad

Recorrido Pre-orden: El recorrido inicia en la Raíz y luego se recorre en pre-orden cada uno de los sub-árboles de izquierda a derecha.

Esta definición puede ser un poco compleja de entender por lo que mejor les dejo la siguiente imagen.



El laberinto estará representado por un array bidimensional de **101x101** llamado **maze** e inicializado a cero (blanco).

La solución estará en la posición `maze[100,99]=3` (verde) y se comenzará en la posición `maze[1,1]=2` (naranja). Los bordes de `maze` se rellenarán con 1 (gris).

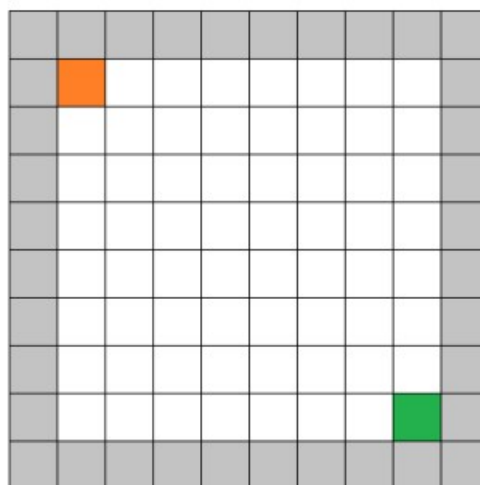


Figura 1- Laberinto vacío de 10x10 con salida en (8,8)

Para realizar el recorrido en profundidad de forma iterativa se usa la estructura de PILA (LIFO).

AYUDA:

<https://www.oscarblancarteblog.com/2014/08/22/estructura-de-datos-arboles/>

Se entregará un único archivo PDF que contenga el notebook de Jupiter con el código fuente, y capturas con ejemplos de ejecución.

1.1 (1 Punto) Índice.

1.2 (3 Puntos) Funcionamiento correcto.

1.3 (2 Puntos) Explicación del funcionamiento y estructuras creadas.

1.4 (2 Puntos) Modifica la matriz de movimientos para permitir saltos en diagonal y añadir más muros (de forma aleatoria) en el interior, olvidar un porcentaje de puntos por dónde se ha pasado.

1.5 (1 Punto) Recorrido en abanico iterativo, indica estructuras creadas.

1.6 (1 Punto) Comparación de los dos algoritmos, dificultades, conclusiones y mejoras (camino más corto).