Ejercicios de AED3, versión 1.732
439 enunciados - 419 soluciones

Alejandro Strejilevich de Loma

June 15, 2022

Introducción

Prescindible

Fui mucho tiempo JTP de Algo3, y gran parte de ese tiempo estuve en la Práctica. Durante ese lapso tomamos muchos exámenes parciales y sus recuperatorios. En algún momento empecé a recopilar los enunciados de los ejercicios que se tomaban, y luego de cierto tiempo empecé a registrar las soluciones de esos ejercicios. El resultado de ese trabajo aparece en este documento.

Ya no soy docente de Algo3, y no quisiera que este material se desperdiciara. Estuve dudando entre cederlo a mis ex compañeros docentes para que armaran más exámenes, o por el contrario ponerlo a disposición de los alumnos (y docentes) para que lo usaran a su criterio. Finalmente me decidí por esto último, ya que pensé que mi obligación como (ex) docente es principalmente hacia los alumnos.

Espero que los docentes de la materia no tomen a mal la publicación de ejercicios de examen. En mi opinión, si los alumnos aprenden a resolver cientos de ejercicios, de tal modo que pueden aprobar un examen conteniendo algunos de esos ejercicios, bien puede considerarse que saben el tema.

Este material era usado por mí para llevar adelante mi trabajo como JTP, de modo que es posible que la presentación tenga errores y no sea en general óptima. Si tengo tiempo y ganas voy a mejorar el documento y en tal caso voy a poner a disposición las nuevas versiones.

Imprescindible

A lo largo de este documento se usa $\mathbb{N}$ para denotar $\mathbb{Z}_{>0}$ (enteros positivos), mientras que se usa $\mathbb{N}_0$ para indicar $\mathbb{Z}_{\geq 0}$ (enteros no negativos). Asimismo, se supone que el grafo trivial no es bipartito; si bien esto es opinable, está en concordancia con las guías de ejercicios que se utilizaban en la materia.

Los ejercicios aparecen agrupados por tema; si un ejercicio pertenece a más de un tema, puede aparecer en cualquiera de ellos, o en una sección especial indicando tal situación. En ocasiones se mencionan ejercicios de las guías que se utilizaban en la materia; tales ejercicios pueden consultarse en el Apéndice A.

Algunas soluciones provistas son muy detalladas, mientras que otras sólo brindan las ideas principales, o alguna versión intermedia entre ambos extremos. Por ejemplo, en los ejercicios de clases de complejidad, es usual que sólo se indique la transformación polinomial, lo cual normalmente no es la solución completa. El nivel de detalle de la solución provista, no necesariamente coincide con el de la solución ideal.

En muchos de los ejercicios se indica su origen, el cual puede ser una página de Internet, un libro, la propuesta de algún docente, entre otros. Sea cual sea el origen, el enunciado final del ejercicio puede ser desde una transcripción más o menos directa del enunciado original, hasta una versión muy modificada de tal enunciado original.

Recomiendo fuertemente a los alumnos *no* basar su estudio en ejercicios resueltos. Lo ideal es estudiar el material teórico y de otro tipo provisto por los docentes, hacer las guías de ejercicios propuestos, y recién en ese momento considerar otras fuentes.

Este material se provee *tal cual es.* Puede contener errores de cualquier tipo, y no me hago responsable de ninguna consecuencia que ocasione su uso. Si usted no está de acuerdo con esto, debería abstenerse de usar este material.

Complejidad asintótica

1. Sean las funciones $f_1, f_2, g_1, g_2 : \mathbb{N} \to \mathbb{R}_{\geq 0}$. Demostrar que si $f_1(n) = O(g_1(n))$ y $f_2(n) = O(g_2(n))$ entonces

Variante 1: $f_1(n) + f_2(n) = O(g_1(n) + g_2(n))$.

Variante 2: $f_1(n) \times f_2(n) = O(g_1(n) \times g_2(n))$.

SOLUCIÓN: Por definición.

TOMADO: 2001C2R1 19-DIC-2001 (Variante 1).

2. Sean las funciones $f_1, f_2, g_1, g_2 : \mathbb{N} \to \mathbb{R}_{\geq 0}$. Demostrar que si $f_1(n) = O(g_1(n))$ y $f_2(n) = O(g_2(n))$ entonces no necesariamente

Variante 1: $f_1(n) - f_2(n) = O(g_1(n) - g_2(n))$.

Variante 2: $f_1(n)/f_2(n) = O(g_1(n)/g_2(n))$.

NOTA: Basado en 3.15 y 3.16 de Manber.

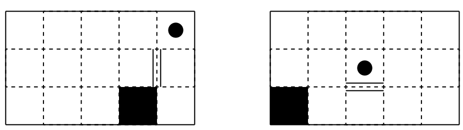
Programación dinámica o BFS

3. Considerar el siguiente juego. Se tiene un laberinto rectangular, formado por casillas. Una casilla es negra, y las demás son blancas. En una de las casillas hay una ficha negra. Algunas casillas pueden estar separadas por vallas de algunas de sus vecinas horizontales o verticales. El objetivo es mover la ficha negra desde su casilla inicial hasta la casilla negra en la menor cantidad posible de movimientos. Un movimiento consiste en desplazar la ficha desde la casilla en la que se encuentra ubicada, a una de sus vecinas horizontales o verticales, siempre que no haya una valla entre ambas casillas. Diseñar un algoritmo eficiente que dado un laberinto indique la mínima cantidad de movimientos necesarios para que la ficha negra ocupe la casilla negra; si tal situación no es posible el algoritmo debe informarlo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(c)$, donde c es la cantidad de casillas del laberinto.

SOLUCIÓN: Armar una matriz con la forma del laberinto, o eventualmente un grafo, y hacer BFS.

TOMADO: 2005C1P1 21-MAY-2005.

4. Considerar la siguiente variante del juego del Ejercicio 3. Se tienen dos laberintos como el descripto. Los rectángulos de ambos laberintos tienen el mismo tamaño, pero usualmente las dos fichas negras, las dos casillas negras y las vallas, no están ubicadas en los mismos lugares. El objetivo es encontrar una secuencia lo más corta posible de movimientos tales que al ser aplicados simultáneamente a las dos fichas negras, ambas ocupen simultáneamente las correspondientes casillas negras. Si un movimiento indica que una ficha negra debe pasar por sobre una valla, esa ficha no se desplaza en ese movimiento. Si un movimiento indica que una ficha negra debe moverse fuera del tablero, se pierde el juego. En el siguiente ejemplo, es posible llevar las dos fichas negras a sus correspondientes casillas negras con la secuencia de 4 movimientos abajo, izquierda, abajo, izquierda.



Diseñar un algoritmo eficiente basado en programación dinámica que dados dos laberintos, indique la mínima cantidad de movimientos necesarios para que las dos fichas negras ocupen simultáneamente las correspondientes casillas negras; si es imposible que eso ocurra el algoritmo debe informarlo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar.

SOLUCIÓN: Numerar las casillas desde 1 en adelante. Considerar una matriz de un número arbitrario de filas (sólo se mantienen las dos últimas), y un número de columnas igual a la cantidad de casillas. En cada posición de la matriz indicar cuáles de las dos fichas negras pueden quedar ubicadas en la casilla correspondiente a la columna, en la cantidad de movimientos correspondiente a la fila. Para calcular cada nueva fila, poner inicialmente que ninguna ficha puede alcanzar ninguna casilla, recorrer la fila anterior, y actualizar la fila nueva considerando para cada casilla los cuatro movimientos posibles (también puede hacerse recorriendo una sola vez la nueva fila, creo). Terminar en cuanto la primera ficha negra alcanza la primera casilla negra, y la segunda hace lo propio. Esto a lo sumo ocurre luego de c^2 iteraciones, que son las combinaciones posibles de las posiciones de las dos fichas. Si no se encontró la solución hasta ese momento, se puede parar. Otra forma es armar un grafo con $O(c^2)$ nodos y ejes; cada nodo representa un estado posible del juego (las posiciones de las fichas negras); para encontrar la solución se hace BFS en tiempo $O(c^2)$; esto puede simularse sobre una matriz de $O(c)$ filas por $O(c)$ columnas, con el mismo costo; cada fila representa la posición de la primera ficha, y cada columna la de la segunda.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2005C1P1 21-MAY-2005.

