SOLUCIONES

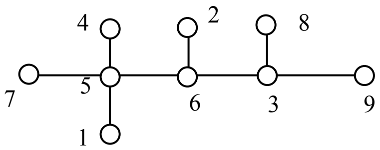
1. **(1,5 puntos)**
   (a) Un árbol T de orden 50 tiene sus vértices de grados 1, 3 ó 6. Al contar sus hojas obtenemos un valor entre 9 y 11. ¿Cuántas hojas tiene el árbol T?

   (b) Construye el árbol etiquetado T' cuyo código de Prüfer es [5, 6, 5, 5, 6, 3, 3]. Construye otro árbol NO etiquetado con la misma sucesión de grados que T' y que no sea isomorfo a T'.

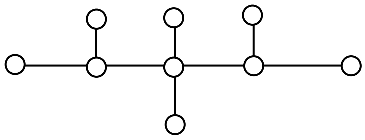
Sol.:
(a) La fórmula de Euler $\Sigma d(v) = 2q$ se convierte en nuestro caso en
$t_1 + 3t_3 + 6t_6 = 2(50 - 1) = 98$, siendo t_j el número de vértices de grado j.
Por tanto, $3t_3 + 6t_6 = 98 - t_1$ (*)
El primer miembro es múltiplo de 3, por lo que $98 - t_1$ también debe serlo. Así el único valor válido de los propuestos es $t_1 = 11$

Pero entonces la expresión (*) queda así: $3t_3 + 6t_6 = 87$, es decir, $t_3 + 2t_6 = 29$
Y como $t_1 + t_3 + t_6 = 50$, es decir, $t_3 + t_6 = 39$, estas dos ecuaciones en t_3 y t_6 no tienen soluciones enteras positivas.
En definitiva NO existe el árbol T con las condiciones indicadas en el enunciado.

(b)



T'

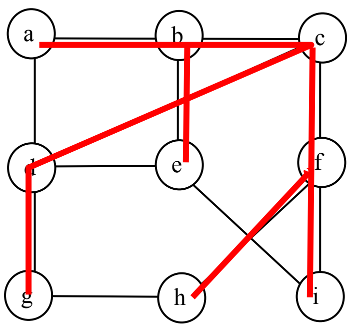

T''

Los árboles T' y T'' no son isomorfos porque el vértice de grado 4 tiene dos vecinos de grado 1 en T' y sólo uno en T''

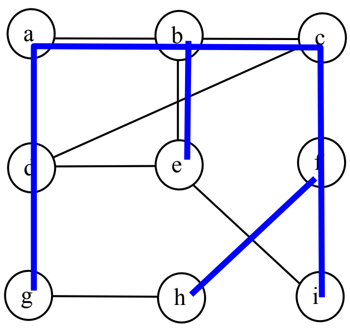
2. **(2 puntos)**
   (a) Define excentricidad de un vértice, diámetro de un grafo y camino diametral en un grafo.

En el grafo G de la figura se piden las siguientes cuestiones:
   (b) Calcula la excentricidad de cada vértice, el diámetro de G, su centro y su periferia.
   (c) Indica un camino diametral de G.
   (d) Dibuja dos árboles generadores de G, uno de diámetro 4 y otro de diámetro 6.

G



G



SOLUCIÓN
ex(c)=ex(d)=2, el resto de vértices tienen excentricidad 3
diam(G) = 3
Centro de G es el subgrafo generado por {c,d}. Periferia de G es el subgrafo generado por los restantes vértices. Un camino diametral es, por ejemplo, adgh
Los árboles en la figura. A la izquierda (rojo) de diámetro 4. A la derecha (azul) de diámetro 6

3. **(1 punto)** Define conjunto-corte en un grafo. Demuestra que la arista de menor peso de un conjunto corte en un grafo ponderado G (con pesos diferentes en las aristas) siempre pertenece al árbol generador mínimo de G.

Explicado en clase y lo podéis ver en cualquier libro de Grafos. Algunos alumnos han sustituido conjunto corte por ciclo. ¿Por qué?

