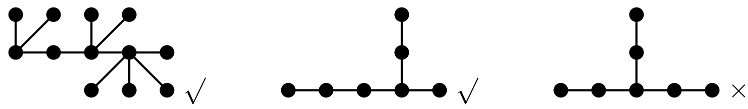
ALGORITMOS Y ESTRUCTURAS DE DATOS III - 1$^{\text{er}}$ Parcial
Fecha examen: 03-OCT-2015 / Fecha notas: a determinar

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

1. Un grafo oruga es un árbol que tiene un camino simple dominante, es decir, tiene un camino simple tal que todo vértice está a lo sumo a un eje de distancia del mismo. En la siguiente figura los dos primeros grafos son grafos oruga, mientras que el tercero no lo es.



Dada una colección de intervalos sobre la recta real, su grafo de intervalos es el grafo que tiene un vértice por cada intervalo, y tal que dos vértices son adyacentes si y sólo si los intervalos correspondientes tienen intersección no vacía. Decimos que G es un grafo de intervalos si y sólo si existe una colección de intervalos tal que G es su grafo de intervalos.

(a) Demostrar que si G es un grafo oruga entonces es un grafo de intervalos. 1.2 p.

(b) Exhibir un grafo de intervalos que no sea un grafo oruga. Justificar. 0.8 p.

2. Determinar para qué valores de n, p, q y h los siguientes grafos son grafos oruga. Justificar. 2 p.

(a) K_n

(b) $K_{p,q}$

(c) árbol binario completo de altura $h \geq 0$

3. Un punto de corte de un grafo es un vértice del mismo tal que al removerlo se obtiene un grafo con más componentes conexas.

(a) Sea G un grafo conexo no trivial, y sea v uno de sus vértices. Demostrar que v es hoja en algún árbol generador de G si y sólo si v no es punto de corte de G. 1 p.

(b) Sea G un grafo conexo no trivial. Usar el primer punto para demostrar que G tiene al menos 2 vértices que no son puntos de corte. 0.5 p.

(c) ¿Sigue valiendo la propiedad del punto anterior si G no es conexo? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar. 0.5 p.

4. Un coleccionista de automóviles está planeando hacer un recorrido por la ciudad manejando su vehículo Ford modelo T fabricado en 1910. A cada cuadra q de la ciudad, el coleccionista le ha asignado un costo $c(q)$ y un beneficio $b(q)$, los cuales son números positivos que dependen del gasto de combustible, el riesgo para el automóvil (para el costo), el impacto en el público, y la belleza del paisaje (para el beneficio). Dado un recorrido R, su costo $c(R)$ es la suma de los costos de las cuadras recorridas, y análogamente su beneficio $b(R)$ es la suma de los beneficios de las cuadras recorridas. Un recorrido R tiene un balance costo-beneficio aceptable si y sólo si $c(R)/b(R) < k$ para cierto número positivo k. El coleccionista quiere saber si existe un recorrido que comience y termine en una misma esquina, y que tenga un balance costo-beneficio aceptable. 2 p.

La ciudad está formada por n esquinas y m cuadras. Cada esquina se identifica por un entero distinto entre 1 y n. Cada cuadra 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 cuadras, pasando eventualmente por esquinas intermedias.

Diseñar un algoritmo eficiente que decida si existe un recorrido como el que quiere hacer el coleccionista. La entrada del algoritmo es la cantidad n de esquinas, la cantidad m de cuadras, y para cada cuadra su esquina inicial, su esquina final, su costo y su beneficio. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

5. Rolando está por jugar a su juego de rol favorito, en el cual existen n personajes. Cada personaje p está caracterizado en el juego por dos atributos enteros positivos $r(p)$ y $s(p)$ que son respectivamente la resistencia y la sabiduría que tiene p. No hay dos personajes con la misma resistencia, ni hay dos personajes con la misma sabiduría. Un personaje p_1 domina a un personaje p_2 si y sólo si $r(p_2) < r(p_1)$ y $s(p_2) < s(p_1)$. Al comenzar la partida Rolando debe elegir el subconjunto de los n personajes que va a utilizar durante el juego. Rolando desea que sea posible ordenar a los personajes que elija "de menor a mayor", es decir, de manera tal que cada uno domine al anterior (excepto el primero). 2 p.

Diseñar un algoritmo que determine la máxima cantidad de personajes que Rolando puede elegir respetando sus deseos. El algoritmo debe tener complejidad temporal $O(n^2)$ y espacial $O(n)$, y estar basado en programación dinámica. La entrada del algoritmo es la cantidad de personajes, y para cada personaje su resistencia y su sabiduría. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar.

SUGERENCIA: Ordenar a los personajes por uno de los atributos y pensar cómo quedan ordenados por el otro atributo los subconjuntos deseables. Aplicar programación dinámica sobre los personajes ya ordenados.

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