5. Hay 3 jarras de capacidades c_1, c_2 y c_3. La i-ésima jarra contiene inicialmente una cantidad de agua x_i, y se desea que contenga una cantidad y_i ($i \in \{1, 2, 3\}$). Como no se dispone de ningún elemento adicional (además de las jarras) que permita manejar el agua, la única manera de obtener el resultado deseado es realizar una secuencia de *movimientos* de agua. Un movimiento de agua consiste en vertir cierta cantidad de agua de una jarra (origen) a otra jarra (destino). Pueden ocurrir dos situaciones en un movimiento de agua: si el contenido actual de la jarra origen no es suficiente para completar la jarra destino, todo el contenido de la jarra origen es vertido en la jarra destino; caso contrario, se vierte exactamente la cantidad de agua necesaria para completar la jarra destino. En ningún momento puede usarse agua que no estuviera inicialmente en las jarras, ni tampoco arrojar agua fuera de las jarras. La capacidad de cada jarra, su contenido inicial y el contenido deseado, son todos enteros no negativos. Además las capacidades cumplen que $1 \leq c_1 \leq c_2 \leq c_3$. Diseñar un algoritmo eficiente que dados los valores c_i, x_i y y_i ($i \in \{1, 2, 3\}$), decida si es posible obtener las cantidades de agua deseadas en cada jarra, realizando únicamente movimientos de agua como se describió. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(c_1 \times c_2)$.

   SUGERENCIA: Programación dinámica o camino mínimo (en realidad, ambos).

   SOLUCIÓN: Armar un grafo con un nodo por cada estado posible de las jarras. Agregar ejes de un estado a otro, si desde el primero se puede pasar al segundo con un movimiento de agua. Un estado está determinado por el contenido de las jarras, desde 0 hasta su capacidad, de modo que hay $(c_1 + 1)(c_2 + 1)(c_3 + 1)$ estados. En cada estado, hay 6 maneras distintas de elegir una jarra origen y otra destino. Elegidas esas jarras, la cantidad de agua que pasa de una a otra sólo depende de sus contenidos, de modo que cada nodo tiene grado a lo sumo 6 (el grado puede ser menor que 6 porque algunos pares origen-destino pueden no producir cambios en los contenidos de agua de las jarras, por ejemplo si la jarra origen está vacía, o la jarra destino está llena). Los estados inicial y final están determinados por el contenido inicial y final de las jarras, respectivamente. El problema consiste en decidir si el nodo final es alcanzable desde el nodo inicial. Para ello basta con asignar peso 1 a todos los ejes, y calcular el camino mínimo entre el nodo inicial y el final (por ejemplo con BFS), o expandir un árbol desde el nodo inicial y ver si el nodo final es alcanzado. En ambos casos esto se puede hacer sin armar el grafo, sobre una matriz lógica de tamaño $(c_1 + 1)(c_2 + 1)(c_3 + 1)$, donde cada posición es verdadera si ese estado correspondiente ya fue visitado. El tamaño del grafo o la matriz puede disminuirse notando que el contenido de la tercera jarra es la suma de los contenidos iniciales menos el contenido de las otras dos. Si se hace esto, antes de comenzar hay que verificar que las sumas iniciales y finales coincidan.

   TOMADO: 2003C1R1 18-JUL-2003, 2003C1RX 11-AGO-2003.

Técnicas algorítmicas (excluyendo BFS)

6. Sea $v = (v_1, v_2, \dots v_n)$ un vector de n objetos, con n potencia de 2. Un elemento mayoría de v es un valor que aparece en v más de $n/2$ veces. Notar que v puede tener o no tener elemento mayoría, pero si lo tiene es único.

(a) Demostrar que si $n \geq 2$ y v_m es el elemento mayoría de v, entonces v_m es el elemento mayoría de la primera o de la segunda mitad de v (o de ambas).

(b) Supongamos que los elementos de v se pueden comparar por igualdad en $O(1)$, aunque no existe un orden entre ellos (es decir, no se puede saber si uno es menor o mayor que otro). Se quiere encontrar el elemento mayoría de v, en caso de que exista. Es sencillo resolver este problema en $O(n^2)$, contando cuántas veces aparece en v cada uno de sus elementos. Diseñar un algoritmo de menor complejidad que la mencionada basado en dividir y conquistar. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN:

(a) Si no fuera mayoría en ninguna de las dos mitades, aparecería a lo sumo $n/2$ veces en v.

(b) Utilizar un procedimiento que devuelva el elemento mayoría y cuántas veces aparece, o una marca en caso de que no exista. Los casos bases son con $n = 1$ (el único elemento es mayoría), y eventualmente con $n = 2$ (si los dos elementos son iguales es mayoría). Para el resto llamar recursivamente en cada mitad. Si la mitad izquierda tiene mayoría v_i que aparece n_i veces, contar cuántas veces aparece v_i en la mitad derecha. Si esa cantidad sumada a n_i es mayor que $n/2$ entonces v_i es mayoría. Si la mitad derecha tiene mayoría v_d que aparece n_d veces, contar cuántas veces aparece v_d en la mitad izquierda. Si esa cantidad sumada a n_d es mayor que $n/2$ entonces v_d es mayoría. Caso contrario devolver que no hay mayoría, de acuerdo al punto anterior. El algoritmo resulta $O(n \log n)$. Puede resolverse en $O(n)$.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2010C2R1 15-DIC-2010.

7. Sea $v = (v_1, v_2, \ldots v_n)$ un vector de números enteros, tal que para $i = 1, 2, \ldots n-1$ se cumple que $|v_i - v_{i+1}| \leq 1$. Sea z un número entero tal que $v_1 \leq z \leq v_n$.

   (a) Demostrar que existe un elemento de v que es igual a z.

   (b) Diseñar un algoritmo eficiente basado en dividir y conquistar que encuentre un subíndice j tal que $v_j = z$. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

   SOLUCIÓN: Búsqueda binaria.

   NOTA: Basado en 6.19 de Manber.

   TOMADO: 2005C1R1 22-JUL-2005.

8. Sea $v = (v_1, v_2, \ldots v_n)$ un vector de números reales, todos distintos entre sí. Se sabe que el vector es *unimodal*, es decir, existe un subíndice p entre 1 y n tal que el vector es creciente hasta v_p, y decreciente a partir de v_p. Se desea encontrar el subíndice p. Es sencillo resolver este problema en $O(n)$, recorriendo secuencialmente el vector v. Diseñar un algoritmo de menor complejidad que la mencionada. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(\log n)$.

   SOLUCIÓN: Búsqueda binaria. Analizar cada elemento y sus dos adyacentes. Si el elemento es mayor que sus dos adyacentes, es v_p; si la terna es creciente, v_p está a la derecha del elemento; si la terna es decreciente, v_p está a la izquierda del elemento.

   En realidad alcanza con analizar cada elemento y un adyacente. Si el par es creciente, v_p no está a la izquierda del elemento; si el par es decreciente, v_p no está a la derecha del elemento.

   También es posible usar búsqueda ternaria o alguna de sus variantes.

   NOTA: Encontrado originalmente en Internet.

   TOMADO: 2008C1R1 18-JUL-2008.

9. Dado un vector de números reales $v = (v_1, v_2, \ldots v_n)$, su mediana es un elemento v_m de v tal que al menos la mitad de los elementos de v son menores o iguales que v_m, y al menos la mitad de los elementos de v son mayores o iguales que v_m. Si v está ordenado, su mediana puede encontrarse en $O(1)$.

