ALGORITMIA

Grado en Ingeniería Informática del Software - Curso 2022-2023

GUIÓN DE LA PRÁCTICA 4

OBJETIVO:

- **Algoritmos ávidos o devoradores**

Plan del Grupo de Intervención Rápida de Defensa

Un ejército enemigo ha desembarcado en las costas de nuestro país invadiendo *n* ciudades.

Los servicios de "inteligencia" están informados de que en cada una de las ciudades invadidas se encuentran e_i efectivos enemigos. Para contraatacar, el *Grupo de Intervención Rápida de Defensa* dispone de *n* equipos listos para intervenir. Cada uno de estos equipos consta de d_j efectivos entrenados para este tipo de acciones y no se pueden pasar efectivos entre equipos. Tenemos que asignar un equipo a cada ciudad, pero para ***garantizar* el éxito** de la intervención en una ciudad, es necesario que dispongamos, al menos, de **tantos efectivos como el enemigo**.

TRABAJO PEDIDO

Diseñar e implementar en Java un algoritmo voraz, para el alto mando de defensa, que proporcione una asignación de cada equipo de intervención a cada ciudad, de forma que se **maximice** el número de victorias.

Describir el **heurístico** empleado en el algoritmo.

Probar el algoritmo implementado para comprobar, que efectivamente, proporciona la **solución correcta** para varios conjuntos de datos de entrada. Se proporcionan varios casos de prueba que utilizan ficheros con el formato utilizado por el código para carga de datos.

¿Qué **complejidad tiene el algoritmo** diseñado? ¿Se podría cambiar el diseño para **mejorar la complejidad**?

Realizar una clase ***DefensaTiempos*** que sirva para ir aumentando el tamaño del problema (n= 10, 20, 40, …, hasta un tamaño grande) y para esos tamaños **mida los tiempos de ejecución medios**. Para medir los tiempos, la generación de enemigos y defensas inicial se hará aleatoriamente, por ejemplo, en rango 0..999 efectivos tanto del enemigo como de los equipos de intervención rápida.

Parte opcional: Si se dispone de la **versión mejorada** del algoritmo para resolver el problema comparar los tiempos. Comprobar si los tiempos obtenidos en el apartado anterior están acordes con la complejidad teórica del algoritmo.

Ficheros proporcionados

Se proporcionan los siguientes ficheros como base para realizar el trabajo pedido de forma correcta:

- Clase principal: Defensa.java
    - Ya están implementados todos los métodos de carga y salida de datos por consola.
    - Se encuentra marcado con la etiqueta //TODO el método de resolución del problema que debe ser implementado con un algoritmo voraz.
- Clase de pruebas unitarias: DefensaTest.java
- Ficheros múltiples para realizar las pruebas:
    - EnemigosXX.txt, contiene los efectivos enemigos de cada ciudad
    - DefensaXX.txt, contiene los efectivos de cada equipo de defensa
    - EsperadaXX.txt, contiene los índices de la asignación de los equipos de defensa a las ciudades

Se entregará:

- *Los ficheros fuente de las clases, que se hayan programado, dentro del paquete o del proyecto Eclipse.*
- *Clases de prueba*
- *Un documento PDF con la tabla de tiempos cubierta y la explicación de la correspondencia con la complejidad teórica*

Se habilitará una tarea en el campus virtual para realizar la entrega. Esta práctica se trabajará durante una sesión y el plazo ***plazo límite es un día antes*** *de la próxima sesión de prácticas de tu grupo.*