

Prácticas de Algorítmica.

3º de Grado en Ingeniería Informática.

Curso 2020-2021.

Práctica 4.

Objetivos.

Con esta práctica se pretende que el alumno implemente un algoritmo basado en la técnica de la programación dinámica.

Enunciado:

El alumno deberá implementar un algoritmo basado en programación dinámica y en el recorrido en profundidad de un grafo para obtener una solución óptima al problema del viajante de comercio.

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 que utiliza el recorrido en profundidad de un grafo para mostrar todos los caminos posibles que salen de un nodo y recorren el resto de nodos (incluido en el archivo **deep_travel_graph_algorithm.hpp**).

En la salida del ejemplo, sólo se muestran las etiquetas de los nodos que componen dichos caminos. Dado que se guardan todos los caminos que salen de un nodo, los podéis aprovechar para obtener los ciclos que pasan por todos los nodos e ir seleccionando el mejor en cada momento. Evidentemente, para ello tendréis que ir acumulando los pesos de los lados que estáis usando para obtener los distintos ciclos.

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 **deep_travel_graph_algorithm.hpp**, denominado **traveling_salesman_dynamic_algorithm.hpp**. Este archivo contendrá una función principal cuyo prototipo será **template <class T> traveling_salesman_dynamic(WGraph<T> &g, WGraph<T> &newGraph)**, donde **g** será el grafo original y **newGraph** será el grafo que contenga la solución del problema. *Esta función principal será la que realice las llamadas a la función recursiva que se encargará de realizar los recorridos pertinentes para obtener todos los posibles ciclos y seleccionar el de menor distancia.*

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

deep_travel_wgraph(*g);

por la llamada

traveling_salesman_dynamic(*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: 10 de octubre.

Fecha máxima de entrega: 24 de noviembre.