

Estructuras de Datos (Gestión), curso 2007/08

Segunda Práctica

INTRODUCCIÓN

Uno de los problemas más famosos en ciencias de cómputo es el **problema del viajante**, que consiste en encontrar la ruta óptima (aquella que minimiza la distancia recorrida) para recorrer un conjunto de n ciudades (representadas por puntos en un plano bidimensional) de manera que sólo se pase una vez por cada una ciudad y se regrese al punto de partida (esto es, se recorra un ciclo). Aunque sencillo de enunciar, se cree que éste problema no se puede resolver en un tiempo polinómico, y que no existe una solución mucho mejor que la de enumerar las $n!$ rutas posibles y escoger la óptima.

Una variante poco conocida es la del **problema del reparto aéreo**: En este problema, una avioneta de reparto debe recorrer también un ciclo de n ciudades, lanzando paquetes en cada una de ellas, pero debido a que es importante que durante el viaje el sol se encuentre en cola, el ciclo se divide en dos partes: **ruta de ida** y **ruta de vuelta**. En la ruta de ida (por la mañana) se debe ir hacia el oeste y en la ruta de vuelta (por la tarde) se debe ir hacia el este. Al igual que en el problema del viajante se parte de una ciudad (en este caso la situada más hacia el este), y se deben recorrer todas sin pasar dos veces por la misma ciudad y terminando en la ciudad de origen. El objetivo es minimizar la distancia total recorrida. Es posible (o no) que ésta versión del problema del viajante admita una solución más eficiente.

DESCRIPCIÓN DETALLADA

La entrada del problema consiste en una lista de datos de n ciudades. Para cada ciudad se conoce su posición horizontal, su posición vertical y un identificador.

La salida del problema consiste en mostrar el ciclo más económico (el que minimice la distancia total recorrida) que cumpla las siguientes condiciones:

- La **ciudad origen** (por la que se comienza) es aquella situada más al este, es decir la de **mayor** posición horizontal.
- Ruta de ida**: Desde la ciudad origen se va pasando por ciudades cada una de ellas más al oeste que la anterior (cada ciudad debe tener una posición horizontal **menor** que la anterior), hasta llegar a la ciudad situada más al oeste (aquella con la posición horizontal más pequeña).
- Ruta de vuelta**: Una vez fijada la ruta de ida la ruta de vuelta es inmediata: Desde la ciudad situada más al oeste (**ciudad extremo**) se recorren el resto de ciudades por las que no se pasó en la ruta de ida en **orden creciente** de posición horizontal, hasta llegar a la ciudad origen.

Ejemplo: En la *figura 1* se muestra un conjunto de puntos correspondientes a las capitales de Castilla y León. La *ciudad de origen* sería en este caso Soria, la de mayor coordenada horizontal, y la *ciudad extremo* Zamora, la de menor coordenada horizontal. En las *figuras 2 y 3* se muestran dos posibles soluciones del problema:

- En la *figura 2* la ruta sería **SO-BU-SG-P-AV-VA-LE-SA-ZA-SO**. En la ruta de ida se pasa por todas las ciudades (desde Soria hasta Zamora) y en la de vuelta por ninguna, simplemente regresamos a la ciudad origen. Esta ruta no es óptima.

- En la *figura 3* la ruta sería **SO-BU-P-VA-LE-ZA-SA-AV-SG-SO**. En la ruta de ida se pasa por Soria, Burgos, Palencia, Valladolid, León y Zamora. En la de vuelta por Salamanca, Ávila y Segovia, desde donde se regresa a la ciudad de origen. Esta ruta es óptima, con una distancia total de 861.8 km.

