

Estructuras de Datos (Sistemas), curso 2005/06

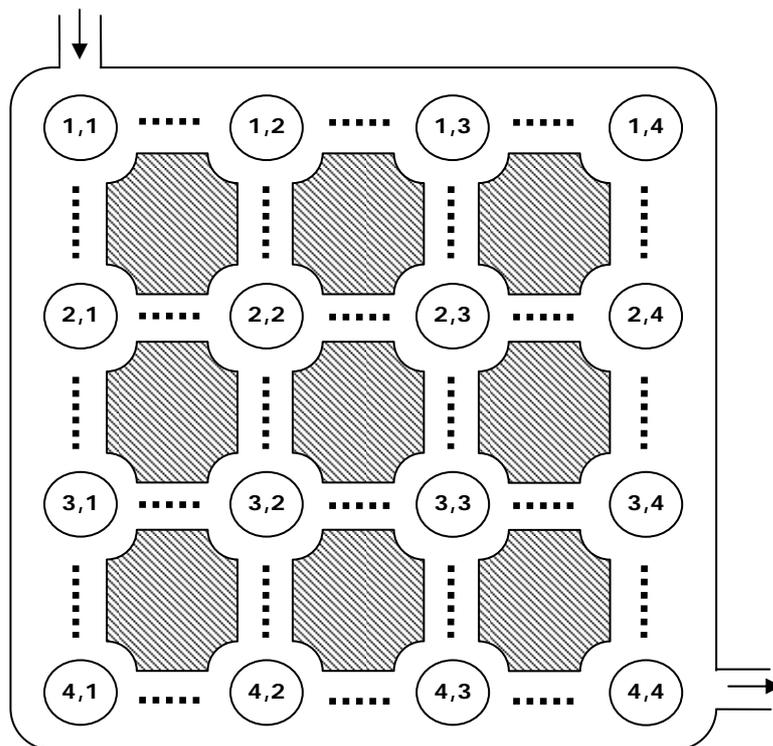
Primera Práctica – Ruta óptima (I)

Introducción

Uno de los problemas más interesantes en computación (del cuál hablaremos con más detalle al llegar al tema de grafos) es el de calcular la ruta óptima entre dos puntos.

En esta práctica vamos a restringir el problema al cálculo de la ruta óptima para atravesar una urbanización y analizaremos la eficiencia de soluciones obtenidas mediante fuerza bruta y backtracking.

La forma de la urbanización va a ser extremadamente simple: Tiene forma cuadrada y consiste en $(n-1)^2$ bloques de casas conectados mediante calles que se unen en una rotonda central, de la manera indicada en el siguiente diagrama donde $n = 4$:



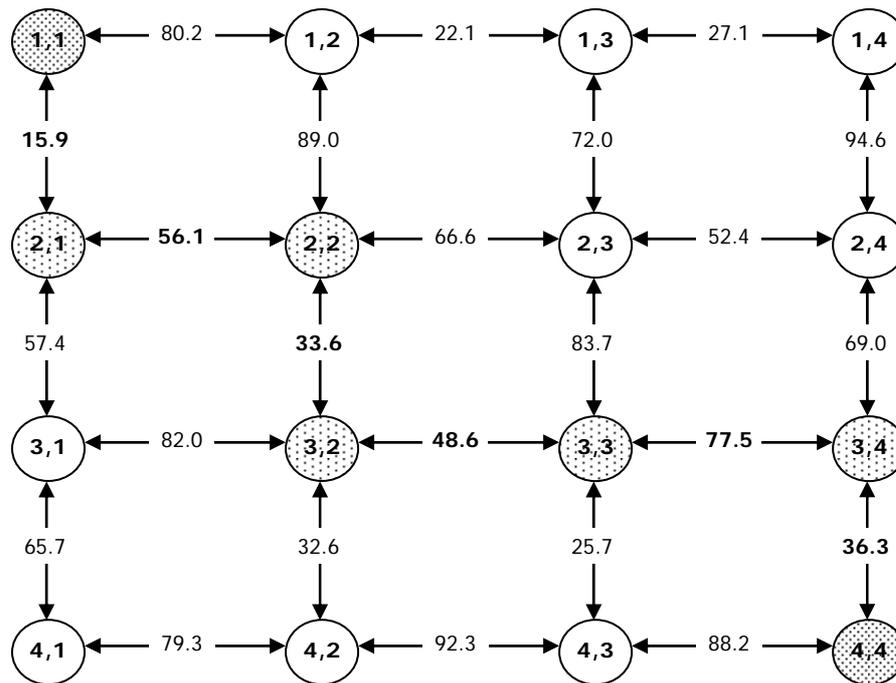
Se puede apreciar que existen n^2 rotondas, indexadas por la fila y columna donde se encuentran. Cada rotonda (salvo las de los límites de la urbanización) conecta cuatro calles, y para atravesar la urbanización se debe seguir una ruta que vaya de la rotonda (1,1) a la rotonda (n,n). Desde una rotonda (i, j) que no esté en los límites de la urbanización se puede acceder a las rotondas $(i-1, j)$, $(i, j+1)$, $(i+1, j)$ e $(i, j-1)$ por las calles correspondientes.

