Presentación

Esta PEC profundiza en los conceptos básicos de la teoría de grafos que cubren los contenidos estudiados en los módulos 4 y 5 de la asignatura. Los ejercicios trabajan tanto los conceptos previos sobre grafos, como una de las clases más importantes de grafos, los árboles, así como dos de los problemas más notables de recorridos de grafos, los grafos eulerianos y los grafos hamiltonianos.

Competencias

En esta PEC se trabajan las siguientes competencias del Grado de Ingeniería Informática:

- Capacidad para utilizar los fundamentos matemáticos, estadísticos y físicos para comprender los sistemas TIC.
- Capacidad para analizar un problema en el nivel de abstracción adecuado en cada situación y aplicar las habilidades y conocimientos adquiridos para resolverlo.

Objetivos

Los objetivos concretos de esta PEC son:

- Saber caracterizar los árboles y, específicamente, los árboles con raíz.
- Saber aplicar los algoritmos de determinación de un ábol generador minimal.
- Identificar los grafos eulerianos y hamiltonianos y caracterizarlos.
- Entender el problema del viajante de comercio (TSP). Conocer y saber aplicar el algoritmo de resolución aproximada de este problema.

Descripción de la PEC a realizar

1. (Valoración de un 20 % = 5 %+5 %+3 %+7 %)

   a) Desde el Departamento de Recursos Humanos de la empresa donde trabajamos nos plantean el siguiente problema. En breve se llevará a cabo un proceso de selección para cubrir varios puestos de trabajo, y hay que diseñar un cuestionario que permita clasificar a los candidatos de acuerdo con un cierto número a de perfiles psicológicos distintos. Se quiere que, para cada perfil psicológico, haya un único conjunto de respuestas al cuestionario que le corresponda (por ejemplo, si el cuestionario consiste en tres preguntas, uno de estos conjuntos podría ser "la segunda respuesta de la primera pregunta, la tercera respuesta de la segunda pregunta, y la primera respuesta de la tercera pregunta"). Por simplicidad, se quiere que cada pregunta tenga exactamente el mismo número b de posibles respuestas. Esto es, el cuestionario consistiría en un número adecuado de preguntas con b posibles respuestas cada una, de manera que el conjunto de las respuestas de un candidato determina exactamente uno de los perfiles sicológicos. Si $a = 100$ y $b = 3$, ¿es posible diseñar tal cuestionario?

   Solución: Podemos ver el cuestionario como un árbol completo b-ario en que cada vértice interno es una pregunta y cada hoja es un perfil psicológico. Sea x el número de vértices internos. Sabemos que se debe cumplir que $a = (b-1)x + 1$. De modo que $x = \frac{a-1}{b-1}$. Como $a - 1 = 99$ no es divisible entre $b - 1 = 2$, no se puede diseñar el cuestionario.

   ---

   b) ¿Qué orden debe tener un bosque con 2 vértices de grado 4, 3 vértices de grado 3, 4 vértices de grado 2, 17 hojas, y formado por 5 árboles?

   Solución: Calculemos la suma de grados de los vértices: $2 \cdot 4 + 3 \cdot 3 + 4 \cdot 2 + 17 \cdot 1 = 42$ (por definición, las hojas tienen grado 1). Por la fórmula de los grados, tenemos que la medida del bosque es $\frac{42}{2} = 21$.

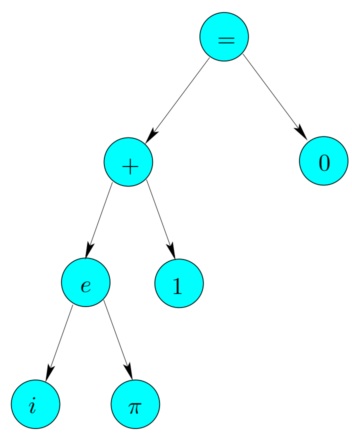
   Por otro lado, sabemos que todo bosque de orden n, medida m y k componentes conexas cumple que $m = n - k$, de donde $n = m + k$. Por lo tanto el orden es $21 + 5 = 26$.

c) Una de las ecuaciones consideradas como más bellas en matemáticas es la llamada *identidad de Euler*, según la cual:

$$e^{i\pi} + 1 = 0.$$

Dibujad el árbol con raíz de la ecuación, y escribid el recorrido en preorden, inorden y postorden de ese árbol.

