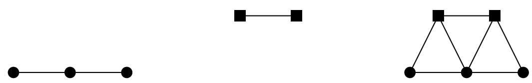
ALGORITMOS Y ESTRUCTURAS DE DATOS III - 1er Parcial
Fecha examen: 12-OCT-2013 / Fecha notas: a determinar

	Nº Orden	Apellido y nombre	L.U.	Cant. hojas[1]
Completar:				
	Nota (Nº)	Nota (Letras)	Docente	
No completar:				

1. Sea $\mathbb{S} = \{S_1, S_2, \ldots S_n\}$ una familia de subconjuntos de un conjunto S, es decir, $S_i \subseteq S$ para $i = 1, 2, \ldots n$. Se define el grafo de intersección de $\mathbb{S}$ como el grafo que por cada S_i tiene un vértice, y tal que los vértices correspondientes a S_i y a S_j son adyacentes si y sólo si $S_i \cap S_j \neq \emptyset$. Sea G cualquier grafo. Demostrar que G es el grafo de intersección de alguna familia de subconjuntos de algún conjunto.

2. (a) Exhibir todos los árboles autocomplementarios no isomorfos. Justificar.
   (b) Un grafo simple se dice 1-árbol si es un árbol con un eje agregado.
   Exhibir todos los 1-árboles autocomplementarios no isomorfos. Justificar.

3. Dado un grafo G, se define su grafo total $T(G)$ como el grafo que tiene un vértice por cada vértice y cada eje de G, y tal que dos vértices de $T(G)$ son adyacentes si y sólo si los elementos correspondientes de G son adyacentes o incidentes. Dado un grafo con ejes G, se define su grafo de líneas $L(G)$ como el grafo que tiene un vértice por cada eje de G, y tal que dos vértices de $L(G)$ son adyacentes si y sólo si los ejes correspondientes de G tienen un extremo en común. En la siguiente figura aparecen, de izquierda a derecha, un grafo G, $L(G)$ y $T(G)$.

   Sea G un grafo con $m \geq 1$ ejes y n vértices de grados $d_1, d_2, \ldots d_n$.
   (a) Determinar el grado de cada vértice de $L(G)$ en función de los grados de los vértices de G. Justificar.
   (b) Demostrar que $L(G)$ tiene m vértices y $-m + \sum_{i=1}^{n} d_i^2/2$ ejes.
   (c) Determinar el grado de cada vértice de $T(G)$ en función de los grados de los vértices de G. Justificar.
   (d) Demostrar que $T(G)$ tiene $m + n$ vértices y $2m + \sum_{i=1}^{n} d_i^2/2$ ejes.
   (e) Demostrar que G y $L(G)$ son subgrafos de $T(G)$.

4. El gobierno de la ciudad decidió construir una nueva calle entre algún par de esquinas a fin de agilizar el tránsito vehicular en la hora pico. El gobierno tiene una lista de calles candidatas que podrían ser construídas, pero no sabe cuál elegir, de modo que te convocaron para resolver el problema. Sin embargo, a vos lo único que te interesa es viajar rápido desde tu casa hasta la facultad, por lo que querés elegir la calle candidata cuya construcción reduciría lo máximo posible la cantidad de calles que hay que recorrer en ese viaje.

   La ciudad está formada por n esquinas y m calles. Cada esquina se identifica por un entero distinto entre 1 y n. Tu casa está en la esquina 1 y la facultad en la esquina n. Cada calle conecta determinado par de esquinas y se puede recorrer desde la primera hacia la segunda. Se puede ir de cualquier esquina a cualquier otra recorriendo las calles, pasando eventualmente por esquinas intermedias.

   Diseñar un algoritmo eficiente para elegir la calle candidata cuya construcción reduciría lo máximo posible la cantidad de calles que hay que recorrer para ir desde tu casa hasta la facultad. La entrada del algoritmo es la cantidad n de esquinas, la cantidad m de calles actuales, la cantidad c de calles candidatas, y para cada calle (actual o candidata) su esquina inicial y su esquina final. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(m + c)$.

5. Sea $v = (v_1, v_2, \ldots v_n) \in \mathbb{N}^n$ un vector conteniendo los valores de las monedas en circulación, y sea $c = (c_1, c_2, \ldots c_n) \in \mathbb{N}^n$ otro vector conteniendo las respectivas cantidades disponibles de monedas de cada valor. Diseñar un algoritmo eficiente basado en programación dinámica que determine la mínima cantidad de monedas que permiten formar un importe dado $w \in \mathbb{N}$. Si es imposible formar ese importe, el algoritmo debe indicarlo. Ejemplos:

   - Para $n = 2$, $v = (5, 1)$, $c = (1, 1000)$ y $w = 10$, el resultado es 6.
   - Para $n = 2$, $v = (5, 2)$, $c = (1, 3)$ y $w = 10$, es imposible formar el importe w.

   Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(w \sum_{i=1}^{n} c_i)$ y espacial $O(w)$.

[1] Incluyendo a esta hoja. Entregar esta hoja junto al examen.