   Sean x e y dos vectores ordenados de números reales, cada uno conteniendo n elementos. Se desea encontrar la mediana global de los dos vectores (como si los elementos de ambos vectores estuvieran en un único vector de $2n$ elementos). Es sencillo resolver este problema en $O(n)$, obteniendo un nuevo vector ordenado donde aparezcan ordenados los elementos de x e y. Diseñar un algoritmo de menor complejidad que la mencionada. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(\log n)$, y está basado en dividir y conquistar.

SOLUCIÓN: La solución que aparece abajo está al menos parcialmente mal. Puede no haber una mediana en x. Un algoritmo que aparentemente funciona es comparar las medianas de los dos vectores; si son iguales, esa es la mediana; si una es mayor que la otra, se puede descartar una mitad de cada vector. Otro algoritmo posible es tomar la mediana de x y hacer búsqueda binaria en y con ese valor; en base a eso calcular la posición de la mediana de x en el vector global; si es menor que la mediana global, descartar la mitad izquierda de x y repetir; si es mayor que la mediana global, descartar la mitad derecha de x y repetir; el algoritmo resulta $O(\log^2 n)$.

Búsqueda binaria. Hay una mediana en x. Se accede al elemento central de x, digamos x_i, y simultáneamente al elemento y_{n-i}. Si son iguales, esa es la mediana; si x_i es mayor, la mediana no está a la derecha; si x_i es menor, la mediana no está a la izquierda.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2011C2R1 14-DIC-2011.

10. Sea $M \in \mathbb{N}^{m\times n}$ una matriz. Se desea obtener un camino que empiece en la casilla superior izquierda ($[1, 1]$), termine en la casilla inferior derecha ($[m, n]$), y tal que minimice la suma de los valores de las casillas por las que pasa. En cada casilla $[i, j]$ hay dos movimientos posibles: ir hacia abajo (a la casilla $[i + 1, j]$), o ir hacia la derecha (a la casilla $[i, j + 1]$). Diseñar un algoritmo eficiente basado en programación dinámica que resuelva este problema. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(mn)$. Exhibir el comportamiento del algoritmo sobre la matriz que aparece a continuación.

$$\begin{bmatrix} 2 & 8 & 3 & 4 \\ 5 & 3 & 4 & 5 \\ 1 & 2 & 2 & 1 \\ 3 & 4 & 6 & 5 \end{bmatrix}$$

SOLUCIÓN: Definir una matriz $S \in \mathbb{N}^{m\times n}$ que en cada casilla almacena la suma del recorrido mínimo hasta esa casilla. Se tiene $S[1, 1] = M[1, 1]$; $S[1, j] = S[1, j - 1] + M[1, j]$ para $j > 1$; $S[i, 1] = S[i - 1, 1] + M[i, 1]$ para $i > 1$; $S[i, j] = \min\{S[i, j - 1], S[i - 1, j]\} + M[i, j]$ para $i, j > 1$. No es necesario almacenar S de manera completa. Para obtener el camino, se define una matriz adicional que indica de dónde proviene el camino mínimo para cada casilla (izquierda o arriba).

NOTA: Incorporado a la Práctica.

TOMADO: 2004C1P1 22-MAY-2004.

11. Sea $v \in \mathbb{N}^n$ un vector, y sea $w \in \mathbb{N}$. Se desea intercalar entre los elementos de v las operaciones $+$ (suma), $\times$ (multiplicación) y $\uparrow$ (potenciación) de tal manera que al evaluar la expresión obtenida el resultado sea w. Para evaluar la expresión se opera de izquierda a derecha ignorando la precedencia de los operadores. Por ejemplo, si $v = (3, 1, 5, 2, 1)$, y las operaciones elegidas son $+$, $\times$, $\uparrow$ y $\times$ (en ese orden), la expresión obtenida es $3+1\times 5\uparrow 2\times 1$, que se evalua como $(((3+1)\times 5)\uparrow 2)\times 1 = 400$. Diseñar un algoritmo eficiente basado en programación dinámica que dados v y w, encuentre una secuencia de operaciones como la deseada, en caso de que tal secuencia exista. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(nw)$.

SOLUCIÓN: Considerar una matriz rectangular M de n filas por w columnas. Si es posible obtener el valor j operando con los i primeros elementos de v, entonces $M[i, j]$ contiene la última operación utilizada. Caso contrario $M[i, j]$ vale cualquier otra cosa arbitraria. Para saber si es posible se analizan las posiciones $j - i$, j/i y $j^{1/i}$ de la fila $i - 1$ de M, siempre que esas posiciones sean números enteros no negativos.

NOTA: Propuesto originalmente por Flavia Bonomo. Incorporado a la Práctica.

TOMADO: 2003C1P1 17-MAY-2003.

12. Sea $p = (p_1, p_2, \ldots p_n)$ un vector de números reales positivos, donde p_i es el precio de determinado bien el i-ésimo día (dentro de cierto período de días). Se desea saber qué día debería comprarse el bien y qué día debería venderse de manera tal de obtener la mayor ganancia posible. Es decir, se buscan subíndices i^* y j^* tales que $1 \leq i^* \leq j^* \leq n$ y $p_{j^*} - p_{i^*} = \max_{1 \leq i \leq j \leq n} p_j - p_i$. El subíndice i^* indica el día de compra, mientras que el subíndice j^* representa el día de venta. Por ejemplo, para $p = (9, 1, 5)$ la respuesta es "comprar el día $i^* = 2$ (al precio $p_2 = 1$) y vender el día $j^* = 3$ (al precio $p_3 = 5$)". Es sencillo resolver este problema en $O(n^2)$, considerando todas las combinaciones válidas de i y j. Diseñar un algoritmo de menor complejidad que la mencionada. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. Si fuera conveniente, suponer que n es potencia de 2. El mejor algoritmo que conocemos tiene complejidad $O(n)$.

    SOLUCIÓN: Esta explicación es para calcular $\max_{1 \leq i \leq j \leq n} p_j - p_i$. Para calcular efectivamente i^* y j^*, hay que mantener también los subíndices que realizan los valores mencionados en la explicación. Una solución consiste en analizar para cada día si es mejor vender ese día que los días anteriores. Si se vende el primer día la ganancia es 0. Si se vende un día posterior, la mayor ganancia se obtiene comprando al menor precio de todos los días anteriores. En definitiva, se necesita saber el mínimo de cada prefijo de p, lo cual puede calcularse en $O(1)$ para cada día, comparando con el mínimo hasta el día anterior. Otra forma está basada en dividir y conquistar, y calcula recursivamente el mínimo, el máximo, y $\max_{i \leq j} p_j - p_i$ en las dos mitades del vector. El caso base es con $n = 1$ (el mínimo y el máximo son el único elemento del vector, mientras que $\max_{i \leq j} p_j - p_i = 0$). Para $n \geq 2$, el mínimo global es el mínimo de los mínimos de las dos mitades, análogamente el máximo, y $\max_{i \leq j} p_j - p_i$ es el máximo entre la solución en una mitad, en la otra, y la diferencia entre el máximo de la mitad derecha y el mínimo de la mitad izquierda. Esto último se debe a que el óptimo puede comprar y vender en una mitad, en la otra, o comprar en la primera y vender en la segunda, y en este caso lo mejor es comprar en el mínimo de la primera mitad y vender en el máximo de la segunda.

    NOTA: Encontrado originalmente en Internet. Es casi igual al Ejercicio 13.

    TOMADO: 2005C2R1 15-DIC-2005.