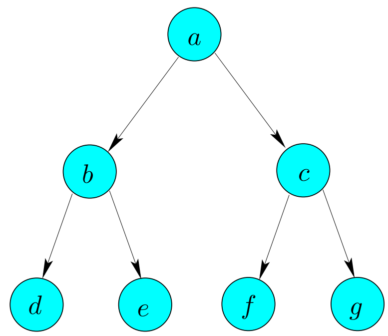
Solución: El árbol es:



Los recorridos son:

- En preorden: $=, +, e, i, \pi, 1, 0$
- En inorden: $i, e, \pi, +, 1, =, 0$
- En postorden: $i, \pi, e, 1, +, 0, =$

d) Se dice que un árbol m-ario es *perfecto* si es completo y todas las hojas están en el mismo nivel. Una manera de implementar un árbol de este tipo es mediante una tabla: en primer lugar está la raíz (el único vértice de nivel 0), luego los vértices de nivel 1, luego los vértices de nivel 2, etc. Por ejemplo, el siguiente árbol binario perfecto



se implementaría con la siguiente tabla (cuyo primer índice es 0):

0	1	2	3	4	5	6
a	b	c	d	e	f	g

En lo que sigue vamos a asumir que, como en el ejemplo, la tabla está indexada a partir de 0. Sea i un número entre 0 y $m-1$. Si un vértice interno ocupa la posición p de la tabla, ¿en qué posición se encuentra su i-ésimo hijo, en función de i, p y m?

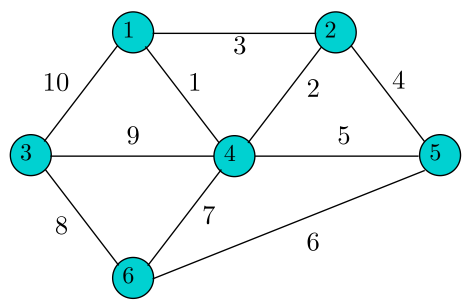
Solución: En primer lugar observamos que en el nivel 0 hay 1 vértice, en el nivel 1 hay m vértices, en el nivel 2 hay m^2 vértices, etc. Llamemos x el vértice interior en la posición p, y llamemos y su i-ésimo hijo. Sea k el nivel de x, y supongamos que x es el j-ésimo vértice de su nivel, donde $0 \leq j < m^k$. Entonces en la tabla antes de x hay j vértices del nivel k, más los $1+m+...+m^{k-1} = \sum_{r=0}^{k-1} m^r$ vértices de los niveles anteriores. Por lo tanto $p = j + \sum_{r=0}^{k-1} m^r$. Por otra parte, el vértice y tiene i hijos de x que van antes en la tabla. Además, cada uno de los j vértices anteriores a x del nivel k tiene m hijos, que van antes que y en el nivel $k+1$. Finalmente, hay que añadir los $1+m+...+m^k = \sum_{r=0}^{k} m^r$ vértices de los niveles anteriores al $k+1$. De modo que la posición de y en la tabla es $i+jm+\sum_{r=0}^{k} m^r = i+jm+1+m\sum_{r=0}^{k-1} m^r = i+1+m(j+\sum_{r=0}^{k-1} m^r) = i+1+mp$.

---

2. (Valoración de un 20 % = 8 %+12 %)

La comunidad de regantes de la comarca ha decidido tomar medidas ante la situación de sequía actual. Para evitar la pérdida de agua debida a la evaporación, se quiere reducir el número de canales por los que se hace circular el agua. Más concretamente, de los canales existentes, se tiene que decidir por cuáles va a circular agua y por cuáles no. Para ello se dispone de un grafo ponderado, donde los vértices son campos, las aristas canales, y el peso de cada arista es el volumen de agua que se evapora en un día en ese canal (medido en m^3). Evidentemente, se quiere que todos los campos queden irrigados.

a) Supongamos que la red de canales es:



¿Por qué canales hay que hacer circular el agua de modo que se minimice la evaporación? ¿Qué volumen de agua se va a evaporar en un día en el mejor de los casos? Utilizad el algoritmo de Prim para responder.

