

Prácticas de Estructuras de Datos.
2º de Grado en Informática.
Curso 2013-2014.

Práctica 4. Grafos. Caminos y distancias mínimas.

Objetivos: Con esta práctica se pretende que el alumno implemente los siguientes algoritmos:

1. Algoritmo para obtener el camino y la distancia mínima entre todos los pares de nodos en un grafo (dirigido o no dirigido).
2. Algoritmo para obtener el árbol abarcador de coste mínimo en un grafo conexo no dirigido.

Material suministrado:

El archivo **main.cpp** contiene el menú de opciones para cargar el grafo o para aplicarle los algoritmos de Floyd o el de Prim, y las funciones invocadas desde el **main.cpp** son funciones de medio nivel y están contenidas en el archivo **funciones.hpp**. Estas funciones son las que se encargarán de invocar a las funciones de la clase **Grafo** (implementa un grafo por medio de la matriz de conexión) y de la clase **AlgoritmosGrafos** (implementa las funciones correspondientes a los algoritmos de Floyd y Prim). El archivo **grafo.hpp** contiene a la clase **Grafo** y el archivo **algoritmosgrafos.hpp** contiene a la clase **AlgoritmosGrafos**. Los archivos **matrizAndalucia.txt** y **matrizEspana.txt** contienen la matriz de conexión de la red de carreteras que enlaza las capitales de provincia de Andalucía y de la península ibérica y los archivos **Andalucia.txt** y **Espana.txt** contienen los nombres de las capitales de provincia.

Enunciado:

En esta práctica se realizarán los siguientes apartados:

1. Para la clase **Grafo** se da su implementación completa. Hay que mirar el código, ver la utilidad de cada una de sus funciones y su implementación.
2. El archivo **funciones.hpp**, con las funciones de medio nivel también está implementado, solo tenéis que ver el código y comprender lo que hace cada función.
3. Implementación del algoritmo de Floyd para obtener las distancias y caminos mínimos en un grafo cualquiera en la clase **algoritmosGrafos**. Para ello se implementarán dos funciones:
 1. Función **floyd**, a partir de la cual se obtendrá la matriz de distancias y de intermedios.
 2. Función **caminoFloyd**, a partir de la cual se obtendrá el camino mínimo entre el nodo origen y un nodo cualquiera.
4. Implementación del algoritmo de Prim para obtener el árbol abarcador de coste mínimo en un grafo conexo no dirigido en la clase **algoritmosGrafos**. Para ello se implementará la función **prim**.

Notas:

1. Los nodos se numeran desde 0.
2. La clase grafo utiliza dos plantillas (templates) una es **G_Nodo** (indica el tipo genérico que define la etiqueta asociada a cada nodo) y la otra es **G_Lado** (indica el tipo genérico que se va a almacenar como peso del lado).
3. Para realizar las pruebas se suministran dos ejemplos, correspondientes a la red de carreteras andaluza y la española (solo almacena como nodos las capitales de provincia). Las matrices de conexión están almacenadas en los archivos **matrizAndalucia.txt** y **matrizEspana.txt**, y los nombres de las capitales están en los archivos **Andalucia.txt** y **Espana.txt**.
4. Los archivos de las capitales son archivos de texto que contienen los nombres de las capitales.
5. Los archivos de la matriz de conexión son también de texto y en la primera línea aparece el número de nodos, en la segunda aparece un valor que es 0 (grafo no dirigido) o 1 (grafo dirigido) y en las siguientes líneas aparece la matriz de conexión almacenada por filas. El valor 32000 indica que no existe lado entre los dos nodos correspondientes.

Comienzo de la práctica: Semana del 31 de Marzo al 4 de Abril.

Tiempo de realización: 3 semanas.

Entrega voluntaria de la práctica: Semana del 28 de Abril al 2 de Mayo.