4. **(1 punto)** Demostrar que las siguientes afirmaciones son equivalentes:

   (a) G es un grafo conexo en el que todas las aristas son puentes.
   (b) G es un grafo acíclico y maximal.

Había anunciado que os preguntaría alguna de las caracterizaciones de los árboles. Cuando se pide demostrar que dos afirmaciones (enunciados) (a) y (b) son equivalentes, se deben realizar DOS demostraciones: (a) $\Rightarrow$ (b) y (b) $\Rightarrow$ (a)

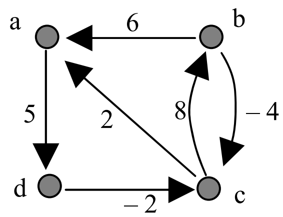
(a) $\Rightarrow$ (b) Vista en clase, en los apuntes, etc.

(b) $\Rightarrow$ (a) Partimos de que G es acíclico maximal y debemos demostrar que es conexo y todas sus aristas son puentes.

Si hubiera alguna arista no-puente formaría parte de un ciclo por lo que todas las aristas de G son puentes.

Y si G no fuera conexo, añadiendo una arista entre vértices de diferente componente conexa no se formaría ningún ciclo pero SÍ perderíamos la maximalidad.

5. **(1,5 puntos)** Describe el algoritmo de Bellman-Ford. Aplícalo al digrafo de la figura para el vértice b. (Dos iteraciones son suficientes).



Vimos en clase, y lo teníais en Moodle, la forma correcta de responder a una pregunta sobre la descripción de un algoritmo. Muchos alumnos han hecho caso omiso a todo ello y han respondido a esta pregunta casi sin palabras, presentando tan solo un par de tablas. ¿Qué calificación pueden esperar?

Repito la explicación de nuevo:
El algoritmo de Bellman-Ford calcula los caminos de peso mínimo desde un vértice a todos los demás en un digrafo con pesos en los arcos. En el digrafo se permiten pesos negativos y el algoritmo es capaz de detectar la existencia de ciclos negativos.

La breve descripción que se pide debe incluir la idea clave del algoritmo y la descripción del paso general.

Idea clave: Actualización de etiquetas en los vértices. La etiqueta de un vértice t(u) indica en cada momento el peso del camino más corto desde s encontrado hasta el momento.

La actualización se produce al considerar un arco ab. Entonces se actualiza la etiqueta del vértice final b

$$t(b) := \min \{t(b), t(a) + w(ab)\}$$

En cada iteración del algoritmo (n – 1 iteraciones, n es el orden del digrafo) se consideran todos los arcos del digrafo uno tras otro (no importa el orden) y se actualizan las etiquetas de sus extremos. La etiqueta final de un vértice u, t(u), es el peso del camino mínimo de s a u.

El digrafo de la figura tiene arcos negativos, pero NO tiene ciclos negativos. La existencia de ciclos negativos se detecta si en la n-sima iteración se produce algún cambio de etiqueta.

Peso de los caminos de peso mínimo desde el vértice b

Primera iteración

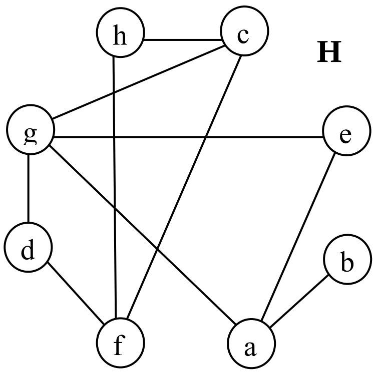
		ba	bc	cb	dc	ca	ad
b	0	0	0	0	0	0	0
a	∞	6	6	6	6	-2	-2
c	∞	∞	-4	-4	-4	-4	-4
d	∞	∞	∞	∞	∞	∞	3