13. Fátima ha descubierto que el precio de las acciones de Alcauciles Amalgamados en el k-ésimo día de una secuencia de n días es $p(k) = f(k)$, para $k = 1, 2, \ldots n$ y cierta función conocida $f(k) \in O(1)$. Fátima quiere saber la máxima disminución en el precio de las acciones dentro del período, es decir, $\max_{1 \leq i \leq j \leq n} p(i) - p(j)$. Diseñar un algoritmo que determine ese valor. El algoritmo debe tener complejidad estrictamente mejor que $\Theta(n^2)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n)$ y espacial $O(1)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

    SOLUCIÓN: Para cada j, la máxima disminución es para el máximo precio del prefijo que termina en la posición j. La máxima disminución general es la máxima sobre todos los j. Lo primero puede calcularse en $O(n)$ comparando el máximo para el prefijo anterior con el precio actual, y restando de eso el precio actual. Lo segundo puede calcularse en $O(n)$ recorriendo todos los posibles j. Todo eso puede hacerse recorriendo los precios una única vez, con $O(1)$ variables auxiliares.

    NOTA: Basado fuertemente en el ACM-ICPC World Finals 2015 Problem A. Es casi igual al Ejercicio 12.

    TOMADO: 2015C1R1 13-JUL-2015.

14. Se tienen k vectores ordenados, cada uno de n elementos, con k potencia de 2. Diseñar un algoritmo que los combine en un único vector ordenado de nk elementos, que tenga complejidad estrictamente mejor que $\Theta(nk^2)$, y que esté basado en dividir y conquistar. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(nk \log k)$.

    SOLUCIÓN: Repartir los vectores en 2 grupos de $k/2$ vectores cada uno, combinar cada grupo recursivamente, y finalmente combinar en un único vector los resultados de los 2 grupos. El caso base es $k = 1$, donde no hay nada que hacer y se resuelve en tiempo $T(1) = O(1)$. Como la combinación de los 2 resultados que contienen a todos los elementos puede hacerse en tiempo lineal, para $k \geq 2$ el tiempo es $T(k) = 2T(k/2) + O(nk)$. Puede demostrarse por inducción que $T(k) = O(nk \log k)$ para $k \geq 2$. Otra forma de analizar la complejidad es argumentar que conforme avanza la recursión, los 2 grupos de $k/2$ vectores van a repartirse en 4 grupos de $k/4$ vectores, y así sucesivamente hasta formar $k/2$ pares de vectores de n elementos. Los $k/2$ pares de vectores de n elementos van a combinarse para formar $k/4$ pares de vectores de $2n$ elementos, los cuales van a combinarse para formar $k/8$ pares de vectores de $4n$ elementos, y así sucesivamente hasta combinar un par de vectores de $nk/2$ elementos. El costo de combinar todos los vectores de una misma cantidad de elementos es $O(nk)$ porque es lineal en la cantidad total de elementos, y como las cantidades posibles de elementos son $O(\log k)$, el costo total es $O(nk \log k)$.

Un algoritmo con la complejidad pedida pero que no usa dividir y conquistar se basa en que en el vector combinado los elementos de cada vector aparecen en el orden original. Una posibilidad es entonces inicializar un heap con el primer elemento de cada vector, y repetidamente extraer el mínimo del heap e insertar el elemento siguiente del vector al cual pertenecía el elemento extraído (mientras haya elementos en el vector). Inicializar el heap es barato y luego se hacen $O(nk)$ operaciones de costo $O(\log k)$, de modo que el costo total es el buscado. Un algoritmo similar pero con costo mayor consiste en ignorar completamente que cada vector está ordenado, insertar todos los elementos en un árbol binario de búsqueda balanceado, y luego recorrer el árbol en orden. La complejidad de este algoritmo es $O(nk \log(nk))$.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2013C1R1 17-JUL-2013.

15. Un intervalo unitario es un subconjunto de números reales comprendidos entre dos valores a distancia 1. Más precisamente, para cualquier $a \in \mathbb{R}$, el conjunto $[a, a+1] = \{r \in \mathbb{R} \text{ tal que } a \leq r \leq a+1\}$ es un intervalo unitario, y todos los intervalos unitarios se describen de esta manera.

    Sea $v = (v_1, v_2, \ldots v_n)$ un vector de números reales. Se desea encontrar la menor cantidad posible de intervalos unitarios tales que cada elemento de v pertenece al menos a uno de esos intervalos. Este problema se conoce como CUBRIMIENTO POR INTERVALOS UNITARIOS.

    (a) Considerar la siguiente estrategia golosa para el problema: mientras haya elementos en v, elegir un intervalo unitario que contenga la mayor cantidad posible de elementos de v, y eliminar de v los elementos que pertenecen al intervalo elegido. Demostrar que esta estrategia no es un algoritmo exacto.

    (b) Diseñar un algoritmo exacto, goloso y eficiente que resuelva el problema. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n \log n)$.

    SOLUCIÓN: El algoritmo propuesto utiliza 3 intervalos para $v = (0, 0.5, 0.9, 1.1, 1.5, 2)$, cuando son suficientes 2 intervalos. Un posible algoritmo comienza ordenando v. Luego elige un intervalo que empiece en el menor elemento de v, remueve de v todos los elementos cubiertos por ese intervalo, y repite el proceso hasta que no queden elementos en v. La elección de los intervalos puede implementarse en $O(n)$ simplemente recorriendo el vector ordenado, de manera que el algoritmo completo es $O(n \log n)$, que es el costo de ordenar.

    NOTA: Encontrado originalmente en Internet.

    TOMADO: 2009C1R1 19-AGO-2009.

16. Sea $v = (v_1, v_2, \ldots v_n)$ un vector de números reales. Para cada par de enteros p y q tales que $1 \leq p \leq q \leq n$, definimos $\text{RMQ}(v, p, q) = \min_{p \leq i \leq q} v_i$ (RMQ significa Range Minimum Query).

    (a) Diseñar un algoritmo eficiente que dados v, p y q, calcule $\text{RMQ}(v, p, q)$. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n)$.

    (b) Diseñar un algoritmo eficiente basado en programación dinámica que dado el vector v, lo preprocese adecuadamente de manera tal que luego pueda calcularse $\text{RMQ}(v, p, q)$ en $O(1)$ para cualquier par de enteros p y q válidos. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n)$, aunque es muy complicado.

    TOMADO: 2001C2P1 20-OCT-2001.

17. Sea $v = (v_1, v_2, \ldots v_n)$ un vector de números reales. Para cada par de enteros p y q tales que $1 \leq p \leq q \leq n$, definimos $\text{suma}(v, p, q) = \sum_{p \leq i \leq q} v_i$. Diseñar un algoritmo que dado el vector v,

lo preprocese adecuadamente de manera tal que luego pueda calcularse suma(v, p, q) en $O(1)$ para cualquier par de enteros p y q válidos. El algoritmo debe tener complejidad estrictamente mejor que $\Theta(n^2)$. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(n)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

SOLUCIÓN: Calcular la suma de cada prefijo en $O(1)$ cada una. Hecho esto, cualquier suma puede calcularse como diferencia de la suma de dos prefijos. El caso $p = 1$ es especial porque el prefijo a restar es nulo.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Es un caso particular del Ejercicio 18.

TOMADO: 2014C2R1 12-DIC-2014.

18. Dada $M \in \mathbb{R}^{m\times n}$, junto con $i_1, i_2, j_1, j_2 \in \mathbb{Z}$ tales que $1 \leq i_1 \leq i_2 \leq m$ y $1 \leq j_1 \leq j_2 \leq n$, definimos suma$(M, i_1, i_2, j_1, j_2) = \sum_{i_1\leq i\leq i_2} \sum_{j_1\leq j\leq j_2} M_{i,j}$. Diseñar un algoritmo que preprocese la matriz M con complejidad estrictamente mejor que $\Theta(m^2n^2)$, de manera tal que luego pueda calcularse suma(M, i_1, i_2, j_1, j_2) con complejidad estrictamente mejor que $\Theta(mn)$ para cualquier entrada válida. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad $O(mn)$ para el preprocesamiento y $O(1)$ para calcular cada suma, todo lo cual es necesario para obtener puntaje máximo en este ejercicio.