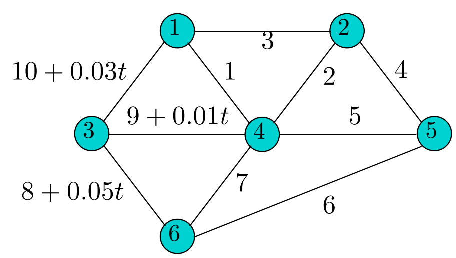
Solución: Elegimos como vértice inicial el vértice 1. En este caso la tabla del algoritmo de Prim es la siguiente:

1	2	3	4	5	6
$(0,1)$	$(\infty,1)$	$(\infty,1)$	$(\infty,1)$	$(\infty,1)$	$(\infty,1)$
$(0,1)*$	$(3,1)$	$(10,1)$	$(1,1)$	$(\infty,1)$	$(\infty,1)$
$(0,1)$	$(2,4)$	$(9,4)$	$(1,1)*$	$(5,4)$	$(7,4)$
$(0,1)$	$(2,4)*$	$(9,4)$	$(1,1)$	$(4,2)$	$(7,4)$
$(0,1)$	$(2,4)$	$(9,4)$	$(1,1)$	$(4,2)*$	$(6,5)$
$(0,1)$	$(2,4)$	$(8,6)$	$(1,1)$	$(4,2)$	$(6,5)*$
$(0,1)$	$(2,4)$	$(8,6)*$	$(1,1)$	$(4,2)$	$(6,5)$

El árbol generador minimal consiste en las aristas $\{1,4\}$, $\{2,4\}$, $\{2,5\}$, $\{3,6\}$, $\{5,6\}$. Estos son los canales por los que circulará el agua. El volumen de agua que se va a evaporar en un día en el mejor de los casos, o sea el coste mínimo, es de 21 m^3.

---

b) Previendo que el problema de la escasez de agua se agrave en las próximas semanas, la comunidad de regantes decide llevar a cabo un estudio para mejorar todavía más el sistema de riego. Como resultado de este estudio, se observa que la exposición al sol de algunos canales varía con el tiempo, de modo que la evaporación de agua no se puede considerar constante. Usando datos históricos, han estimado el volumen de agua (en m^3) que se evapora el día t en cada canal de la red, donde $t \in \{0,1,2,\ldots,100\}$. En el siguiente grafo se muestran estas estimaciones:



Para cada día $t \in \{0,1,2,\ldots,100\}$, se quiere saber por qué canales hay que hacer circular el agua para minimizar la evaporación. Utilizad el algoritmo de Kruskal para responder.

Pista: dividid el tiempo en intervalos de modo que en cada intervalo la ordenación por pesos de las aristas sea la misma.

Solución: Después de analizar el grafo, vemos que sólo cambian en función del tiempo los pesos de las 3 aristas de mayor peso. Como el peso aumenta con t, sólo cambiará la ordenación relativa de estas 3 aristas.

Observamos que $8 + 0{,}05t \leq 10 + 0{,}03t$ si y sólo si $0{,}02t \leq 2$, o sea, $t \leq 100$. Y que $9 + 0{,}01t \leq 10 + 0{,}03t$, esto es, $0 \leq 1 + 0{,}02t$, para todo $t \geq 0$. Por lo tanto, la arista $\{1,3\}$ siempre va a tener el mismo peso o mayor que las aristas $\{3,4\}$ y $\{3,6\}$.

Así pues, sólo hay que determinar cómo se comparan los pesos de las aristas $\{3,4\}$ y $\{3,6\}$. Tenemos que $8+0{,}05t \leq 9+0{,}01t$ si y sólo si $0{,}04t \leq 1$, esto es, $t \leq 25$. De modo que si $t \leq 25$ la arista $\{3,6\}$ tiene un peso menor o igual al de la arista $\{3,4\}$, y viceversa si $t \geq 25$.

Distinguimos pues dos casos. Si $t \leq 25$, la lista ordenada de las aristas de menos peso a más peso es:

Aristas	**Pesos**
$\{1,4\}$	1
$\{4,2\}$	2
$\{1,2\}$	3
$\{2,5\}$	4
$\{4,5\}$	5
$\{6,5\}$	6
$\{6,4\}$	7
$\{3,6\}$	$8+0{,}05t$
$\{3,4\}$	$9+0{,}01t$
$\{3,1\}$	$10+0{,}03t$

Tenemos que elegir las 5 primeras aristas (porque hay 6 vértices) que no formen ningún ciclo. Las marcaremos con un asterisco; las aristas descartadas porque forman ciclo las marcaremos en negrita. Resulta la siguiente tabla:

Aristas	**Pesos**
$\{1,4\}*$	1
$\{4,2\}*$	2
{ 1, 2 }	
$\{2,5\}*$	4
{ 4, 5 }	
$\{6,5\}*$	6
{ 6, 4 }	
$\{3,6\}*$	$8+0{,}05t$

Por lo tanto, un árbol generador minimal está formado por las aristas: $\{1,4\}$, $\{4,2\}$, $\{2,5\}$, $\{6,5\}$, $\{3,6\}$, con un peso total de $21+0{,}05t\ m^3$.