El problema consiste en calcular cuál es el **tiempo mínimo** que se tarda en atravesar la urbanización, yendo de la rotonda (1,1) a la rotonda (n,n) (**Nota:** No es necesario conocer la ruta óptima, tan sólo el tiempo de la ruta óptima).

Se cumplen las siguientes propiedades:

- Se tarda un **tiempo nulo** en pasar por cualquier rotonda.
- Sin embargo recorrer una calle (definida como el tramo de carretera entre dos rotondas) requiere un determinado tiempo, **que no es el mismo para todas**. De hecho la entrada del problema consiste en dar los tiempos de recorrido de las $2 \cdot n \cdot (n-1)$ calles de la urbanización.
- Se tarda lo mismo en recorrer una calle en un sentido que en el contrario.
- Para fijar los límites del problema supondremos que $n < 100$ y que los tiempos de recorrido de las calles son valores reales entre 10 y 100 segundos.

En la siguiente figura se muestra, de forma más esquemática, el diagrama anterior incluyendo los tiempos de recorrido de cada calle:



Con estos valores, el tiempo óptimo es de 268.0 segundos, y la ruta óptima pasa por las rotondas sombreadas (recordar que no es necesario obtener la ruta óptima, tan solo su tiempo).

Descripción Detallada

Se deberán crear dos programas que resuelvan el problema anterior, uno debe utilizar la estrategia de fuerza bruta y el otro la estrategia de backtracking, ambas implementadas utilizando recursividad. Ambos programas reciben como entrada el valor n que indica el tamaño de la urbanización y los tiempos de recorrido de cada calle de acuerdo al siguiente formato:

- Una línea con el valor de n .
- Siguen n líneas con $n-1$ valores reales separados por espacios que representan los tiempos necesarios para recorrer las calles horizontales.
- Siguen $n-1$ líneas con n valores reales separados por espacios que representan los tiempos necesarios para recorrer las calles verticales.

Las líneas representan filas (de menor a mayor) y en cada fila el orden es columnas de menor a mayor.

La salida consistirá en dos líneas. La primera es un número real que indica el tiempo de la ruta óptima (el menor tiempo posible para recorrer la urbanización desde la rotonda (1,1) a la rotonda (n,n)), y la segunda es un número entero que indica el número de llamadas recursivas que ha realizado en total el programa para obtener el resultado (ese valor se utilizará para estimar la eficiencia de los programas).

Para el ejemplo descrito anteriormente, la entrada que recibe el programa y la salida que debe proporcionar serían las siguientes (el número de llamadas es ficticio):

Datos de entrada:

4
80.2 22.1 27.1
56.1 66.6 52.4
82.0 48.6 77.5
79.3 92.3 88.2
15.9 89.0 72.0 94.6
57.4 33.6 83.7 69.0
65.7 32.6 25.7 36.3

Salida del programa:

268.0
1234

Nota: Tanto la entrada como sobre todo la salida del programa deben respetar exactamente los formatos descritos, ya que serán evaluados de forma automática por otro programa utilizando redirección. En concreto los datos de entrada se reciben por la entrada estándar (como si se introdujeran por teclado, abstenerse de pedir nombres de fichero o mostrar mensajes del tipo “pulse una tecla para comenzar”). Respecto a la salida, evitar mensajes de salutación, dedicatorias u otras florituras: debe ser exactamente tal como se describe.