    SOLUCIÓN: Definir suma$(M, i_1, i_2, j_1, j_2) = 0$ si $i_2 = 0$ o $j_2 = 0$. Calcular suma$(M, 1, i, 1, j) = M_{i,j} +$ suma$(M, 1, i-1, 1, j) +$ suma$(M, 1, i, 1, j-1) -$ suma$(M, 1, i-1, 1, j-1)$ en $O(1)$ cada una. Hecho esto, cualquier suma puede calcularse como suma y resta de sumas ya calculadas, ya que suma$(M, i_1, i_2, j_1, j_2) =$ suma$(M, 1, i_1-1, 1, j_1-1) +$ suma$(M, 1, i_2, 1, j_2) -$ suma$(M, 1, i_1-1, 1, j_2) -$ suma$(M, 1, i_2, 1, j_1-1)$. La complejidad es $O(mn)$ para el preprocesamiento y $O(1)$ para cada consulta.

    Otra posibilidad es preprocesar cada fila como indica el Ejercicio 17, y descomponer cada consulta en $O(m)$ consultas del Ejercicio 17. La complejidad es $O(mn)$ para el preprocesamiento y $O(m)$ para cada consulta.

    Se puede operar de forma análoga sobre las columnas, con complejidad $O(mn)$ para el preprocesamiento y $O(n)$ para cada consulta.

    NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Generaliza al Ejercicio 17.

    TOMADO: 2017C1R1 14-JUL-2017.

19. Dadas dos cadenas de caracteres $s = (s[1], s[2], \ldots s[p])$ y $t = (t[1], t[2], \ldots t[q])$, se dice que t es una subsecuencia de s si y sólo si existen q valores $i_1 < i_2 \cdots < i_q$ tales que $t[j] = s[i_j]$ para $j = 1, 2, \ldots q$. Por ejemplo, "`hoyeselfindelmundo`" tiene como subsecuencias a "`oye`", "`feudo`" y "", pero no a "`nifles`".

    Sean v y w dos cadenas de caracteres. Sean $S(v)$ y $S(w)$ los conjuntos de subsecuencias de v y w, respectivamente. Sea $S(v, w) = S(v) \cap S(w)$, es decir, el conjunto de cadenas de caracteres que son subsecuencias tanto de v como de w. Sea LCS(v, w) la longitud máxima de las cadenas de $S(v, w)$, es decir, la longitud de la subsecuencia más larga que aparece tanto en v como en w (LCS significa Longest Common Subsequence).

Variante 1: Diseñar un algoritmo eficiente basado en programación dinámica que dadas v y w, calcule LCS(v, w). Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(pq)$ y espacial $O(\min\{p, q\})$, donde p es la longitud de v, y q es la longitud de w.

Variante 2: Dado $k \in \mathbb{N}$ y dos cadenas de caracteres v y w, ambas de longitud ℓ, se desea calcular LCS(v, w) si este valor es por lo menos $\ell - k$; si ocurriera que LCS(v, w) es menor que $\ell - k$, no interesa el valor exacto. Diseñar un algoritmo eficiente basado en programación dinámica que resuelva este problema. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(k\ell)$ y espacial $O(k)$.

SOLUCIÓN:

Variante 1: Definir $M[i,j] = \mathrm{LCS}((v[1], v[2], \dots v[i]), (w[1], w[2], \dots w[j]))$. Se cumple $M[i,0] = 0$; $M[0,j] = 0$; $M[i,j] = \max\{M[i-1,j], M[i,j-1], M[i-1,j-1] + \delta_{v[i],w[j]}\}$ (delta de Kronecker).

Variante 2: Puede resolverse de la misma manera, pero sólo es necesario calcular una banda de ancho $O(k)$.

TOMADO: 2001C2RX 29-DIC-2001, 2002C1P1 18-MAY-2002 (Variante 1), 2002C1R1 17-JUL-2002 (Variante 2).

20. Una empresa ha decidido invertir dinero en algunas de sus sucursales. A fin de decidir la forma de hacerlo, el directorio ha pedido a cada sucursal que elabore un conjunto de entre 1 y 3 propuestas de cambios que la sucursal puede realizar, indicando para cada propuesta el costo estimado y la ganancia esperada si los cambios correspondientes se llevan a cabo. El directorio quiere elegir, de entre todos los subconjuntos posibles de propuestas, aquel que maximice la suma de las ganancias esperadas, pero con la restricción adicional de que se financie a lo sumo una propuesta por cada sucursal. Diseñar un algoritmo eficiente basado en programación dinámica que resuelva este problema. La entrada del algoritmo está formada por la cantidad de sucursales ($S \in \mathbb{N}$), el dinero disponible ($D \in \mathbb{N}$), y las propuestas de las sucursales; cada propuesta es una terna (s, c, g), donde s es la sucursal que hizo la propuesta, $c \in \mathbb{N}$ es el costo estimado de la propuesta, y $g \in \mathbb{N}$ es la ganancia esperada de la propuesta. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(S \times D)$.

    SUGERENCIA: Considerar una matriz rectangular M de $O(S) \times O(D)$ elementos, donde $M[i,j]$ representa el mejor subconjunto de propuestas restringido a considerar las primeras i sucursales, y tal que llevar a cabo todas las propuestas de ese subconjunto cuesta a lo sumo j.

    NOTA: Si en el input las propuestas no están ordenadas por sucursal, es posible hacerlo en $O(S)$ usando bucket-sort.

    TOMADO: 2004C1R1 21-JUL-2004.

21. Sea $v = (v_1, v_2, \dots v_n)$ un vector de números reales positivos, todos distintos entre sí. Decimos que un subconjunto de elementos de v es *independiente* si y sólo si todo par de elementos del subconjunto no aparece en posiciones consecutivas de v. Diseñar un algoritmo eficiente que encuentre un subconjunto de elementos independiente cuya suma sea máxima. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(n)$, y está basado en programación dinámica.

    SOLUCIÓN: Es un caso particular del Ejercicio 25. Lo que sigue es para calcular el valor del óptimo. Para determinar el subconjunto proceder de la manera habitual almacenando la opción elegida en la recursión.

    Una posibilidad es definir $f(i)$ como la solución restringida a las primeras i posiciones de v. Se cumple $f(1) = v_1$; $f(2) = \max\{v_1, v_2\}$; $f(i) = \max\{f(i-1), f(i-2) + v_i\}$ para $i = 3, 4, \dots n$. Esto funciona ya que si se decide no agregar el elemento actual, lo mejor es tomar el óptimo hasta la posición anterior, mientras que si se decide agregarlo, el elemento anterior no se puede agregar, y lo mejor es tomar el óptimo hasta la segunda posición anterior. El algoritmo puede modificarse para admitir números negativos, incluyendo entre los parámetros del máximo, el óptimo de la segunda posición anterior sin sumar el elemento actual (que podría ser negativo, en cuyo caso nunca conviene agregarlo).

    Otra posibilidad es pensarlo como un problema de camino máximo con dos estados por cada posición i de v, uno que elige al elemento v_i, y otro que lo descarta. Al estado que descarta a v_i se puede pasar desde el estado que descarta o desde el que elige a v_{i-1}, mientras que al estado que elige a v_i sólo se puede pasar desde el estado que descarta a v_{i-1}. Eso sugiere las definiciones $d(1) = 0$ y $e(1) = v_1$; $d(i) = \max\{d(i-1), e(i-1)\}$ y $e(i) = d(i-1) + v_i$ para $i = 2, 3, \dots n$. La solución definitiva es $f(i) = \max\{d(i), e(i)\}$ para $i = 1, 2, \dots n$. Se puede demostrar por inducción en i que esta solución coincide con la anterior.

