



Yendose por las ramas

Enunciado

El objetivo del trabajo práctico es resolver los problemas propuestos de diferentes maneras, realizando posteriormente una comparación entre los diferentes algoritmos utilizados.

Se debe:

1. Describir los problemas a resolver dando ejemplos de los mismos y sus soluciones.

Luego, por cada método de resolución:

2. Explicar de forma clara, sencilla, estructurada y concisa, las ideas desarrolladas para la resolución del problema. Para esto se pide utilizar pseudocódigo y lenguaje coloquial combinando adecuadamente ambas herramientas (**¡sin usar código fuente!**). Se debe también justificar por qué el procedimiento desarrollado resuelve efectivamente el problema.
3. Deducir una cota de complejidad temporal del algoritmo propuesto (en función de los parámetros que se consideren correctos) y justificar por qué el algoritmo desarrollado para la resolución del problema cumple la cota dada.

4. Dar un código fuente claro que implemente la solución propuesta.

El mismo no sólo debe ser correcto sino que además debe seguir las *buenas prácticas de la programación* (comentarios pertinentes, nombres de variables apropiados, estilo de indentación coherente, modularización adecuada, etc.).

Por último:

5. Realizar una experimentación computacional para medir la performance de los programas implementados, comparando el desempeño entre ellos. Para ello se debe preparar un conjunto de casos de test que permitan observar los tiempos de ejecución en función de los parámetros de entrada, analizando la idoneidad de cada uno de los métodos programados para diferentes tipos de instancias.

Para cada problema se listan los algoritmos que se deben considerar.

Solo se permite utilizar `c++` como lenguaje para resolver los problemas. Se pueden utilizar otros lenguajes para presentar resultados.

La entrada y salida de los programas **deberá hacerse por medio de la entrada y salida estándar del sistema**. No se deben considerar los tiempos de lectura/escritura al medir los tiempos de ejecución de los programas implementados.

Deberá entregarse un informe impreso que desarrolle los puntos mencionados.

Problemas

Hiperconectados

La planificación de las ciudades es una herramienta fundamental para mitigar los problemas del crecimiento poblacional. Al tener muchas ciudades, un problema clásico a resolver es el del diseño de redes de conexión entre las mismas, ya sea para los medios de transporte como los automóviles o los trenes, así como para los diferentes tipos de servicios básicos como el agua potable o la electricidad. En este caso resolveremos un problema asociado a la conectividad del servicio más básico e importante para la humanidad, internet.

Tenemos un conjunto de n ciudades, para las cuales queremos crear una red de conexiones que cumpla que todo par de ciudades está conectado por al menos un camino de nuestra red. Para lograr esto, tenemos m potenciales conexiones entre pares de ciudades. Cada potencial conexión posee tres atributos: dos ciudades a y b , que indica simétricamente que ciudades serían conectadas y c que representa el costo de instalar dicha conexión. Nuestro objetivo es crear la red indicada de manera tal que el costo total de la misma sea el mínimo, para lo cual ya sabemos que existen algoritmos que nos pueden ayudar específicamente con dicha tarea. Sin embargo, nuestro problema no se reduce simplemente a esto. Si bien las conexiones tienen un valor asociado que nos indica cuanto nos costará su instalación, hay otras variables asociadas a las conexiones que se están ocultando por una cuestión de simplicidad. Una conexión podría tener un costo de instalación bajo, pero a la vez podría ser una conexión no muy deseada por encontrarse en una zona sísmica, lo cual aumentaría las chances de tener que ser sometida a reparaciones. También podría haber una conexión con costo asociado bajo, pero que su instalación genere un descontento público por el lugar donde debe realizarse la instalación.

Es por estas situaciones que más allá de encontrar la forma menos costosa de armar una red entre las n ciudades tal que cada par de ciudades se encuentre conectado de al menos una manera (situación que denominaremos Anexión General Minúscula o AGM por sus siglas), tendremos aparte el siguiente problema. Por nuestros conocimientos en el área sabemos que la solución AGM del problema no tiene porque ser única, queremos saber entonces por cada posible conexión si la misma tiene la propiedad de 1) Estar en todo AGM del problema 2) No estar en ningún AGM del problema 3) Estar en algún AGM del problema.