Si $t \geq 25$, la lista ordenada de las aristas de menos peso a más peso es:

Aristas	**Pesos**
$\{1,4\}$	1
$\{4,2\}$	2
$\{1,2\}$	3
$\{2,5\}$	4
$\{4,5\}$	5
$\{6,5\}$	6
$\{6,4\}$	7
$\{3,4\}$	$9+0{,}01t$
$\{3,6\}$	$8+0{,}05t$
$\{3,1\}$	$10+0{,}03t$

Tenemos que elegir las 5 primeras aristas que no formen ningún ciclo. Siguiendo la misma notación que antes, resulta la siguiente tabla:

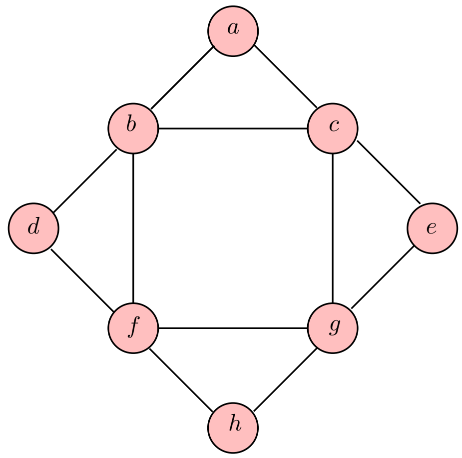
Aristas	**Pesos**
$\{1,4\}*$	1
$\{4,2\}*$	2
{ 1, 2 }	
$\{2,5\}*$	4
{ 4, 5 }	
$\{6,5\}*$	6
{ 6, 4 }	
$\{3,4\}*$	$9+0{,}01t$

Por lo tanto, un árbol generador minimal está formado por las aristas: $\{1,4\}$, $\{4,2\}$, $\{2,5\}$, $\{6,5\}$, $\{3,4\}$, con un peso total de $22+0{,}01t\ m^3$.

---

3. (Valoración de un 20 % = 6 %+8 %+6 %)

   a) Sea $G=(V,A)$ un grafo conexo tal que todos sus vértices tienen grado par. Dad un algoritmo para orientar las aristas de A de modo que, para cada vértice $v\in V$, el grado de entrada de v sea el mismo que el grado de salida, y justificad por qué es correcto. Aplicad vuestro algoritmo al siguiente grafo:



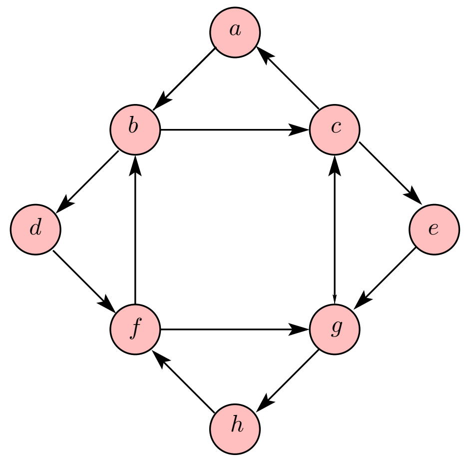
Solución: Un grafo conexo que tiene todos sus vértices de grado par es euleriano. Equivalentemente, el conjunto de las aristas A se puede expresar como $A = A_1 \cup \ldots \cup A_k$, donde el conjunto de las aristas de cada A_i forma un circuito, y se cumple que $A_i \cap A_j = \emptyset$ si $i \neq j$. Como son disjuntos, podemos orientar las aristas de cada A_i de modo que se obtenga un circuito dirigido. De esta manera tenemos que, en cada vértice por donde pasa, el circuito tiene tantos arcos de entrada como de salida. Así, si tomamos un vértice $v \in V$ arbitrario y contamos arcos incidentes en v sobre todos los A_i, tendremos tantos arcos de entrada como de salida. Pero eso significa precisamente que v tiene el mismo grado de entrada que de salida.

