PRÁCTICAS DE LA ASIGNATURA MATEMÁTICA DISCRETA.
Ingenierías Técnicas de Informática de Gestión y de Sistemas.
2º Curso. Primer Cuatrimestre.
Curso 2006-2007

Tema 3: TEORÍA DE GRAFOS.

1. Crea una función que reciba la matriz de un grafo y devuelva el número de aristas. (fácil)
2. Crea una función que devuelva un vector con el grado de cada vértice. (fácil)
3. Crea una función que compruebe si dos grafos pueden ser isomorfos a partir de los grados de los vértices. (fácil-[2])
4. Modifica la función "sonIsomorfos" dada en los ejemplos para que sea más eficiente, teniendo en cuenta los grados de los vértices. (fácil)
5. Crea una función que reciba dos matrices de dos grafos y si son isomorfos, devuelva el isomorfismo. (medio)
6. Crea una función que reciba la matriz de un grafo y compruebe si K_3 es un subgrafo. (medio)
7. Crea una función que reciba la matriz de un grafo y delvuelva si es completo o no. (fácil)
8. Crea una función que reciba la matriz de un grafo y delvuelva si es regular o no. (medio-[2])
9. Crea una función que reciba la matriz de un grafo y devuelva si tiene un circuito euleriano o no. (fácil-[2])
10. Crea una función que reciba la matriz de un grafo y devuelva si tiene un camino euleriano o no. (fácil-[2])
11. Crea una función que reciba la matriz de un grafo y si este es euleriano, devuelva un circuito euleriano. (difícil-[9])
12. Crea una función que reciba la matriz de un grafo y si este admite un camino euleriano, devuelva un camino euleriano. (difícil-[10])
13. Crea una función que reciba la matriz de un grafo y si este admite un camino euleriano, devuelva un camino euleriano. (fácil-[11+29])
14. Crea una función que reciba la matriz de un grafo y los índices de dos vértices y compruebe si están conectados (por un camino (simple)). (fácil)
15. Crea una función que reciba la matriz de un grafo y compruebe si hay un ciclo. (fácil)
16. Crea una función que reciba la matriz de un grafo y devuelva si este es conexo o no. Podéis hacerlo comprobando que la componente conexa de 1 es todo el grafo, o aplicando la suma y producto de matrices vista en clase. (fácil)

17. Crea una función que reciba la matriz de un grafo y devuelva el número de componentes conexas. Podéis hacerlo mediante la función vista en los ejemplos, con una función recursiva, o a partir de la suma y producto de matrices vista en clase. (medio)

18. Crea una función que reciba la matriz de un grafo, compruebe si tiene un punto de corte (ver ejercicios) y devuelva si puede ser Hamiltoniano o no. (medio-[16])

19. Crea una función que reciba la matriz de un grafo y compruebe si puede ser Hamiltoniano por el criterio dado en clase. (muy difícil-[17])

20. Crea una función que reciba la matriz de un grafo y devuelva si es Hamiltoniano o no. (muy difícil)

21. Crea una función que reciba la matriz de un grafo y devuelva, si existe, un ciclo Hamiltoniano. (muy difícil)

22. Crea una función que reciba la matriz de un grafo y devuelva si puede ser plano aplicando el primer corolario visto en clase. (fácil)

23. Crea una función que reciba la matriz de un grafo y devuelva si puede ser plano aplicando el segundo corolario visto en clase. (fácil-[6])

24. Crea una función que reciba la matriz de un grafo y devuelva si es un árbol o no a partir de su definición. (fácil-[15])

25. Crea una función que reciba la matriz de un grafo y devuelva si es un árbol o no. Usad los criterios que aparecen en los ejercicios que relacionan el número de aristas y vértices y la conexión para caracterizar los árboles. (fácil-[16])

26. Crea una función que reciba la matriz de un grafo y devuelva si tiene ciclos o no, comprobando que cada componente conexa es un árbol. (medio)

27. Crea una función que reciba la matriz de un grafo etiquetado (la matriz en lugar de tener unos, tiene el coste de cada arista) y aplique el algoritmo de Dijsktra para obtener la distancia entre el primer y el último vértice. (difícil)

28. Crea una función que reciba la matriz de un digrafo etiquetado (la matriz en lugar de tener unos, tiene el coste de cada arista) y dos vértices y aplique el algoritmo de Dijsktra para obtener la distancia entre los dos vértices. (difícil)

29. Crea una función que reciba la matriz de un grafo etiquetado (la matriz en lugar de tener unos, tiene el coste de cada arista) y dos vértices y aplique el algoritmo de Dijsktra para obtener el camino más corto entre los dos vértices.