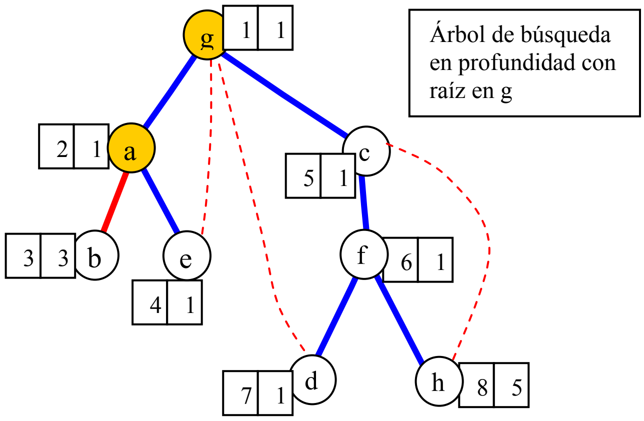
Segunda iteración

		ba	bc	cb	dc	ca	ad
c	0	0	0	0	0	0	0
a	-2	-2	-2	-2	-2	-2	-2
b	-4	-4	-4	-4	-4	-4	-4
d	3	3	3	3	3	3	3

Como no ha habido ninguna actualización de etiquetas en la segunda iteración, el algoritmo para y las etiquetas son definitivas. Indican el peso del camino mínimo al vértice respectivo.

6. **(1,5 puntos)** Aplica al grafo H de la figura el algoritmo de búsqueda en profundidad, indicando el doble etiquetado de cada vértice. Detecta los vértices-corte y las aristas puente de H a partir del doble etiquetado indicando la condición sobre las etiquetas que los caracteriza.
(La búsqueda en profundidad debe empezar en el vértice **g** y los vértices se eligen siguiendo el orden alfabético)
Construye otro grafo simple H' con la misma sucesión de grados que H y no isomorfo a H.



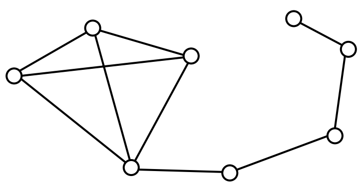


El vértice raíz g es vértice-corte porque tiene dos hijos.
La condición en las etiquetas para que un vértice (no raíz) v sea corte es que exista un hijo z de v tal que: 2ª etiq(z)≥ 1ª etiq(v) . Esto solo se cumple en el vértice a.

Una arista uv con 1ª etiq(u) < 1ª etiq(v) es puente si 2ªetiq(v) > 1ªetiq(u) . En el grafo H la única arista que cumple esta condición es ab.

Para construir un grafo H' con la misma sucesión de grados que H, es decir, con sucesión de grados 4,3,3,3,2,2,2,1 se pueden aplicar dos métodos: (1) "a ojo" mediante ensayo y error, que puede servir porque solo hay 8 vértices, o (2) aplicar al algoritmo que detecta si una sucesión es gráfica. Aplicando este segundo método se obtiene, por ejemplo, el grafo H' mostrado en la figura.

4,3,3,3,2,2,2,1
2,2,2,1,2,2,1
2,2,2,2,2,1,1
1,1,2,2,1,1
2,2,1,1,1,1
1,0,1,1,1
1,1,1,1,0
0,1,1,0



H'

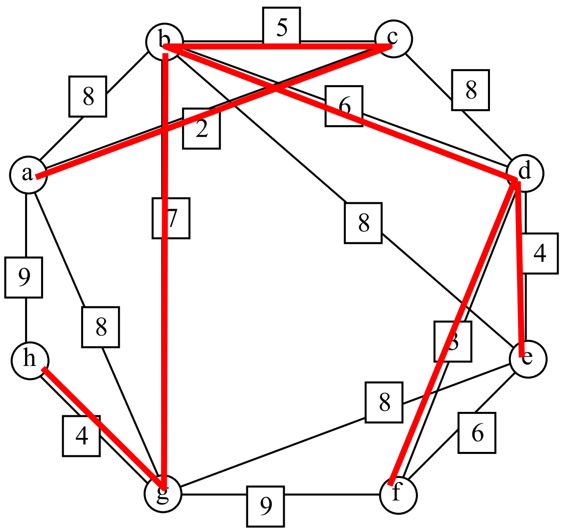
H' no es isomorfo a H porque, por ejemplo, los vértices de grado 2 en H' forman un camino y en H no