Así pues, basta con tomar una partición del conjunto de aristas en circuitos disjuntos y orientar las aristas de cada circuito para que forme un circuito dirigido. Para calcular esta partición podemos, por ejemplo, usar el algoritmo de Hierholzer, que en cada iteración encuentra un nuevo ciclo disjunto.

Para el grafo del enunciado, el algoritmo de Hierholzer daría lugar a la siguiente tabla, donde ya escribimos las aristas con la orientación adecuada:

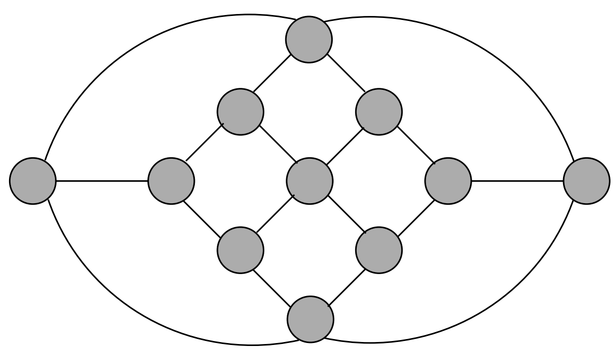
Iteración	v	C'	C
0	a		
1	a	a, b, c, a	a, b, c, a
2	b	b, d, f, b	a, b, d, f, b, c, a
3	f	f, g, h, f	$a, b, d, f, g, h, f, b, c, a$
4	g	g, c, e, g	$a, b, d, f, g, c, e, g, h, f, b, c, a$

El grafo orientado sería entonces:



---

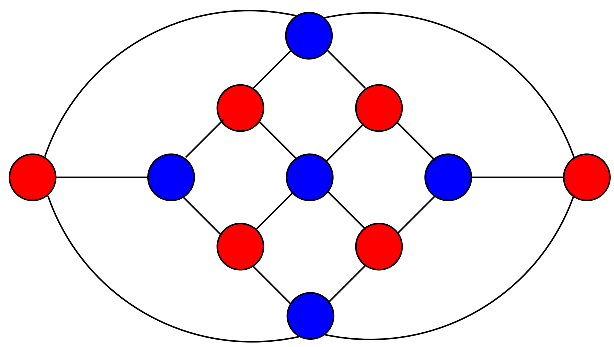
b) Describid un algoritmo basado en la búsqueda en profundidad para determinar si un grafo dado es bipartito. Aplicadlo al siguiente grafo. ¿Es hamiltoniano?



Solución:

Un grafo es bipartito si se pueden pintar los vértices con dos colores (por ejemplo, rojo y azul) de modo que no haya vértices adyacentes pintados con el mismo color. Usando esta observación, podemos determinar si un grafo es bipartito del modo siguiente. Primero, escogemos un vértice arbitrario y lo pintamos de color rojo. Entonces empezamos una búsqueda en profundidad a partir de ese vértice. Cada vez que desempilamos un vértice v, recorremos cada uno de sus vértices w adyacentes. Si w no está pintado, lo pintamos con el color opuesto al de v (esto es, azul si v está pintado de color rojo, o rojo si v está pintado de color azul), y lo apilamos. Si w ya está pintado con el color opuesto al de v, ya ha sido visitado previamente y no hace falta volverlo a apilar. Y si w ya está pintado pero con el mismo color al de v, hemos encontrado un ciclo de longitud impar y por lo tanto el grafo no es bipartito. Si se consigue explorar todo el grafo sin llegar a ningún conflicto, el coloreado define la partición que demuestra que el grafo es bipartito.

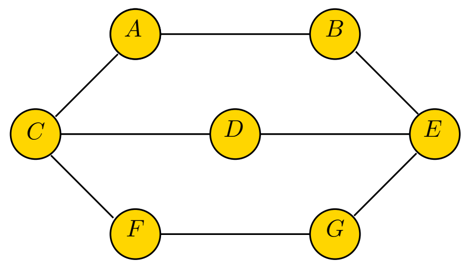
Aplicando este algoritmo al grafo del enunciado se obtiene que en efecto es bipartito:



Como el número de vértices pintados de color rojo (6) es distinto del número de vértices pintados de color azul (5), el grafo no puede ser hamiltoniano.

---

c) ¿Es hamiltoniano el siguiente grafo? En caso afirmativo, mostrad un ciclo hamiltoniano. En caso negativo, justificad por qué no puede existir tal ciclo.