Programa basado en fuerza bruta

El programa basado en fuerza bruta generará y medirá todas las rutas posibles desde (1,1) hasta (n,n). Es importante darse cuenta que, debido a que los tiempos de recorrido de las calles siempre van a ser mayores que cero, **no se deben tener en cuenta rutas que contengan ciclos** (aquellas en que se pase más de una vez por alguna rotonda), ya que es imposible que puedan ser la ruta óptima (recorrer el ciclo tarda tiempo, y se llega al mismo punto de comienzo de él).

Se sugiere utilizar un enfoque recursivo, donde el problema se plantee de la siguiente forma: Para ir desde (i, j) hasta (n,n) nos movemos desde (i, j) hasta sus rotondas accesibles no visitadas todavía y desde esas rotondas, recursivamente, intentamos llegar a (n,n).

El programa debe llevar la cuenta del número de llamadas recursivas realizadas en total para presentar ese valor como resultado. Cada llamada recursiva debe realizar operaciones no recursivas que tarden un tiempo $O(1)$.

Programa basado en backtracking

El programa basado en backtracking debe ser una variante del basado en fuerza bruta donde además se añada algún criterio de poda. Se sugiere añadir como mínimo el criterio de que si estamos generando una ruta y nos llegamos en la rotonda (i, j), y el tiempo desde (1,1) a (i, j) ya es mayor o igual que el mejor tiempo calculado hasta el momento para ir de (1,1) a (n,n) entonces la ruta actual no puede ser la óptima (todavía no hemos llegado al destino y existe otra ruta que sí ha llegado tardando menos). Se valorará la inclusión de otros criterios adicionales, aunque se recomienda comprobar cuidadosamente que los criterios no puedan eliminar posibles rutas óptimas.

Estudio de la eficiencia

Una vez que disponemos de los programas y se ha comprobado que funcionan correctamente (se proporcionarán casos de prueba en la página de la asignatura), el último paso es intentar obtener una estimación del **tiempo promedio** de ambos programas en función de n .

Se intentará ajustar el tiempo promedio a una cota del tipo $\Theta(\alpha^n)$ o del tipo $\Theta(\alpha^{n^2})$, donde α es un valor real. Por lo tanto para **cada** programa (independientemente del otro) se debe analizar cuál de las dos fórmulas da un mejor ajuste y determinar el valor de α .

Se sugiere que para realizar éste análisis, ya sea manualmente o con la ayuda de otro programa (que no es necesario presentar) se haga lo siguiente:

- Se obtenga una lista de número de llamadas promedio para distintos valores de n . Para ello se fija un valor de n , se ejecuta el programa muchas veces (cuantas más mejor) dándole como entrada urbanizaciones de tamaño n y tiempos de recorrido aleatorios y se promedian los números de llamadas realizados en cada ejecución.
- A partir de esa lista se intenta obtener el mejor ajuste a una función del tipo $c \cdot \alpha^n$ o $c \cdot \alpha^{n^2}$ (c es una constante arbitraria) utilizando una hoja de cálculo, otro programa o cualquier medio que se desee.

Presentación y evaluación de la práctica

En la página web de la asignatura (<http://www.infor.uva.es/~cvaca/asigs/eds.html>) se proporcionan detalles sobre la manera en que se evaluará la parte práctica de la asignatura en ambas convocatorias.

- Los programas se podrán realizar en los lenguajes Pascal (o su variante Delphi), Eiffel o Java, en todos los casos utilizando únicamente las librerías/clases básicas.
- La práctica puede ser realizada por un grupo de 1, 2 ó 3 personas matriculadas en la asignatura, aunque el número de personas que forman el grupo tendrá impacto en su calificación, como se explica en la página de la asignatura.
- Antes del 30 de abril de 2006 se debe comunicar, ya sea personalmente al profesor o mediante correo electrónico a la dirección cvaca@infor.uva.es, la formación o disolución de grupos de más de una persona. A partir de esa fecha no se permitirá la inclusión de ninguna persona en otro grupo (aunque sí el abandono o expulsión).
- Antes del 25 de mayo de 2006 los grupos enviarán a la dirección de correo antes indicada el código fuente de los programas, indicando en el texto del mensaje el nombre de los componentes del grupo.
- La evaluación de las prácticas será presencial. Los grupos presentarán los programas, el resultado del análisis y responderán a las preguntas del profesor, en una fecha, lugar y orden de presentación que se indicará posteriormente (cuando se conozca la composición de los grupos). La fecha de presentación será igual o posterior al 25 de mayo.