    Otra solución peor es armar un grafo con un nodo por cada número, y un eje de un nodo a los nodos posteriores que podrían formar parte del mismo subconjunto independiente (es decir, todos menos el inmediato posterior). Se agrega también un nodo inicial con ejes a todos los otros nodos, y un nodo final con ejes desde todos los otros nodos. Los ejes que salen de un nodo se etiquetan con el número asociado al nodo (o con 0 si se trata del nodo inicial). Para encontrar el subconjunto se calcula camino máximo, que en un digrafo sin circuitos puede hacerse en $O(m)$, que en este caso es $O(n^2)$.

NOTA: Encontrado originalmente en Internet. Los números son distintos al solo efecto de que sus posiciones no sean ambiguas.

TOMADO: 2007C1P1 19-MAY-2007, 2013C1P1 18-MAY-2013.

22. Tarzán se desplaza por la selva balanceandose de liana en liana. Usualmente, de cada liana pasa a otra adyacente, aunque es capaz de pasar por alto una liana si toma suficiente envión en la liana inmediatamente anterior. Esta mañana Tarzán quiere ir desde su casa del árbol hasta la cascada. Sea n la cantidad de lianas en el camino entre esos dos sitios, y sea $p_i \in [0, 1]$ la probabilidad de que la i-ésima liana no se rompa cuando Tarzán se cuelga de ella ($i = 1, 2, \ldots n$). Diseñar un algoritmo eficiente que elija las lianas que debería utilizar Tarzán a fin de maximizar la probabilidad de llegar a destino sin que se rompa ninguna liana. Si Tarzán utiliza las L lianas $i_1, i_2, \ldots i_L$, dicha probabilidad está dada por $\prod_{j=1}^{L} p_{i_j}$. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(n)$, y está basado en programación dinámica.

    SOLUCIÓN: Calcular un nuevo vector que en cada posición tiene la solución restringida al prefijo del vector original hasta esa posición, con la restricción adicional de utilizar la liana de esa posición. Finalmente, tomar el máximo de las últimas dos posiciones (o agregar una liana ficticia de probabilidad 1). La primera y la segunda posición se calculan de manera inmediata. Para cada nueva posición tomar el máximo entre las dos posiciones anteriores, y multiplicarlo por el valor del elemento actual. Esto funciona ya que si se decide utilizar la liana actual, la anterior puede o no usarse, y hay que llegar a ese lugar de manera óptima. Otra forma es armar un grafo y hacer algo parecido a camino máximo, pero es más raro.

    NOTA: Es casi igual al Ejercicio 23.

    TOMADO: 2008C1P1 17-MAY-2008.

23. El sapo Pepe quiere cruzar un río. Del agua asoman n objetos formando un camino entre las dos orillas, a los cuales Pepe puede saltar para cruzar. Para cualquier par de objetos consecutivos, Pepe debe saltar al menos a uno de ellos, ya que no es capaz de saltar demasiado lejos. Sea $p_i \in [0, 1]$ la probabilidad de que el i-ésimo objeto se hunda cuando Pepe salta a ese objeto. Diseñar un algoritmo que elija los objetos a los cuales Pepe debe saltar a fin de maximizar la probabilidad de que ninguno se hunda. Si Pepe salta a los k objetos $i_1, i_2, \ldots i_k$, dicha probabilidad está dada por $\prod_{j=1}^{k} \left(1 - p_{i_j}\right)$. El algoritmo debe tener complejidad $O(n)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

    NOTA: Es casi igual al Ejercicio 22.

    TOMADO: 2018C2R1 12-DIC-2018.

24. Luego de los cursos en Europa, el Magnate de la Bondiola logró ganar muchísimo dinero con los puestos que tiene en la franja de la Costanera que está al sur de Ciudad Universitaria. Su consejera espiritual le recomendó expandir y diversificar su actividad. Es así que el Magnate compró los n negocios de una cuadra del barrio de Belgrano, con el objetivo de instalar comercios en cada uno de ellos. Cada comercio puede ser de uno de los siguientes 3 tipos: venta y reparación de computadoras (1); venta de lencería (2); puesto de bondiola (3). Se conoce el costo c_{ij} de instalar un comercio tipo i en el j-ésimo negocio de la cuadra ($i = 1, 2, 3$; $j = 1, 2, \ldots n$). El Magnate quiere instalar los n comercios, pero sin que haya dos o más comercios consecutivos del mismo tipo, ya que competirían entre sí. Diseñar un algoritmo eficiente que determine el costo mínimo para instalar los n comercios teniendo en cuenta la restricción mencionada. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n)$ y espacial $O(1)$ (adicional a los datos).

    SOLUCIÓN: En esencia es camino mínimo en un digrafo sin ciclos. El problema puede representarse con un grafo con un nodo por cada par (i, j), con arcos desde $(i', j-1)$ con $i' \neq i$. Los arcos que llegan (o salen, si se prefiere) de (i, j) se etiquetan con c_{ij}. Si es necesario se agrega un nodo inicial y otro final, y se calcula el camino mínimo, lo cual puede hacerse con costo proporcional a n usando el algoritmo que se usa en control de proyectos. Equivalentemente se puede plantear un algoritmo de programación dinámica, definiendo una matriz M que en la posición $M[i, j]$ almacene el costo mínimo de instalar los primeros j comercios con el j-ésimo de tipo i. Se tiene $M[i, 0] = 0$; $M[i, j] = c_{ij} + \min_{i' \neq i}\{M[i', j-1]\}$ para $j \geq 1$. El resultado es $\min_i M[i, n]$.

NOTA: Propuesto originalmente por Michel Mizrahi.

TOMADO: 2012C1R1 18-JUL-2012.

25. El Magnate de la Bondiola piensa reventar la Costanera instalando puestos en la franja que está al sur de Ciudad Universitaria. Tiene una lista de n posibles lugares donde ubicar sus puestos. Para cada posible lugar conoce la distancia d_i en metros desde la puerta del Pabellón 1 de Ciudad Universitaria, y una estimación $b_i > 0$ del beneficio que le daría un puesto instalado ahí ($i = 1, 2, \ldots n$). Esta información está ordenada de forma creciente por distancia. El empresario tiene dinero suficiente para instalar los n puestos. Sin embargo, sabe que si dos puestos están muy cerca, el beneficio de cada uno se reduce, de modo que no quiere que ningún par de puestos instalados quede a menos de m metros de distancia uno de otro. Esto significa que si dos puestos $i_1 < i_2$ son instalados, entonces debe cumplirse que $d_{i_2} - d_{i_1} \geq m$. Diseñar un algoritmo que determine el mayor beneficio estimado que se puede obtener con la instalación de los puestos. El algoritmo debe tener complejidad $O(n^2)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial $O(n)$.

    SOLUCIÓN: Definir un vector B que en la posición i-ésima almacena el máximo beneficio esperado restringido al prefijo de tamaño i y tal que además se instala el puesto i-ésimo. Definir otro vector M que en la posición i-ésima almacena el máximo del prefijo de tamaño i de B. Es claro que la solución es $M[n]$, ya que instalar al menos un puesto es mejor que no instalar ninguno. Para calcular $B[i]$ tomar b_i y sumarle $B[j]$ donde j es el mayor índice $j < i$ tal que $d_i - d_j \geq m$. Para calcular $M[i]$ tomar el máximo entre $M[i-1]$ y $B[i]$. Para determinar j en cada paso, mantener una variable que empieza en 0 y que en cada paso crece mientras no se viole la condición. El costo amortizado de mantener esta variable es $O(1)$. Alternativamente se puede buscar un nuevo j en cada paso restando repetidamente 1 de i, o haciendo búsqueda binaria, con un costo por iteración de $O(n)$ u $O(\log n)$ respectivamente. Hay formas alternativas de resolverlo. Por ejemplo, calcular para cada posición el óptimo del prefijo que termina en ella, que es el máximo entre el óptimo del prefijo anterior (no se instala el puesto actual), y el beneficio del puesto actual sumado al óptimo del prefijo que termina en el mayor puesto anterior al actual a distancia al menos m (se instala).