Solución: Observamos que si eliminamos los vértices C y E y sus aristas nos quedan 3 componentes conexas: $\{D\}$, $\{A, B\}$ y $\{F, G\}$. Como sólo hemos eliminado 2 vértices, el grafo no puede ser hamiltoniano.

4. (Valoración de un 20 % = 3 %+4 %+2 %+7 %+4 %)

Una conocida marca automovilística nos pide ayuda para mejorar el proceso de pintado de sus coches. Los coches se pintan por tandas, cada uno con un color, caracterizado por una tripleta de números (r, g, b). Siempre que llega un coche que hay que pintar de color diferente al coche pintado justo antes, hay que limpiar las pistolas. Esta tarea de limpieza para pasar del color (r_1, g_1, b_1) al color (r_2, g_2, b_2) tiene un coste de transición

$$t((r_1, g_1, b_1), (r_2, g_2, b_2)) = \max(|r_1 - r_2|, |g_1 - g_2|, |b_1 - b_2|).$$

La fórmula del coste de transición es válida en particular en el caso que uno de los colores sea $(16, 16, 16)$. Este es un color ficticio que se usa para representar cuando no hay pintura en las pistolas. Inicialmente las pistolas están limpias, o sea, "cargadas" del color $(16, 16, 16)$. Y cuando finalmente se termina la tanda de coches, hay que volver a limpiar las pistolas, para que estén listas para cuando llegue la siguiente tanda.

a) Dados n colores distintos, considerad el grafo completo de orden n en el que cada vértice es un color, y la arista del color i al color j tiene peso el coste de transición de i a j. Demostrad que este grafo ponderado satisface la desigualdad triangular.

Solución:

En efecto, dados tres colores (r_1, g_1, b_1), (r_2, g_2, b_2) y (r_3, g_3, b_3), se cumple que:

$$
\begin{aligned}
& t((r_1, g_1, b_1), (r_2, g_2, b_2)) \\
= \; & \text{máx}(|r_1 - r_2|, |g_1 - g_2|, |b_1 - b_2|) \\
\leq \; & \text{máx}(|r_1 - r_3| + |r_3 - r_2|, |g_1 - g_3| + |g_3 - g_2|, |b_1 - b_3| + |b_3 - b_2|) \\
\leq \; & \text{máx}(|r_1 - r_3|, |g_1 - g_3|, |b_1 - b_3|) + \text{máx}(|r_3 - r_2|, |g_3 - g_2|, |b_3 - b_2|) \\
= \; & t((r_1, g_1, b_1), (r_3, g_3, b_3)) + t((r_3, g_3, b_3), (r_2, g_2, b_2))
\end{aligned}
$$

b) Para cada coche c denotamos por color(c) el color con el que se tiene que pintar. Supongamos que tenemos una tanda, o sea una secuencia de coches $C = (C_1, C_2, ...)$ con dos posiciones i y j tales que $i + 1 < j$ y color(C_i) = color(C_j). Demostrad que la secuencia en la que quitamos C_j de su posición y lo colocamos a continuación de C_i tiene un coste no superior al de C.

Solución: Veamos cómo el cambio afecta al coste de la secuencia. Por un lado, al quitar C_j de su posición por la desigualdad triangular tenemos que

$$t(k_{j-1}, k_j) + t(k_j, k_{j+1}) \geq t(k_{j-1}, k_{j+1}),$$

donde k_{j-1} = color(C_{j-1}), k_j = color(C_j) y k_{j+1} = color(C_{j+1}), de modo que el coste no puede empeorar. Por otro, al colocar C_j a continuación de C_i, como tienen el mismo color no se incrementa el coste. Teniendo en cuenta ambas observaciones, podemos concluir que la nueva secuencia tiene un coste no superior al de la original.

c) De ahora en adelante supongamos que los colores disponibles son: $(0, 0, 0)$, $(1, 1, 1)$, $(2, 2, 7)$ y $(3, 3, 3)$ (a los que hay que añadir el color ficticio $(16, 16, 16)$). Rellenad la siguiente tabla con los costes de transición de un color a otro.

