

Teoría de gráficas

Programa 2.

Profesor: M. C. Jorge Antonio García Galicia

Fecha de entrega: Viernes 28 de octubre del 2011

Enunciado

Dada una red formada por un grafo G junto con una función de costo $f : V(G) \rightarrow \mathbb{R}^+$. El *árbol de expansión mínimo* es aquel árbol T tal que $V(T) = V(G)$ y que $\sum_{\forall e_i \in E(T)} f(e_i)$ tenga el valor mínimo posible.

Debes realizar un programa que el usuario pueda introducir un grafo G y una función f y debe imprimir al usuario un árbol de expansión mínima T de la red formada por f y G , así como el costo del árbol $\sum_{\forall e_i \in E(T)} f(e_i)$.

Sugerencias

- Usa el algoritmo de Prim que vimos en clase.
- Puedes pedir al inicio el orden p y el tamaño q . Luego pide la matriz de adyacencia de preferencia de manera eficiente (pide solo los elementos debajo de una diagonal), mientras vas pidiendo la matriz de adyacencia cada que el usuario diga que hay una arista pide el valor f de esa arista.
- Lo mas conveniente es usar una lista de adyacencia para formar el grafo, pero puedes trabajar con la matriz de adyacencia.
- Puedes visitar [http://en.wikipedia.org/wiki/Graph_\(data_structure\)](http://en.wikipedia.org/wiki/Graph_(data_structure)) y http://en.wikipedia.org/wiki/Prim's_algorithm para darte ideas.
- O consultar el capítulo 9 sección 4 (Árboles de expansión mínima) de [2] y el capítulo 23 sección 2 (Algorithms of Kruskal and Prim) de [1].

Aspectos a calificar

Los aspectos a calificar son los siguientes.

- La correctez del programa. Debe funcionar bien para todas las redes que reciba de entrada.
- La robustez del programa. Que sea capaz de manejar algunas entradas invalidas del usuario.
No están obligados a manejar excepciones (aunque pueden contar como puntos extras) así que el programa puede simplemente terminar en error ante una entrada invalida.
- El estilo de la programación. Si siguen o no patrones de IS, si el código es legible y compacto, si siguen la filosofía del lenguaje (es decir que el programa este bien estructurado, si el lenguaje es estructurado o totalmente orientado a objetos si el lenguaje es orientado a objetos).
- La portabilidad. Que el programa *compile* bien en cualquier arquitectura y plataforma. Si usan C apearse al ANSI C, si usan Java apagarse al API oficial, etc.

Entrega

La entrega es por correo electrónico a la dirección que aparece al final de este documento. Se deben de entregar los archivos fuente. Y en el cuerpo del correo deben venir los integrantes del equipo (3 personas máximo) e instrucciones sobre como usar el programa.

El lenguaje a usar es libre, pero recuerden que lo probare en un ambiente GNU/Linux y tal vez en uno Windows. Así que algunas sugerencias son:

- C
- C++
- Java
- Matlab u Octave
- Python
- Perl
- bash
- C# (apegarse al estándar mono)

Correo del profesor: nemediano@gmail.com

Sitio web del curso: <http://www.nemediano.com.mx/teoriaGrafos>

Referencias

- [1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. McGraw-Hill, The MIT Press, 2 edition, 2001.
- [2] Richard Johnsonbaugh. *Matemáticas Discretas*. Pearson, Prentice Hall, 6 edition, 2005.