SOLUCIONES

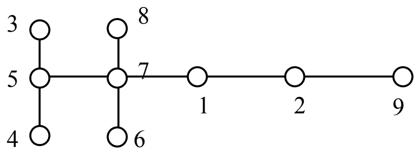
1. **(1,5 puntos)**
   (a) ¿Puede existir un grafo de orden 22, con 50 aristas y tal que todos sus vértices tengan grado 3 ó 4?
   (b) Un grafo G y su complementario son ambos 8-regulares. ¿Cuál es el orden de G?
   (c) Construye el árbol etiquetado cuyo código de Prüfer es [5, 5, 7, 7, 7, 1, 2]. ¿Puedes indicar cuál es la sucesión de grados del árbol antes de construir el árbol?

SOLUCIONES

(a) Sea x el número de vértices de grado 3 y 22 – x el de grado 4. Por la fórmula de los grados,
$3x + 4(22 – x) = 2·50 =100$, luego $– x =100 – 88 = 12$, que es imposible.
Por tanto, NO puede existir un grafo con esas condiciones.

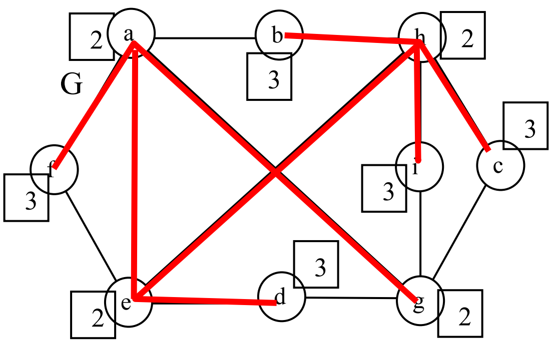
(b) En un grafo G 8-regular cada vértice v tiene 8 vecinos. Si G' es también 8-regular, entonces hay otros 8 vértices que no son vecinos de x en G. Luego el grafo tiene 17 vértices.

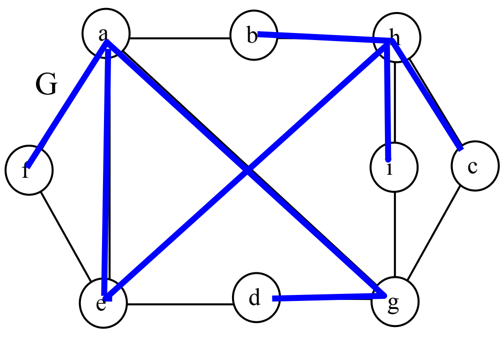
(c) Cada vértice de grado k aparece k – 1 veces en el código de Prüfer. Por tanto, la sucesión de grados del árbol será 4, 3, 2, 2, 1, 1, 1, 1, 1
Se parte del código [5, 5, 7, 7, 7, 1, 2] y la lista de vértices [1,2,3,4,5,6,7,8,9]
Se elige primer elemento del código C y primero de la lista (que no esté en C), se forma la arista correspondiente y se eliminan esos elementos. Se prosigue hasta que restan sólo dos elementos en la lista que se conectan para formar la última arista del árbol.



2. **(1,5 puntos)**
   (a) Define excentricidad de un vértice, diámetro de un grafo y camino diametral en un grafo.
   (b) Demuestra que, para n>4, $diam(P_n) + diam(P_n') = n + 1$ (P_n es el camino con n vértices y P_n' su complementario) ¿Qué sucede para n=3 y n=4?

En el grafo G de la figura se piden las siguientes cuestiones:
   (c) Calcula la excentricidad de cada vértice y el diámetro de G.
   (d) Dibuja dos árboles generadores de G, uno de diámetro 4 y otro de diámetro 5.