	$(16, 16, 16)$	$(0, 0, 0)$	$(1, 1, 1)$	$(2, 2, 7)$	$(3, 3, 3)$
$(16, 16, 16)$					
$(0, 0, 0)$					
$(1, 1, 1)$					
$(2, 2, 7)$					
$(3, 3, 3)$					

Solución: La tabla es la siguiente:

	$(16,16,16)$	$(0,0,0)$	$(1,1,1)$	$(2,2,7)$	$(3,3,3)$
$(16,16,16)$	0	16	15	14	13
$(0,0,0)$	16	0	1	7	3
$(1,1,1)$	15	1	0	6	2
$(2,2,7)$	14	7	6	0	4
$(3,3,3)$	13	3	2	4	0

d) Supongamos que llega la siguiente tanda de 15 coches, cada uno con el correspondiente color con el que se tiene que pintar:

(1) Coche A, color $(0,0,0)$;

(2) Coche B, color $(2,2,7)$;

(3) Coche C, color $(3,3,3)$;

(4) Coche D, color $(0,0,0)$;

(5) Coche E, color $(1,1,1)$;

(6) Coche F, color $(0,0,0)$;

(7) Coche G, color $(1,1,1)$;

(8) Coche H, color $(2,2,7)$;

(9) Coche I, color $(0,0,0)$;

(10) Coche J, color $(1,1,1)$;

(11) Coche K, color $(2,2,7)$;

(12) Coche L, color $(0,0,0)$;

(13) Coche M, color $(1,1,1)$;

(14) Coche N, color $(2,2,7)$;

(15) Coche O, color $(3,3,3)$.

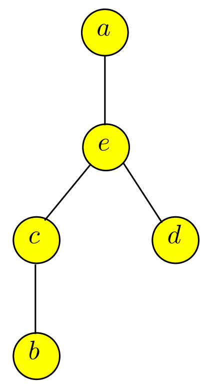
Encontrad una ordenación de los coches cuyo coste total de limpieza sea como mucho el doble del coste total de la solución óptima. ¿Qué coste tiene la ordenación de vuestra respuesta? Indicad qué algoritmo(s) usáis para calcularla.

Solución: En primer lugar observamos que, en base a lo visto en el apartado $4b$, podemos colocar consecutivamente todos los coches que hay que pintar del mismo color. Esto reduce el problema a encontrar una secuencia de los colores de coste mínimo donde aparezca cada color exactamente una vez, excepto el $(16, 16, 16)$, que es el inicio y final de la secuencia. Se trata pues de un problema de TSP. Como los pesos cumplen la desigualdad triangular, concretamente es un problema de TSP con desigualdad triangular. Estamos por lo tanto en condiciones de aplicar el algoritmo TSP-aproximado, del que tenemos la garantía que dará una solución cuyo coste es como mucho el doble del coste de la solución óptima.

Para simplificar la notación, vamos a representar los colores con letras, de modo que $a = (16, 16, 16)$, $b = (0, 0, 0)$, $c = (1, 1, 1)$, $d = (2, 2, 7)$ y $e = (3, 3, 3)$. Escogemos un color, por ejemplo el a, y calculamos un árbol generador a partir de este color usando el algoritmo de Prim. Obtenemos la siguiente tabla:

a	b	c	d	e
$(0, a)$	(∞, a)	(∞, a)	(∞, a)	(∞, a)
$(0, a)*$	$(16, a)$	$(15, a)$	$(14, a)$	$(13, a)$
$(0, a)$	$(3, e)$	$(2, e)$	$(4, e)$	$(13, a)*$
$(0, a)$	$(1, c)$	$(2, e)*$	$(4, e)$	$(13, a)$
$(0, a)$	$(1, c)*$	$(2, e)$	$(4, e)$	$(13, a)$
$(0, a)$	$(1, c)$	$(2, e)$	$(4, e)*$	$(13, a)$

A partir de la tabla vemos que un árbol generador minimal consiste en las aristas $\{a, e\}$, $\{b, c\}$, $\{c, e\}$, $\{d, e\}$:



Un recorrido en preorden de este árbol es por ejemplo a, e, c, b, d. Añadimos finalmente la arista $\{d, a\}$, y así obtenemos el ciclo a, e, c, b, d, a, con coste $13 + 2 + 1 + 7 + 14 = 37$. Sabemos que éste es como mucho el doble del coste de la solución óptima.

Para dar con una ordenación de los coches con coste 37, simplemente reemplazamos cada color por los coches que se tienen que pintar de ese color, dispuestos en cualquier orden, por ejemplo:

C,	O,				(color e)
E,	G,	J,	M,		(color c)
A,	D,	F,	I,	L,	(color b)
B,	H,	K,	N.		(color d)

---

e) ¿Por qué motivo puede ser más conveniente usar el algoritmo de Prim que el de Kruskal como subprocedimiento en el algoritmo TSP-aproximado?

