

GUIÓN DE LA PRÁCTICA 5

OBJETIVO:

- Algoritmos de programación dinámica

Encontrar el número de caminos a un punto de una ciudad

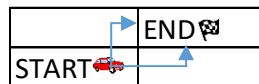
Supongamos que estamos en una ciudad cuya estructura es rectangular y que queremos saber cuántas posibilidades tenemos para ir desde un punto a otro de dicha ciudad.

Tendremos en cuenta las siguientes restricciones:

- Estamos en alguna posición del suroeste de la ciudad y queremos ir a alguna posición más al noreste de dicha ciudad
 - Como tendremos mucha prisa, siempre avanzaremos por las calles hacia el norte (arriba) o hacia el este (derecha)
- Puede ocurrir que nos encontremos con obstáculos que nos impidan el paso por una calle

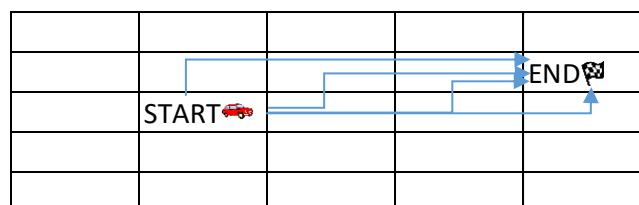
Veamos algunos ejemplos:

Ejemplo 1



En este caso la ciudad es pequeña y queremos saber el número de opciones para desplazarnos desde las coordenadas de START hasta END cumpliendo con las restricciones. El resultado, **obviamente es 2**, ya que primero podemos ir a la derecha y luego arriba o bien primero podemos ir arriba y luego a hacia la derecha.

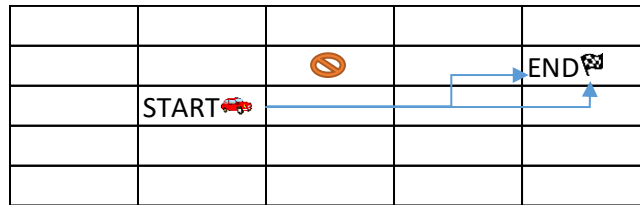
Ejemplo 2



En este caso el número de posibilidades para llegar desde START hasta END **aumenta a 4**:

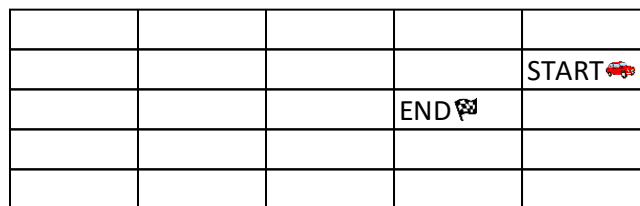
- 1) arriba – derecha – derecha – derecha
- 2) derecha – arriba – derecha – derecha
- 3) derecha – derecha – arriba – derecha
- 4) derecha – derecha – derecha - arriba

Ejemplo 3



Ahora nos encontramos con un obstáculo, lo que impide que pasemos por la coordenada (1,2). Así, nos quedan **2 posibilidades** para llegar a END.

Ejemplo 4



En este caso el resultado dado **será -1**, lo que indica que no es posible desplazarnos desde una posición más al noreste hacia el suroeste. Esto puede verse como una simplificación del algoritmo, el cual podría ampliarse para permitir movimientos al sur (abajo) y al oeste (izquierda).

Esta idea, o variaciones de ella, pueden aplicarse a diferentes dominios, como por ejemplo en videojuegos o robótica, en donde podría ser interesante saber cuántos caminos directos hay desde un punto a otro.

TRABAJO PEDIDO

SE PIDE diseñar e implementar un algoritmo utilizando programación dinámica para resolver este problema de forma óptima.

*Si utiliza **Eclipse**, se creará el proyecto **practica05_PD<3 últimas cifras dni>** con todas las clases necesarias.*

*Si utiliza **JDK**, cree los paquetes necesarios para esta práctica.*

*Las clases necesarias se crearán dentro del paquete **alg<dnipropio>.p5***

- Implementar en Java dicho algoritmo de tal forma que calcule el resultado final de cuántos caminos disponemos para ir desde el punto inicial al final de la ciudad.
- Se dispondrá de 9 ficheros de prueba:
 - caso1.txt – Resultado esperado: 2 opciones
 - caso2.txt – Resultado esperado: 252 opciones
 - caso3.txt – Resultado esperado: 0 opciones
 - caso4.txt – Resultado esperado: 6406484391866534976 opciones
 - caso5.txt – Resultado esperado: 4 opciones
 - caso6.txt – Resultado esperado: 2 opciones
 - caso7.txt – Resultado esperado: -1 (entrada no válida)
 - caso8.txt – Resultado esperado: -1 (entrada no válida)
 - caso9.txt – Resultado esperado: -1 (entrada no válida)

- c) ¿Qué complejidad tiene el algoritmo diseñado? ¿Crees que se podría cambiar el diseño para mejorar la complejidad?

En cuanto a la estructura de los ficheros de texto y tomando como **ejemplo** caso6.txt, su contenido sería el siguiente:

5,5
1,2
2,1;1,4

Representando lo siguiente:

- Primera línea: **tamaño de la ciudad**
 - o El primer 5 sería el ancho de la ciudad (número de calles en sentido horizontal)
 - o El segundo 5 sería el alto de la ciudad (número de calles en sentido vertical)
- Segunda línea: **coordenadas de los diferentes obstáculos** en las calles de la ciudad
 - o En la coordenada x=1 e y=2 hay un obstáculo (podría haber más separados por punto y coma). Si no hubiera obstáculos está línea estaría vacía.
- Tercera línea: **punto inicial y destino**
 - o El camino comienza en la coordenada x=2 e y=1 y deberá finalizar en la coordenada x=1 e y=4

Los ficheros se cargarán de la siguiente forma:

➤ CarreterasCiudad [RUTA_FICHERO]

El programa deberá mostrar los datos leídos originalmente en formato tabla y los resultados finales obtenidos para cada uno de los casos de prueba, incluyendo el estado final de la tabla y el número de posibles caminos desde el origen hasta el destino.

Se entregará:

- *Los ficheros fuente de las clases, que se hayan programado, dentro del paquete o del proyecto Eclipse.*
- *Un documento Word con una pequeña explicación del algoritmo utilizado y sus complejidades. El documento también debe incluir las tablas creadas con programación dinámica para los 9 casos de ejemplo (excepto para el caso 4, ya que resultaría demasiado grande).*

Se habilitará una tarea en el campus virtual para realizar la entrega. El plazo de entrega será de 1 semana y 1 día a partir de finalizar la sesión.