

Programación III Virtual – 2do Cuatrimestre 2021

Trabajo Práctico Obligatorio

Enunciado del problema “Laberinto Mínimo”

Laberinto Mínimo consiste en un tablero de $N \times M$ que en sus posiciones tienen números positivos, excepto en aquellas que no se pueden visitar que tiene el valor -1. El objetivo es partir de una posición dada (X_i, Y_i), llegar a una posición (W_i, Z_i), moviéndose solo a las casillas adyacentes, siempre y cuando no tenga el valor -1 y no haya sido visitada previamente. Además se debe considerar que sea el recorrido debe ser aquel cuya suma de los valores de las casillas visitadas sea **mínima**.

Ejemplo: si tenemos el siguiente tablero, donde el recorrido comienza en la posición (1,0) y finaliza en (3,4)

3	2	1	3	1
5	-1	4	-1	2
-1	8	4	5	1
5	6	-1	2	-1

El resultado optimo sería:

Costo 10

3	2	1	3	1
5	-1	4	-1	2
-1	8	4	5	1
5	6	-1	2	-1

Otros resultados posibles, que no serían el mínimo son:

Costo 14

3	2	1	3	1
5	-1	4	-1	2
-1	8	4	5	1
5	6	-1	2	-1

Costo 16

3	2	1	3	1
5	-1	4	-1	2
-1	8	4	5	1
5	6	-1	2	-1

Costo 12

3	2	1	3	1
5	-1	4	-1	2
-1	8	4	5	1
5	6	-1	2	-1

Costo 11

3	2	1	3	1
5	-1	4	-1	2
-1	8	4	5	1
5	6	-1	2	-1

El problema deberá resolver utilizando la técnica de **bactraking** y utilizando el lenguaje **JAVA**.

Problema a resolver

Se deberá diseñar un algoritmo en donde se tiene como:

Entrada:

1. Un tablero de NxM con los valores en sus posiciones
2. Posición inicial (X,Y)
3. Posición final (Z,W)

Salida:

Lista con las posiciones visitadas.

Se les provee de un proyecto en el cual se incluyen la interface **ResolverLaberintoInterface.java** con el método **resolverLaberinto**, que recibe tres parámetros de entrada, una matriz de entero y dos posiciones (X,Y), y devuelve una lista de las posiciones visitadas. No se puede modificar la entrada ni la salida, conservando los tipos de datos indicados. Solo se pueden agregar métodos adicionales en la clase **ResolverLaberintoImplementacion.java** que implementa la interface mencionada anteriormente, si así se requiere para poder llevar a cabo la resolución.

El proyecto a entregar debe contener:

- Una implementación de la interfaz "**ResolverLaberintoInterface**" que está dentro del paquete dado "**LibreriaTPO2C-LunesVirtual-2021.jar**". El algoritmo implementado debe estar en el método **resolverLaberinto** de la clase "**ResolverLaberintoImplementacion**" dado en el proyecto "**ProyectoTPO2C-LunesVirtual-2021.jar**"

Resolución del enunciado

Se solicita:

- Implementar los métodos necesarios para resolver el problema. Se deberán entregar los archivos **.java** correspondientes.
- Confeccionar el informe según el template que se entrega, en donde deberán indicar la técnica utilizada, explicar la estrategia del algoritmo implementado, el pseudocódigo (no código), el cálculo de complejidad temporal y las conclusiones del trabajo.

Forma de Entrega y Defensa Oral

La entrega del TPO será a través de los grupos asignados:

- La **versión final** de la implementación de la resolución del problema, junto con el informe solicitado deberá ser entregado con **fecha límite: 08/11/2021**. Los grupos que no realicen la entrega **estarán desaprobados**.
- La **defensa oral** se realizará el día **15/11/2021**, **por Teams**, cada grupo tendrá asignado un slot de tiempo para la defensa, todos los integrantes del grupo deben estar presentes durante la defensa (quién no esté tendrá el trabajo desaprobado).

Grupos

El trabajo deberá ser resuelto en forma grupal. Los grupos serán conformados por el docente con cuatro integrantes, o menos dependiendo de la cantidad de alumnos del curso.

Aprobación

Para la aprobación del trabajo práctico obligatorio se deberá ***tener aprobada tanto la entrega del TPO como la defensa oral del mismo.***