    Para guardar el camino más corto, creamos un vector *anterior* con tantos elementos como vértices tenga el grafo. Inicializamos ese vector a 0, excepto para el vértice inicial, que lo inicializamos con el número de dicho vértice (ej: si el vertice inicial es el 3, $anterior(3) = 3$. Cada ver que actualicemos $L(j)$ como $L(i) + d(i,j)$, actualizamos anterior a $anterior(j) = i$.

Finalmente, para construir el camino, sólo tenemos que partir del vértice final e ir mirando cuáles son los vértices anteriores hasta llegar al primero.

Nótese que si todos los costes son uno, el algoritmo de Dijkstra devuelve un camino simple que une los dos vértices. (difícil)

30. Crea una función que reciba la matriz de un digrafo etiquetado (la matriz en lugar de tener unos, tiene el coste de cada arista) y dos vértices y aplique el algoritmo de Dijsktra para obtener el camino más corto entre los dos vértices (Véase ejercicio 29). (difícil)

31. Crea una función que reciba la matriz de un grafo y devuelva si es plano o no.(muy, muy difícil)

32. Crea una función que reciba la matriz de un grafo y diga si es bipartito o no. (muy difícil)

33. Crea una función que reciba la matriz de un grafo y lo coloree por el algoritmo de Welch y Powell. (difícil)

Nota: Los números entre corchetes indican ejercicios anteriores que simplifican la programación.

Reparto por grupos:

Cuando hay varios ejercicios separados por una barra, elegid uno.

Los ejercicios 4,5, 19, 20, 21, 31 y 32 son opcionales para todos los grupos.

Comprobad que tenéis asignados cinco ejercicios, 3 fáciles, uno medio y uno difícil y que todos los ejercicios de los que dependen también los tenéis asignados.

- Grupo 1: 2, 11/12/13, 15, 24/25, 26
- Grupo 2: 2, 7, 8, 14, 27/28/29/30
- Grupo 3: 7, 14, 18, 22, 33
- Grupo 4: 2, 11/12/13, 14, 18, 22
- Grupo 5: 1, 6, 15, 23, 27/28/29/30
- Grupo 6: 2, 3, 6, 16, 33
- Grupo 7: 2, 11/12/13, 10, 24/25, 26
- Grupo 8: 2, 10, 17, 24, 27/28/29/30
- Grupo 9: 2, 15, 24/25, 26, 33
- Grupo 10: 2, 11/12/13, 16, 17, 23
- Grupo 11: 2, 3, 6, 16, 27/28/29/30
- Grupo 12: 2, 7, 8, 14, 33
- Grupo 13: 2, 11/12/13, 15, 18, 24
- Grupo 14: 1, 7, 16, 17, 27/28/29/30
- Grupo 15: 2, 14, 18, 23, 33
- Grupo 16: 2, 11/12/13, 14, 18, 23
- Grupo 17: 7, 14, 18, 22, 27/28/29/30
- Grupo 18: 2, 14, 18, 22, 33
- Grupo 19: 2, 11/12/13, 16, 24/25, 26

Normas de entrega:

1. Al finalizar la práctica, se entregarán únicamente los archivos fuentes, correspondientes a las funciones que se tenían que codificar y a un programa principal en el que se probarán las funciones.

2. En cada uno de los fuentes que se entreguen, se pondrá en la cabecera un comentario con el número del grupo y los nombres y apellidos de los integrantes.

3. Cada archivo debe estar convenientemente explicado mediante comentarios. Se valorará positivamente que se utilicen los resultados de la Teoría de Grafos para aumentar la eficiencia de los programas. En ese caso se debe incluir un comentario con el resultado que se haya utilizado.

4. Los archivos se guardarán en un directorio cuyo nombre será

$$Grupo + ngrupo + inicialesapellidos.$$

   Por ejemplo, si vuestro grupo es el 12 y los apellidos son Álvarez, Fernández y Rodríguez, el directorio se llamará $grupo12afr$.

5. El directorio se comprimirá en un archivo con el mismo nombre y la extensión correspondiente al programa de compresión utilizada (a ser posible, zip, tar, rar o gz).

6. El archivo comprimido se enviará a *trinidad@unex.es*, antes de las doce de la noche dela fecha indicada en la página web de la asignatura.