(b) Es inmediato comprobar, para n>4, que $diam(P_n) = n – 1$ y que $diam(P_n') = 2$ pues en el complementario $dist(u,v) \leq 2$ para dos vértices cualesquiera.
Para n<4 el complementario no es conexo y para n=4, $diam(P_4) + diam(P_4') = 6 = n + 2$

(c) Las excentricidades aparecen en el dibujo. El diámetro de G es 3. En rojo un árbol generador de diámetro 4 y en azul otro de diámetro 5

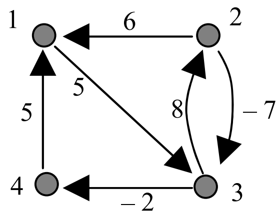
3. **(1,5 puntos)** Demuestra que la arista de mayor peso de un ciclo en un grafo ponderado G (con pesos diferentes en las aristas) nunca pertenece al árbol generador mínimo de G.

En los apuntes de clase, en el libro que seguimos, en cualquier libro de grafos, etc.

4. **(1 punto)** Sean G un grafo y v un vértice de G. Sea x otro vértice de G tal que dist(v,x) = exc(v). Demuestra que x no es un vértice-corte de G. Utiliza este resultado para demostrar que en cualquier grafo siempre hay, al menos, dos vértices que no son vértices-corte.

SOLUCIÓN
Si x fuera vértice-corte entonces en G – x habría un vértice z en distinta componente conexa que v. Todos los caminos de v a z pasarían por x, luego dist(v,x) > dist(v,z), en contradicción con que dist(v,x)=exc(v).
Por el resultado anterior dos vértices a, b que determinan el diámetro del grafo, es decir, dist(a,b) = diam(G) nunca serán vértices corte.

5. **(1 punto)** ¿Qué problema resuelve el algoritmo de Bellman-Ford? Describe el algoritmo brevemente y aplícalo al digrafo de la figura. (Al menos una iteración completa)



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

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

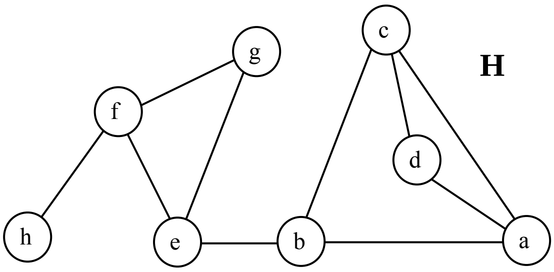
En cada iteración del algoritmo (n – 1 iteraciones, n es el orden del digrafo) se consideran todos los arcos del digrafo uno tras otro (no importa el orden) y se actualizan las etiquetas de sus extremos. La etiqueta final de un vértice u, t(u), es el peso del camino mínimo de s a u.

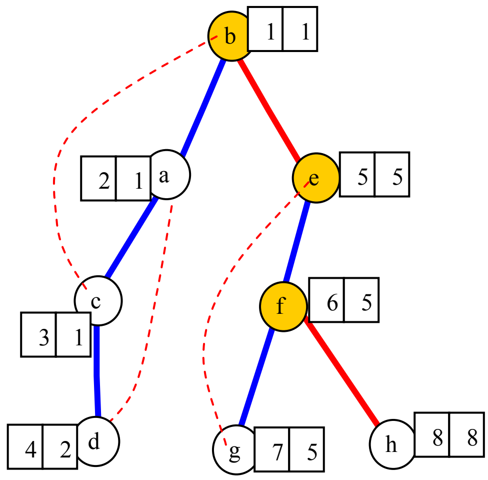
El digrafo de la figura tiene arcos negativos, pero NO tiene ciclos negativos. La existencia de ciclos negativos se detecta si en la n-sima iteración se produce algún cambio de etiqueta.

Primera iteración (para caminos de peso mínimo desde el vértice 2):

		21	32	13	41	23	34
2	0	0	0	0	0	0	0
1	∞	6	6	6	6	6	6
3	∞	∞	∞	11	11	-7	-7
4	∞	∞	∞	∞	∞	∞	-9

6. **(1 punto)** 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 **b** y los vértices se eligen siguiendo el orden alfabético)





Los vértices-corte son b, f y e.

