

Metodología y Tecnología de la Programación.
Prácticas del primer cuatrimestre
Curso 2.010-2.011

PRÁCTICA 1.

La página web de una agencia de viajes desea añadir una pequeña aplicación para uso y disfrute de sus clientes. Su idea fundamental consiste en automatizar la consulta de trayectos posibles entre ciudades de una región minimizando el tiempo de viaje. Podemos considerar la región de la siguiente manera:

- En la región existe un conjunto finito de ciudades.
- Existen diversos medios de transporte entre ciudades: avión, autobús, barco, etc.
- Para cada par de ciudades, (A,B), sale desde A hacia B cada día, a lo sumo, un medio de transporte. También, desde B hacia A, sale cada día, a lo sumo un medio de transporte.
- Son conocidos los horarios de salida y las duraciones de todos los viajes de todos los transportes existentes.

Diseñar una aplicación con las siguientes funcionalidades:

1. Modificar el conjunto de las ciudades.
2. Modificar el transporte entre dos ciudades.
3. Establecer el camino más rápido entre dos ciudades elegidas, si existe, indicando la siguiente información:
 - Hora de salida y llegada tanto de las ciudades origen y destino como de las escalas en ciudades intermedias.
 - Tiempo de espera en la ciudad origen y en las ciudades intermedias.
 - Tipo de medio de transporte utilizado.

PRÁCTICA 2.

Diseña un algoritmo que generalice el algoritmo de ordenación por mezcla de tal forma que se procesen k sublistas (donde k es un parámetro de entrada al algoritmo).

PRÁCTICA 3.

Objetivo: Consideremos un mar y las islas que contiene. Supongamos que la navegación por este mar es tan peligrosa que los barcos necesitan no alejarse demasiado de los faros que pueda haber en algunas islas. Deseamos diseñar un programa que determine el camino más corto entre dos posiciones, si es que es posible, para un mapa dado.

Representación de los datos:

Utilizamos una matriz M de la siguiente forma:

$$M(i, j) = \begin{cases} 0, & \text{Existe agua en la posición (i,j);} \\ 1, & \text{Existe tierra en la posición (i,j);} \\ 2, & \text{Existe un faro en la posición (i,j).} \end{cases}$$

El programa:

Diseña una aplicación con Builder C++, utilizando una algoritmo de vuelta atrás, que determine el camino de menor longitud entre dos posiciones que debe seguir un barco. El camino debe ser seguro: en cada posición del viaje la distancia a algún faro debe de ser inferior a R .