

ejercicio 1

José A. Mañas

23.2.2018

ejercicio 1

- algoritmos sobre grafos
 - BFS: camino mínimo (saltos)
 - Dijkstra: camino mínimo (distancias)
 - Edmonds-Karp: flujo máximo
- programar y probar

algoritmos

En un grupo de entrega formado por 1, 2 o 3 alumnos, se suma el último dígito del DNI de cada alumno

- si es par: algoritmo BFS
- si es impar: algoritmo Dijkstra
- opcionalmente se puede elegir Edmonds-Karp

Tareas:

1. buscar el algoritmo en Internet
2. adaptar el código al marco del enunciado

ficheros

- javadoc

facilitados	del alumno
<ul style="list-style-type: none">• class Screen• class Example_1• class Example_2• class GraphTest0	<ul style="list-style-type: none">• class Graph• class Node• class Link
<ul style="list-style-type: none">• class BFSTest0	<ul style="list-style-type: none">• class BFS• class BFSTest
<ul style="list-style-type: none">• class DijkstraTest0	<ul style="list-style-type: none">• class Dijkstra• class DijkstraTest
<ul style="list-style-type: none">• class EdmondsKarpTest0	<ul style="list-style-type: none">• class EdmondsKarp• class EdmondsKarpTest

tareas

1. preparar pruebas: xxxTest
2. preparar las implementaciones
 - class xxx
3. poner a punto usando las pruebas
4. entregar

buscar en Internet

- ¿Se puede buscar una solución en Internet?
 - sí
- ¿Qué es lo que debo buscar?
 - opción 1: busque el algoritmo explicado en lenguaje natural (ej. en español)
 - y prográmelo
 - probablemente necesite usar `java.util.List<>` y `java.util.Map<>`
 - opción 2: busque el algoritmo resuelto en java
 - tendrá que envolver el algoritmo adaptando sus estructuras de datos de Graph a las de la solución encontrada

pruebas

- venga de donde venga la solución, ¡pruébela!
- si esta mal, la responsabilidad será del alumno
- probablemente
 - los casos singulares (situaciones límite) los tenga que diseñar
 - para casos normales puede recurrir a los mil ejemplos resueltos en Internet
 - en todo caso debe evitar escenarios en los que haya varias soluciones correctas
 - (las pruebas de vuelven más complejas sin aportar más seguridad al código)

reparto de trabajo en grupo

- si el grupo es de 2 o más alumnos, conviene repartir tareas, al menos deben ser diferentes personas las que preparen los casos de prueba y las que preparen la solución
- luego se entrega todo junto

entrega

- package es.upm.dit.adsw.ej1
 - *.java
 - o sea, todos los ficheros de ese paquete