El vértice raíz b es 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: 2^a etiq(z)$\geq$ 1^a etiq(v)

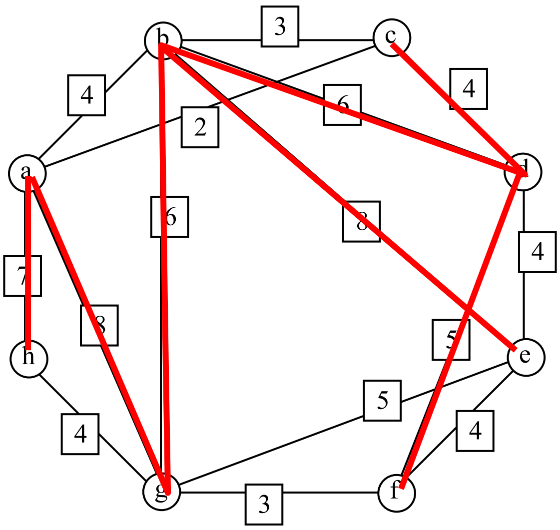
Una arista uv con 1^a etiq(u) < 1^a etiq(v) es puente si 2^aetiq(v) > 1^aetiq(u) . En el grafo H las aristas que cumplen esta condición son be y fh

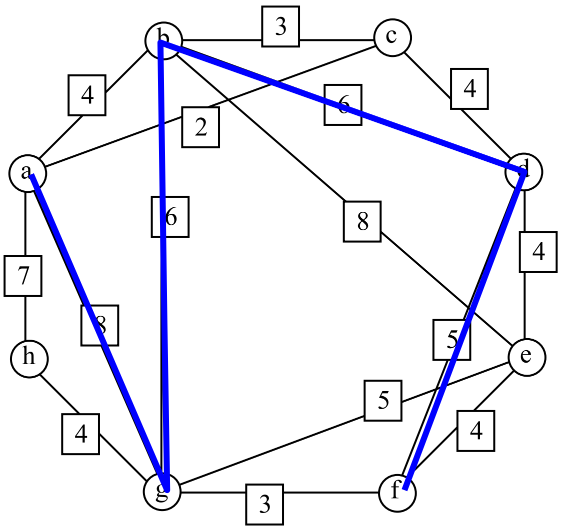
7. **(2,5 puntos)** En el grafo de la figura se representa, en forma esquemática, la red de canales de la Huerta Baja. La etiqueta de cada arista indica el caudal, en decenas de litros por segundo, en el correspondiente tramo de la red. Por ejemplo, la etiqueta 4 de la arista ab indica que por el canal correspondiente pueden fluir 40 litros por segundo entre esos dos puntos. El caudal de un camino en la red es el mínimo de los caudales de los canales que lo forman. Así el caudal del camino ahgf es de 30 litros por segundo.
Se pide:

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

b) Describe un algoritmo que determine, dada una red con caudales en las aristas y dos vértices *u* y *v* cualesquiera, el camino entre *u* y *v* de caudal máximo.

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





(a) Probaremos que el árbol generador de peso máximo (MaxST) contiene un camino de caudal máximo entre dos vértices cualesquiera de la red.

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

Dem.: Sea P ese camino en T y e la arista que determina su caudal. Es decir c(P)=c(e)

Supongamos que hay otro camino P' en G de de u a v con mayor caudal, c(P')>c(P). Este camino P' no contiene a la arista e, por lo que debe contener a una arista e' que conecte las dos componentes de T – e.

Además el caudal de e' es c(e')>c(e). Por tanto, si consideramos el árbol T – e + e' tendremos un árbol de mayor peso que T, contradiciendo que T sea el de peso máximo.

(b)

Algoritmo

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

Segundo paso. Dados u y v, hallar en T el único camino entre u y v. (Por DFS o BFS con raíz en u)

(c) En rojo MaxST y en azul el camino de caudal máximo entre a y f contenido en MaxST, de 50 litros por minuto de caudal.