Parámetros y formato de entrada/salida

La entrada consistirá de una primera línea con dos enteros n y m , correspondientes a la cantidad de ciudades y a la capacidad de posibles conexiones, respectivamente. Luego le sucederán m líneas con tres enteros, a_i , b_i y c_i . a_i y b_i indicarán las ciudades a conectar, mientras que c_i indicará el costo asociado. Asumiremos que todos los valores en la entrada serán enteros positivos.

La salida consistirá de m líneas. Cada línea tendrá un *string* que debe indicar si la posible

conexión i está en *toda* solución, *alguna* solución, o *ninguna*.

Algoritmos

Para la solución del problema, se deben considerar los siguientes algoritmos en sus resoluciones intermedias:

- Algoritmo de Prim.
- Algoritmo de Kruskal.
- Algoritmo de Kruskal con *Path Compression*.

Hiperauditados

Como encargados de la instalación de la red de internet mencionada en el anterior problema, debemos asegurarnos que las conexiones estén funcionando una vez que la red se encuentre operativa. Dada la importancia de la tarea, no podemos delegar la responsabilidad, ni podemos confiar en alguna herramienta (como un *ping*) para saber si todas las ciudades se encuentran correctamente conectadas. No tenemos otra opción que ir específicamente a la ciudad que queremos auditar su conectividad para hacer las pruebas *in situ*. Dada esta situación, queremos saber la mejor forma de movernos entre dos ciudades gastando la menor cantidad de plata posible en combustible. En nuestro mapa tenemos n ciudades y m rutas bidireccionales que conectan pares de ciudades. Cada ruta tiene un valor l_i que indica la cantidad de litros de nafta que necesita nuestro vehículo para recorrerla. A su vez, cada ciudad tiene un costo asociado c_i que indica el precio de un litro de nafta en dicha ciudad. Si sabemos que nuestro auto tiene un tanque de nafta con capacidad máxima para 60 litros, queremos saber cual es el costo mínimo para movernos entre un par de ciudades dadas.

Parámetros y formato de entrada/salida

La entrada consistirá de una primera línea con dos enteros n y m , correspondientes a la cantidad de ciudades y a la cantidad de rutas, respectivamente. A continuación, habrá n líneas. Cada línea tendrá un entero c_i , representando el costo del litro de combustible en la ciudad i . Luego le sucederán m líneas con tres enteros, a_i , b_i y l_i . a_i y b_i indicará las ciudades conectadas por la ruta, mientras que l_i indicará la cantidad de litros necesarios para recorrerla. Asumiremos que todos los valores en la entrada serán enteros positivos.

La salida consistirá de $n * (n - 1) / 2$ líneas. Cada línea tendrá tres enteros a_i , b_i , s_i . a_i y b_i debe indicar las ciudades consideradas, cumpliéndose que $a_i < b_i$. s_i indicará el costo mínimo necesario para ir de una ciudad a otra. Las líneas en la salida deberán estar ordenadas según el orden lexicográfico de las ciudades pertinentes.

Algoritmos

Para la solución del problema, se deben considerar los siguientes algoritmos en sus resoluciones intermedias:

- Algoritmo de Dijkstra.

- Algoritmo de Dijkstra con *priority queue*.
- Algoritmo A* (Solo para grafos que cumplan la desigualdad triangular).
- Algoritmo de Bellman-Ford.
- Algoritmo de Floyd-Warshall.
- Algoritmo de Dantzig.

Fechas de entrega

- *Formato Electrónico:* Domingo 27 de Mayo de 2018, hasta las **23:59 hs**, enviando el trabajo (informe + código) a la dirección `algo3.dc@gmail.com`. El subject del email debe comenzar con el texto [TP2] la lista de apellidos de los alumnos.
- *Formato físico:* Lunes 28 de Mayo de 2018, a las 18 hs. en la clase teórica.

Importante: El horario es estricto. No se considerarán los correos recibidos después de la hora indicada.