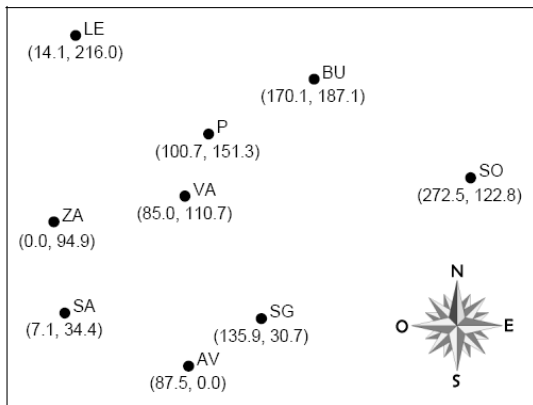


Fig. 1

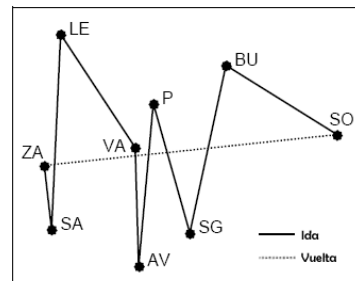


Fig. 2

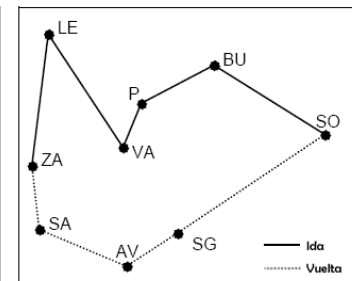


Fig. 3

La distancia entre dos ciudades se calcula de la manera habitual: La distancia entre una ciudad con coordenadas (x_1, y_1) y otra con coordenadas (x_2, y_2) es $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

REQUISITOS DEL PROGRAMA

Se debe crear un programa que resuelva, *de la manera más eficiente posible*, el problema planteado, y puesto éste programa se compilará y se someterá a una batería de pruebas en un proceso automático, es importante que tanto la entrada de datos como la salida de resultados sean **exactamente** como se describe a continuación:

Al comienzo del programa éste no debe mostrar ningún mensaje, pantalla de bienvenida, etc. Tan sólo leerá de la **entrada estándar** lo siguiente:

- La primera línea contendrá un número entero, n , que indicará el número de ciudades del problema.
- A continuación seguirán n líneas cada una de ellas consistirán en dos números reales (posición horizontal y vertical) y una cadena de texto (identificador de la ciudad). Los tres datos estarán separados entre sí por espacios en blanco.

El programa calculará la solución óptima del problema y escribirá en la **salida estándar** lo siguiente (nuevamente sin mensajes explicativos ni información extra):

- Una línea donde se escribe la distancia de la ruta óptima.
- En la(s) línea(s) siguientes se describe la ruta escribiendo los identificadores de las ciudades por donde se pasa separados entre sí por guiones (fijaros que la ciudad origen debe ser la primera y la última). No debe hacerse ningún intento por formatear esta cadena para que tenga una apariencia “bonita” en pantalla.

Se supondrá que las entradas son correctas y no se hará ningún intento por validarlas.

No se debe suponer que las ciudades están ordenadas por algún criterio.

Si lo necesitáis podéis suponer que el número de ciudades, n , será siempre menor que 1000, y que los identificadores de las ciudades no superan los 30 caracteres.

Ejemplo: Con el problema planteado en la *figura 1*, la ejecución del programa mostraría lo siguiente:

| | | |
|------------------------------|---|---------|
| 9 | } | Entrada |
| 87.5 0.0 AV | | |
| 170.1 187.1 BU | | |
| 14.1 216.0 LE | | |
| 100.7 151.3 P | | |
| 85.0 110.7 VA | | |
| 135.9 30.7 SG | | |
| 272.5 122.8 SO | | |
| 0.0 94.4 ZA | | |
| 7.1 34.4 SA | | |
| 861.8 | } | Salida |
| SO-BU-P-VA-LE-ZA-SA-AV-SG-SO | | |

DOCUMENTACIÓN QUE SE DEBE ENTREGAR

Se deberá entregar el código fuente del programa (se permiten los lenguajes Eiffel, Java, Free-Pascal o Delphi/Kylix) y un documento (en formato Word, Látex (DVI) o Acrobat PDF) donde se indique lo siguiente:

- Nombre de los alumnos que han realizado la práctica.
- Descripción de la estrategia de diseño utilizada para resolver la práctica.
- Una estimación del orden del algoritmo utilizado en función de n , el número de ciudades. Si no podéis obtener una estimación teórica hacer un análisis experimental parecido al de la primera práctica.

Estos datos se enviarán como documentos adjuntos en un **correo electrónico** enviado a la dirección cvaca@infor.uva.es, donde el título del mensaje será **PRAC-ED-2** y el contenido el nombre de los alumno(s) que la han realizado, el día **24 de enero de 2008** antes de las **23:59**.