

Prácticas de Algorítmica.

3º de Grado en Ingeniería Informática.

Curso 2019-2020.

Práctica 4.

Objetivos.

Con esta práctica se pretende que el alumno implemente un algoritmo basado en la técnica de los algoritmos voraces.

Enunciado:

El alumno deberá implementar un algoritmo voraz para obtener una solución aproximada al problema del viajante de comercio utilizando el algoritmo voraz visto en el último apartado del tema 6.

Para su realización se suministra el material correspondiente a las prácticas de grafos que usasteis en las prácticas de Estructuras de Datos y un ejemplo de uso de dicho material. El programa principal está contenido en el archivo **test.cpp**.

En el ejemplo suministrado se implementa un algoritmo (incluido en el archivo **random_graph_algorithm.hpp**) mediante el cual se crea un grafo a partir de otro grafo que ha sido cargado desde un fichero. El nuevo grafo contendrá los mismos nodos que el anterior y solo los lados cuyo peso sea mayor que 200.

Dado que la solución del problema del viajante de comercio serían los lados que conforman el camino mínimo, y que estos lados enlazan a todos los nodos del grafo, la solución obtenida se almacenará en un nuevo grafo.

Para implementar la práctica habrá que crear un archivo similar al **random_graph_algorithm.hpp**, denominado **traveling_salesman_greedy_algorithm.hpp**. Este archivo contendrá una única función cuyo prototipo será **template <class T> traveling_salesman_greedy(WGraph<T> & g, WGraph<T> & newGraph)**, donde **g** será el grafo original y **newGraph** será el grafo que contenga la solución del problema.

Por otra parte, en el programa principal, la función de medio nivel **random_graph_algorithm()**, será sustituida por la función **traveling_salesman_greedy_algorithm()**. El código de ambas será similar, lo único que hay que cambiar sería la llamada:

```
create_random_wgraph_from_wgraph(*g, newGraph);
```

por la llamada

```
traveling_salesman_greedy(*g, newGraph);
```

La función ya implementada **template <class T> bool fold_wgraph(std::ostream& out, WGraph<T> & g)** (contenida en el archivo **graph_utils.hpp**) será la que se usará para mostrar el grafo solución.

Nota: Para realizar las pruebas se suministra el fichero **Andalucia.txt** correspondiente a las capitales de provincia andaluzas. En la información de cada nodo se proporciona la latitud y longitud de cada uno. Esta información no se usará para nada en la aplicación.

Fecha de comienzo: 23 de octubre.

Fecha máxima de entrega: 10 de noviembre.