Solución: El algoritmo de Prim para un grafo de orden n y medida m tiene coste $O(n^2)$, mientras que el algoritmo de Kruskal tiene coste $O(m \log m)$. Pero todo grafo que satisface la desigualdad triangular tiene que ser completo, de modo que $m = O(n^2)$. Así pues, el coste del algoritmo de Kruskal dentro del algoritmo TSP-aproximado es $O(n^2 \log n)$, que es peor que el coste del algoritmo de Prim.

---

5. (Valoración de un 20 %) Cuestionario de evaluación Moodle

Dentro del aula de la asignatura, en el Campus Virtual, encontraréis una nueva herramienta (Moodle) en la parte derecha. En este Moodle hay un cuestionario con diversas preguntas que debéis resolver como último ejercicio de esta PEC.

Leed atentamente las siguientes instrucciones **antes de abrir el cuestionario**:

- Los contenidos que se evalúan en este cuestionario corresponden a los módulos 4 (*Árboles*) y 5 (*Grafos eulerianos y grafos hamiltonianos*). Es importante que hayáis asimilado estos conocimientos **antes de abrir el cuestionario**.
- Para visualizar correctamente el cuestionario utilizad el Chrome o el Firefox.
- El cuestionario estará abierto durante el plazo de la PEC y lo podéis resolver cuando queráis. De todas formas, una vez lo abráis tendréis un **tiempo limitado** para resolverlo (1 hora y 15 minutos).

  Importante: El cuestionario quedará cerrado a las 23:59 de la fecha límite de entrega. Si empezáis a hacerlo después de las 22:59 del último día, ¡tendréis menos de una hora y cuarto para hacerlo!
- Las respuestas a las preguntas se tienen que introducir directamente en el cuestionario Moodle. No es necesario que las entreguéis junto con el resto de respuestas de la PEC.
- Las preguntas del cuestionario son aleatorias: cada estudiante recibirá un enunciado diferente.
- Disponéis de **2 intentos** para resolver el cuestionario. El objetivo de tener dos intentos es poder solventar posibles problemas que hayáis tenido en la realización del cuestionario, ya sean problemas técnicos o bien que hayáis abierto el cuestionario por error. Por tanto, debéis tener en cuenta que:
  - La nota que obtendréis en el cuestionario será la de vuestro último intento.
  - Después del 1r intento, no recibiréis la calificación obtenida ni recibiréis feedback sobre vuestra propuesta de solución. Por lo tanto, no recomendamos usar el 2o intento para intentar mejorar nota, ya que puede ser que obtengáis una nota inferior.
  - Si usáis el 2o intento, el enunciado que encontraréis será diferente del del 1r intento.
  - Podéis realizar los dos intentos en días diferentes, siempre que sea dentro del plazo de la PEC. Dispondréis de 1 hora y 15 minutos para cada intento.
  - Cada vez que iniciéis el cuestionario contará como un intento, aunque no enviéis la respuesta. Por ejemplo, **si habéis hecho el 1r intento y volvéis a abrir el**

cuestionario, invalidaréis vuestro 1r intento y os quedaréis con la nota del 2o intento.

Recursos

Recursos Básicos

- Módulo didáctico 4. árboles.
- Módulo didáctico 5. Grafos eulerianos y grafos hamiltonianos.
- Colección de problemas.

Recursos Complementarios

- PECs y exámenes de semestres anteriores.
- Programario para el estudio de algoritmos sobre grafos.
- Enlaces: Applets interactivos sobre algoritmos de grafos.

Criterios de valoración

- La PEC se tiene que resolver **de forma individual**. En caso de que hayáis consultado recursos externos, es necesario referenciarlos.
- Es necesario justificar la respuesta de cada apartado. Se valorará tanto el resultado final como la justificación dada.
- En los apartados donde sea necesario aplicar algún algoritmo, se valorará la elección del algoritmo apropiado, los pasos intermedios, el resultado final y las conclusiones que se deriven.

Formato y fecha de entrega

Hay que entregar **único documento** PDF con las respuestas de todos los ejercicios. El nombre del fichero tiene que ser: **PEC2_Apellido1Apellido2Nombre.pdf**.

Este documento se tiene que entregar en el espacio **Entrega y Registro de EC** del aula **antes de las 23:59 del día 04/05/2023**. **No se aceptarán entregas fuera de plazo.**