Algoritmos y Estructuras de Datos III
Segundo Cuatrimestre 2019
Trabajo Práctico 2

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Modelando problemas reales con grafos

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. **En caso de que el algoritmo resuelva de manera exacta 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

Separando la paja del trigo

En *aprendizaje por computadora* uno de los problemas más fundamentales es el de *clusterizar*. Clusterizar significa agrupar un conjunto de objetos en diferentes subconjuntos por alguna nocion de similaridad intraconjunto (o disimiliradad interconjunto). Por ejemplo, si tenemos una nube de puntos en un espacio métrico, nos gustaría realizar divisiones (*clusters*) que agrupen a los conjuntos de puntos que se encuentran juntos.

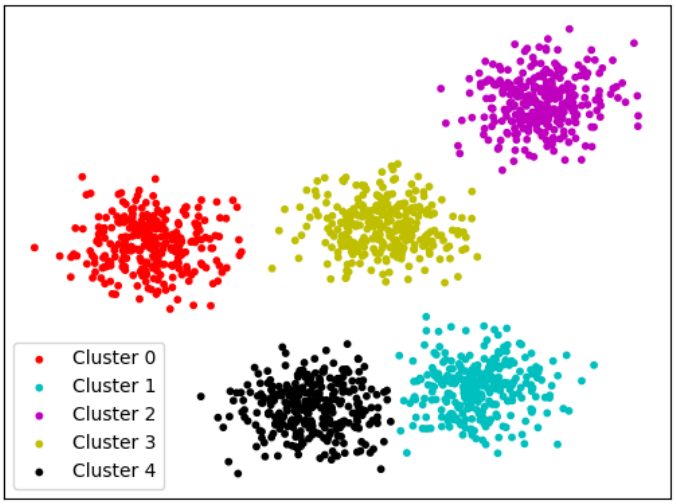

Figura 1: Puntos clusterizados

En la figura 1 vemos un ejemplo de este problema. En esta figura los clusters son evidentes a simple vista, aunque existen algunos puntos que no son tan fáciles de clasificar.

En otros casos los clusters pueden parecer claros intuitivamente pero difíciles de separar de una manera algorítmica. El problema de clustering resulta de gran interés ya que son muchas las aplicaciones en donde se puede utilizar. Puntos como los de la figura 1 pueden representar situaciones muy variadas como dígitos manuscritos que queremos identificar o pacientes que se sometieron a un determinado análisis médico.

En nuestro caso, buscaremos abordar este problema de una perspectiva de grafos, utilizando la noción de Árbol Generador Mínimo para generar los clusters deseados.

Parámetros y formato de entrada/salida

La entrada consistirá de una primera línea con un entero n, correspondiente a la cantidad de puntos a clusterizar. Luego le sucederán n líneas con dos números reales, x_i, y_i. x_i y y_i indicarán las coordenadas del punto i.

La salida consistirá de n líneas. Cada linea i tendrá un número que debe indicar a qué cluster pertenece el punto i.

Algoritmos

Para resolver el problema de clusterización se deberá utilizar las nociones presentadas en la sección VII de [1]. Dentro de la idea presentada en dicho trabajo, se debe experimentar con la noción de *eje inconsistente* para generar diferentes tipos de clusterizaciones.

Se debe utilizar las dos definiciones de eje inconsistente utilizadas en las secciones VII y XVII de [1]. Ambas definiciones poseen parámetros que se pueden variar para conseguir diferentes soluciones al problema.

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*.

Para la experimentación, se recomienda utilizar instancias conocidas como las que se pueden encontrar en $http://cs.joensuu.fi/sipu/datasets/$

Arbitraje

En el mundo de las finanzas se denomina *arbitraje* al hecho de comprar y vender un recurso de manera simultanea para generar una ganancia, aprovechandose de los desbalances de los precios en diferentes mercados. La idea es que al ser operaciones simultaneas el arbitrajista no corre riesgo en sus transacciones. En general, los desbalances necesarios para darse una situación de arbitraje provienen de desincronizaciones de los mercados.

Supongamos que hay un recurso R disponible para vender y comprar en dos mércados M_1 y M_2. En M_1 nos dan 56.80 pesos por vender una unidad de R y nos cuesta 57.50 pesos comprar una unidad. En M_2 en cambio nos dan 56.90 pesos por vender una unidad de R y nos cuesta 57.65 pesos comprar una unidad. En este caso no existe oportunidad de arbitraje. No hay forma con estos dos mercados que vendamos y compremos unidades del recurso R y generemos una ganancia. Sin embargo, en un determinado momento del día sucede un evento que valoriza el recurso R. M_1 se entera de este suceso y actualiza sus cotizaciones. Luego de este cambio nos dan 57.67 pesos por vender una unidad de R y nos cuesta 59.95 pesos comprar una unidad. Las noticias llegan más tarde a M_2 por lo que todavía no se entero del suceso que valorizó a R, por los que sus precios siguen intactos por un período más. En este caso sí existe oportunidad de arbitraje. Podemos simultaneamente comprar varias unidades del recurso R en M_2 y vender varias unidades en M_1. De esta manera, por cada unidad que se compra/vende estamos obteniendo 0.02 pesos sin ningún tipo de riesgo.

Afortunadamente, esta situación no es algo que suceda a menudo. De hecho, en el caso de que suceda, muchas veces no incluye un único recurso sino que involucra a varios recursos en varios mercados distintos.

En este problema lo que nos gustaría saber es si existe oportunidad de arbitraje entre varias divisas (Pesos, Dolares, Libras, Reales, Bitcoins, etc.). Para eso contamos con la tasa de cambio entre cada par de divisas. En caso de existir arbitraje, queremos saber que divisas son las involucradas en el proceso.

Parámetros y formato de entrada/salida

La entrada consistirá de una primera línea con un entero n, correspondiente a la cantidad de divisas diferentes a tener en cuenta. A continuación, habrá n líneas. Cada línea tendra n números reales c_{ij}, representando el multiplicador que se debe aplicar a una unidad de la divisa i al cambiar a la divisa j.

La salida consistirá de 1 línea. Al comenzar la línea se indicará si existe oportunidad de arbitraje. Si no existe, se debe imprimir el string NO y finalizar la salida en ese lugar. Si existe, se debe imprimir el string SI, seguido del ciclo de divisas que producen el arbitraje.

Algoritmos

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

- Algoritmo de Bellman-Ford.
- Algoritmo de Floyd-Warshall.

Fechas de entrega

- *Formato Electrónico:* Jueves 24 de Octubre de 2019, 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:* Viernes 25 de Octubre de 2019, a las 18 hs.

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

Referencias

[1] C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. on Computers*, C-20(1):68–86, jan 1971.