    NOTA: Propuesto originalmente por Martiniano Eguía.

    TOMADO: 2010C2P1 16-OCT-2010.

26. Graph Computing es una empresa cuyos integrantes se organizan como en una jerarquía clásica (árbol dirigido con raíz). La empresa está por lanzar al mercado un nuevo y revolucionario microprocesador de doble núcleo. Debido a esto, los directivos quieren organizar una fiesta lo más divertida posible. Inicialmente se pensó en invitar a todos los integrantes de la empresa, pero esta idea luego se descartó porque aflorarían problemas laborales que harían que la fiesta no fuera divertida. Se decidió por lo tanto que no asistieran simultáneamente un empleado y su superior inmediato. Como muchos conjuntos de personas cumplen con esa restricción, se calificó a cada persona con un número real no negativo que indica el grado de diversión de esa persona, y se decidió invitar al conjunto de personas que maximizara la suma de los grados de diversión de las personas del conjunto. Diseñar un algoritmo eficiente que decida quiénes asistirán a la fiesta. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n)$, donde n es la cantidad de integrantes de la empresa, y está basado en programación dinámica.

    SOLUCIÓN: Cada nodo desde la raiz se calcula recursivamente como el máximo de: el peso de ese nodo más la suma de los nietos (si el nodo se incluye), y la suma de los hijos (si el nodo no se incluye).

    NOTA: Propuesto originalmente por Leandro Montero.

    TOMADO: 2007C1R1 17-JUL-2007.

27. Sea $v = (v_1, v_2, \ldots v_n) \in \mathbb{N}^n$ un vector conteniendo los valores (distintos entre sí) de las monedas en circulación. Se dispone de una cantidad arbitrariamente grande de monedas de cada valor. Diseñar un algoritmo eficiente basado en programación dinámica que determine la cantidad de

maneras distintas de formar con esas monedas un importe dado $w \in \mathbb{N}$, con $w \geq \max_{i=1,2,\dots n} v_i$. Ejemplos:

- Para $n = 1$, $v_1 = 4$ y $w = 7$, el resultado es 0.
- Para $n = 1$, $v_1 = 4$ y $w = 8$, el resultado es 1.
- Para $n = 2$, $v_1 = 4$, $v_2 = 2$ y $w = 8$, el resultado es 3.

Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(nw)$ y espacial $O(w)$.

SOLUCIÓN: Sea $F(i, j)$ la cantidad de maneras de formar el importe j eligiendo monedas entre los primeros i valores en circulación. Queremos $F(n, w)$. Se cumple $F(0, 0) = 1$; $F(0, j) = 0$ para $j = 1, 2, \dots w$; $F(i, j) = \text{IF}(j \geq v_i; F(i-1, j) + F(i, j - v_i); F(i-1, j))$ para $i = 1, 2, \dots n$ y $j = 0, 1, \dots w$. La parte recursiva es la suma de la cantidad de maneras que no usan v_i, más la cantidad de maneras que usan al menos una vez v_i. Eventualmente se puede empezar por $i = 1$ en vez de 0 definiendo $F(1, j) = \text{IF}(j \bmod v_1 = 0; 1; 0)$ para $j = 0, 1, \dots w$, con el mismo resultado. Se puede implementar sobre una matriz, manteniendo una única fila a la vez. Como cada posición se calcula en $O(1)$, las complejidades son las esperadas. Se puede implementar sobre un único vector, iterando sobre i y luego sobre j, digamos $f(j) = \text{IF}(j \geq v_i; f(j) + f(j - v_i); f(j))$.

Una forma alternativa es $F(i, j) = \sum_{k=0,1,\dots j/v_i} F(i-1, j - kv_i)$ que representa la suma de las cantidades de maneras que usan k veces v_i, y el resto lo devuelven con las otras monedas. Esta forma es equivalente a la anterior ya que mientras sea posible restar repetidamente v_i de j, usando la expresión $F(i, j) = F(i-1, j) + F(i, j - v_i)$ resulta

$$
\begin{array}{rcccccccl}
F(i,j) & = & F(i-1,j) & + & F(i,j-v_i) & & & & \\
 & = & F(i-1,j) & + & F(i-1,j-v_i) & + & F(i,j-2v_i) & & \\
 & = & F(i-1,j) & + & F(i-1,j-v_i) & + & F(i-1,j-2v_i) & + & F(i,j-3v_i) \\
 & = & \cdots & & & & & &
\end{array}
$$

Se puede implementar sobre una matriz, manteniendo dos filas a la vez, por lo que la complejidad espacial es la misma. Cada posición (i, j) se calcula en $O(j/v_i) = O(w/v_i)$, de modo que la fila i completa se calcula en $O(w^2/v_i)$, y en total tenemos $O(w^2 \sum_{i=1}^{n} 1/v_i) = O(w^2 \sum_{i=1}^{n} 1/i) = O(w^2 \log n)$ ya que los valores v_i son distintos entre sí.

La complejidad temporal de la segunda forma es peor que la de la primera (ya que $w \geq \max_{i=1,2,\dots n} v_i \geq n$), pero la segunda forma se puede adaptar para el caso en que la cantidad de monedas de cada valor es finita, tomando k desde 0 hasta la cantidad de monedas de cada valor. No es claro que la primera forma se pueda adaptar de manera similar, manteniendo su complejidad.

NOTA: Propuesto originalmente por Darío Guzik.

TOMADO: 2009C1P1 16-MAY-2009.

28. Sea $v = (v_1, v_2, \dots v_n) \in \mathbb{N}^n$ un vector conteniendo los valores (distintos entre sí) de las monedas en circulación, y sea $c = (c_1, c_2, \dots c_n) \in \mathbb{N}^n$ 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 $w \in \mathbb{N}$, con $w \geq \max_{i=1,2,\dots n} v_i$. 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 $O(\min\{sw, w^2 \log n\})$ y espacial $O(w)$, donde $s = \sum_{i=1}^{n} c_i$.

SOLUCIÓN: Una posibilidad es considerar una lista $m_1, m_2, \dots m_s$ en la que para cada i hay c_i copias del valor v_i. No es necesario armar explícitamente esta lista. De manera similar al Ejercicio 3.15 de la Práctica, definir $f(i, j)$ como la mínima cantidad de monedas que permiten formar el importe j eligiendo valores entre los primeros i de la lista. Queremos $f(s, w)$. Se cumple $f(0, 0) = 0$; $f(0, j) = +\infty$ para $j = 1, 2, \dots w$; $f(i, j) = \text{IF}(j \geq m_i; \min\{f(i-1, j), f(i-1, j - m_i)\}; f(i-1, j))$

para $i = 1, 2, \dots s$ y $j = 0, 1, \dots w$. Se puede implementar sobre una matriz, manteniendo dos filas a la vez, por lo que la complejidad espacial es $O(w)$. Como cada posición se calcula en $O(1)$, la complejidad temporal es $O(sw)$.

Otra posibilidad es definir $g(i, j)$ como la mínima cantidad de monedas que permiten formar el importe j eligiendo monedas entre los primeros i valores en circulación. Queremos $g(n, w)$. Se cumple $g(0, 0) = 0$; $g(0, j) = +\infty$ para $j = 1, 2, \dots w$; $g(i, j) = \min_{k=0,1,\dots \min\{c_i, j/v_i\}}\{g(i-1, j-kv_i)\}$ para $i = 1, 2, \dots n$ y $j = 0, 1, \dots w$. Se puede implementar sobre una matriz, manteniendo dos filas a la vez, por lo que la complejidad espacial es $O(w)$. Respecto de la complejidad temporal, cada posición (i, j) se calcula en $O(\min\{c_i, j/v_i\}) = O(\min\{c_i, w/v_i\})$, de modo que la fila i completa se calcula en $O(\min\{wc_i, w^2/v_i\})$, y en total tenemos $O(\min\{w\sum_{i=1}^{n} c_i, w^2\sum_{i=1}^{n} 1/v_i\}) = O(\min\{sw, w^2\sum_{i=1}^{n} 1/i\}) = O(\min\{sw, w^2 \log n\})$ ya que los valores v_i son distintos entre sí.