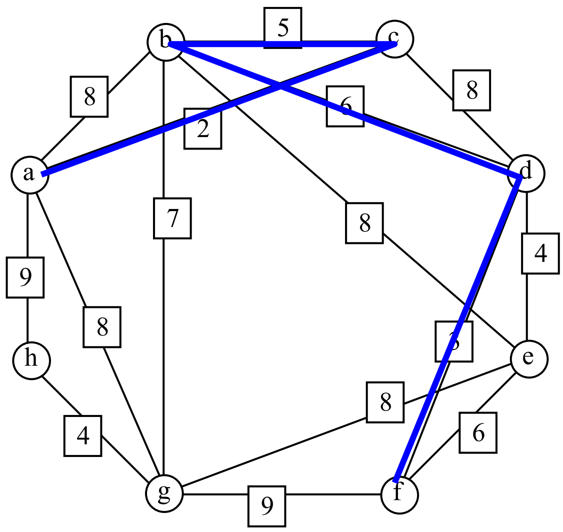
0,0,0

Como hemos comentado muchas veces en clase, hay que justificar TODAS las afirmaciones que se hagan. No basta con el dibujo de H'. Hay que decir por qué H' NO es isomorfo a H.

7. **(2 puntos)**

El grafo de la figura describe una red de pasadizos subterráneos en el juego DANGERPATHS. La etiqueta de cada arista indica el *nivel de peligrosidad* de atravesar el correspondiente pasadizo (cuanto mayor sea la etiqueta mayor será el peligro). El *nivel de peligrosidad* de un camino entre dos vértices de la red es el nivel máximo de los pasadizos que se recorran en el camino. Por ejemplo el camino bged tiene un nivel 8 de peligrosidad. Queremos construir los caminos más seguros (de menor nivel de peligrosidad) entre cada par de vértices.





Se pide:

a) Demuestra que existe cierto árbol generador de la red que contiene un camino de peligrosidad mínima entre dos vértices cualesquiera de la red.

b) Describe un algoritmo que determine, dada una red con etiquetas en las aristas indicando su nivel de peligrosidad y dos vértices *u* y *v* cualesquiera, el camino entre *u* y *v* de peligrosidad mínima.

c) Comprueba su funcionamiento en la red de la figura. ¿Cuál es el camino más seguro entre a y f?

Resolvimos en clase un ejercicio muy similar a éste. Resulta incomprensible la falta de respuestas coherentes por parte de muchos alumnos.

SOLUCIÓN

(a) Probaremos que el árbol generador de peso mínimo (MST) contiene un camino de peligrosidad mínima entre dos vértices cualesquiera de la red

Lema. Sea G un grafo conexo con pesos en las aristas que indican su peligrosidad y T el árbol generador de peso mínimo. Entonces para cada par (u,v) de vértices de G el único camino entre u y v en T, es un camino de peligrosidad mínima en G.

Dem.: Sea P ese camino en T y e la arista con su nivel de peligro. Es decir, c(P) = c(e)

Supongamos que hay otro camino P' en G de de u a v con menor peligro, c(P') < c(P). Este camino P' no contiene a la arista e (pues en caso contrario su peligro sería el de P), por lo que debe contener a una arista e' que conecte las dos componentes conexas de T – e.

Además el nivel de peligro de e' es c(e') < c(e). Por tanto, si consideramos el árbol T – e + e' tendremos un árbol de menor peso que T, contradiciendo que T sea el de peso mínimo.

(b) Algoritmo

Primer paso. Construir el árbol generador de peso mínimo T (algoritmo de Prim, Kruskal o Borůvka).

Segundo paso. Dados u y v, hallar en T el único camino entre u y v (algoritmo de búsqueda en profundidad o en anchura en el árbol T)

Se deben describir ambos algoritmos.

(c) En rojo T= MST(G) y en azul un camino de peligrosidad mínima entre a y f contenido en T. Hay otros caminos en G con el mismo nivel 6 de seguridad, por ejemplo acbdef