





Trabajo práctico Nº 2¹

El arqueólogo y aventurero Indiana Croft se encuentra en Busca de la Cuna de la Vida, y finalmente la ha encontrado en el Templo Maldito. El problema es que le esperan un montón de habitaciones llenas de trampas... como ésta.

Indiana Croft está en una sala cuadrada de 64 m² (es decir, de forma 8 x 8) donde cada piedra del suelo (cuadradas, de 1 x 1 metros) tiene un número escrito (0, 1, 2 ó 3). Siguiendo las instrucciones que aparecen en un antiguo diario, Indiana Croft sabe que el número de cada baldosa representa la cantidad de veces que puede ser pisada antes de que se cierre automáticamente la puerta de la habitación. Aún más, el diario también especifica que para poder abrir la salida de la sala es necesario pisar todas las baldosas exactamente el número de veces que tienen grabado. Para poder volver a pisar una baldosa es necesario haber estado en otra antes (es decir, no se puede “saltar” en vertical sobre la misma piedra para conseguir pisarla varias veces; es necesario cambiar de baldosa); Indiana Croft solo puede desplazarse de una baldosa a otra que esté adyacente en horizontal o vertical.

Afortunadamente, Indiana Croft puede contactar contigo por radio, y te ha dictado los números de todas las baldosas, que son los que aparecen en la tabla adjunta (sólo a modo de ejemplo), así como la posición de la entrada () y de la salida () de la habitación. Debes diseñar un algoritmo que encuentre una forma de abrir la puerta de manera que Indiana Croft pueda salir por ella.

							
0	0	0	1	2	2	3	2
0	0	1	0	1	0	2	2
1	2	3	2	2	0	0	0
0	0	0	0	3	3	2	2
0	1	1	2	2	0	0	2
1	3	0	0	1	1	1	3
0	2	1	1	1	1	0	1
0	2	1	0	1	1	0	0
							

¹ Extraído de Cátedra de Algoritmia y Complejidad – Jesus LazaroGarcia – Universidad de Alcalá

Entradas que debe tomar el sistema:

Debe leer el archivo de texto entrada.txt, conteniendo:

En la primer línea la cantidad de filas y la cantidad de columnas de la matriz, separado por coma.

En la segunda línea la fila, y la columna de inicio (sería la primer celda que pisa), los índices comienzan desde 0. Los valores de fila y columna separados por coma.

Tener en cuenta que consideramos que la celda 0,0 se encuentra en la esquina superior izquierda de la matriz.

En la tercer línea, la fila, y la columna de salida (sería la última celda que pisa), los índices comienzan desde 0. Los valores de fila y columna separados por coma.

A partir de la cuarta línea en adelante, la enumeración de los valores de cada celda de la matriz, organizados por fila, una fila por renglón, y separando las columnas por coma.

Por ejemplo, a continuación se muestra un archivo entrada.txt que define una matriz de 3x3, con un valor 2 puesto en todos los casilleros, salvo en una de las diagonales que tendrá valor 1, iniciando en el extremo inferior izquierdo y terminando en el superior derecho:

3,3

2,0

0,2

1,2,2

2,1,2

2,2,1

FIN DE ARCHIVO

Obviamente la matriz no debe ser de tamaño fijo, sino que deberá leerse desde el archivo.

Y puede pasar también que la habitación no sea cuadrada sino que sea rectangular, por ejemplo 10x6.

En caso que lo requiera puede suponer un tamaño máximo de 20x20.

Salidas del sistema:

Archivo de texto llamado salida.txt, que deberá contener:

En la primer línea la cantidad total de estados en el cual entró recursivamente el backtracking, o sea cantidad de nodos del espacio de soluciones que visitó.

En la segunda línea y a partir de ella, si existe solución, se listará la secuencia de pasos desde la entrada hasta la salida que resuelve el problema. Estos pasos se darán como una dupla <fila, columna>por renglón.

En caso que no exista solución debe haber un -1 en la línea 2 y nada más.

Análisis y Diseño de Algoritmos II – Laboratorio 2015

Ejemplo de archivo salida.txt en caso de haber encontrado la solución visitando 154 estados de backtracking :

154

0,2

0,1

1,1

1,2

Y así seguiría con una <fila, columna>por cada paso que debe dar hacia la salida en la solución.

Ejemplo de archivo salida.txt en caso que no encuentre solución luego de pasar por 234 estados de backtracking:

234

-1

Informe requerido:

- Deberá contener el planteo del espacio de soluciones y la forma en que lo va generando.
- Discusión sobre posibles estrategias de poda.
- Diferencia en la cantidad de estados que visita el backtracking ante un mismo archivo de entrada, para el caso con poda y sin poda.