NOTA: Propuesto originalmente por Michel Mizrahi. Es un caso particular del Ejercicio 29.

TOMADO: 2013C2P1 12-OCT-2013.

29. Sea $v = (v_1, v_2, \dots v_n) \in \mathbb{N}^n$ un vector conteniendo los valores (distintos entre sí) de las monedas en circulación, y sea $c = (c_1, c_2, \dots c_n) \in (\mathbb{N} \cup \{+\infty\})^n$ 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 $w \in \mathbb{N}$, con $w \geq \max_{i=1,2,\dots n} v_i$. Si es imposible formar ese importe, el algoritmo debe indicarlo. Ejemplos:

   - Para $n = 2$, $v = (5, 1)$, $c = (1, +\infty)$ y $w = 10$, el resultado es 6.
   - Para $n = 2$, $v = (5, 3)$, $c = (+\infty, +\infty)$ y $w = 7$, es imposible formar el importe w.

   Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar.

   SOLUCIÓN: Sean $s_{\mathbb{N}} = \sum_{c_i \in \mathbb{N}} c_i$, $s_\infty = \sum_{c_i = +\infty} 1$, y $s = s_{\mathbb{N}} + s_\infty$.

   Una posibilidad es considerar una lista $m_1, m_2, \dots m_s$ en la que para cada i hay c_i copias del valor v_i si $c_i \in \mathbb{N}$ y una copia si $c_i = +\infty$. No es necesario armar explícitamente esta lista. Luego podemos definir $f(i, j)$ como en el primer algoritmo propuesto para el Ejercicio 28. Queremos $f(s, w)$. Se cumple $f(0, 0) = 0$; $f(0, j) = +\infty$ para $j = 1, 2, \dots w$; $f(i, j) = \mathrm{IF}(j \geq m_i; \min\{f(i-1, j), f(i-d_i, j-m_i)\}; f(i-1, j))$ para $i = 1, 2, \dots s$ y $j = 0, 1, \dots w$, donde $d_i = 0$ si hay infinitas monedas del valor m_i y $d_i = 1$ caso contrario. Tal como vimos en el Ejercicio 28, la complejidad temporal es $O(sw)$ y la espacial es $O(w)$.

   Otra posibilidad es utilizar directamente el segundo algoritmo propuesto para el Ejercicio 28. La complejidad espacial es $O(w)$ como en ese ejercicio. Respecto de la complejidad temporal, la fila i completa se calcula en $O(\min\{wc_i, w^2/v_i\})$ si $c_i \in \mathbb{N}$, y en $O(w^2/v_i)$ si $c_i = +\infty$. El total es $O(\min\{s_{\mathbb{N}}w + w^2 \log s_\infty, w^2 \log n\})$.

   Una tercera posibilidad es calcular para $j = 0, 1, \dots w$ el mínimo de formar j con las monedas finitas, y el resto con las infinitas, de manera óptima en cada parte. La primera parte se puede calcular con cualquiera de los dos algoritmos propuestos para el Ejercicio 28; la complejidad espacial de esta parte es $O(w)$ en ambos casos, mientras que la complejidad temporal es $O(s_{\mathbb{N}}w)$ si se utiliza el primer algoritmo, y $O(\min\{s_{\mathbb{N}}w, w^2 \log(n - s_\infty)\})$ si se utiliza el segundo. La segunda parte se puede calcular como en el Ejercicio 3.10 de la Práctica con complejidad espacial $O(w)$ y temporal $O(s_\infty w)$. Finalmente, se puede elegir el mínimo en $O(w)$.

   NOTA: Propuesto originalmente por Michel Mizrahi. Generaliza al Ejercicio 28.

   TOMADO: 2013C2R1 16-DIC-2013.

30. Es necesario realizar n cortes transversales en un bloque de granito de ℓ cm de longitud. Los cortes están representados por sus distancias al borde izquierdo del bloque, y se encuentran ordenados de izquierda a derecha. El granito es irregular, de modo que no es posible reubicar los cortes para obtener porciones similares a las que se obtienen con los cortes previstos. Además, el granito es muy denso y por lo tanto difícil de manejar. En consecuencia el costo de cada corte es proporcional a la longitud del bloque que se está cortando. Si bien los cortes no pueden reubicarse, es posible reducir el costo total si los mismos se realizan en un orden adecuado. Supongamos que se quieren realizar $n = 2$ cortes ubicados a 100 cm y a 250 cm del borde izquierdo de un bloque de 300 cm de

longitud. Si se corta primero a los 250 cm del borde izquierdo se tiene un costo de 300 para ese corte; se obtienen dos porciones, una de 250 cm de longitud y otra de 50 cm de longitud; la primera porción debe cortarse a los 100 cm de su borde izquierdo, con un costo adicional de 250; el costo total es $300 + 250 = 550$. Por el contrario, si se corta primero a los 100 cm del borde izquierdo se tiene un costo de 300 para ese corte; se obtienen dos porciones, una de 100 cm de longitud y otra de 200 cm de longitud; la segunda porción debe cortarse a los $250 - 100$ cm de su borde izquierdo, con un costo adicional de 200; el costo total es $300 + 200 = 500$. Diseñar un algoritmo eficiente basado en programación dinámica que determine el mínimo costo con el que se pueden realizar los cortes. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar.

SUGERENCIA: Definir una matriz M tal que $M[i, j]$ es el costo mínimo para realizar los cortes en la porción del bloque que va desde el corte i-ésimo hasta el corte j-ésimo.

SOLUCIÓN: $M[i, j] = \ell_{ij} + \min_{i<k<j}\{M[i, k] + M[k, j]\}$, donde ℓ_{ij} es la longitud de la porción que va desde el corte i-ésimo hasta el corte j-ésimo.

NOTA: Propuesto originalmente por Federico Lebrón.

TOMADO: 2011C2P1 15-OCT-2011.

31. Una cadena de caracteres correcta (CCC) es una cadena s que cumple una de las siguientes condiciones:

    - es nula (tiene longitud cero);
    - es de la forma "`(`c`)`", "`[`c`]`" o "`{`c`}`", donde "c" es una CCC;
    - es la concatenación de dos CCC, cada una de menor longitud que la de s.

    Por ejemplo, "", "`([]{})`" y "`()[]{}`" son CCC, mientras que "`][`", "`(()`", "`{]`" y "`(`" no lo son. Notar que toda CCC tiene longitud par, y que cualquier cadena de caracteres de longitud par puede transformarse en una CCC modificando algunos de sus caracteres (eventualmente ninguno). Diseñar un algoritmo eficiente que dada una cadena de caracteres de longitud par formada únicamente por los caracteres contenidos en "`()[]{}`", indique la mínima cantidad de caracteres de la cadena que es necesario modificar a fin de transformarla en una CCC. Por ejemplo, el resultado de aplicar el algoritmo a "", "`][`" y "`(()`" debe ser respectivamente 0, 2 y 1. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n^3)$, donde n es la longitud de la cadena de caracteres, y está basado en programación dinámica.

    SOLUCIÓN: Básicamente es aplicar la definición de CCC. Sea $f(i, j)$ que indica el valor buscado para la subcadena que va de la posición i a la j. Se tiene $f(i, i-1) = 0$ (subcadena nula); $f(i, j) = \min\{g(i, j) + f(i+1, j-1), \min_{k=i+1,i+3,\ldots j-2}\{f(i, k) + f(k+1, j)\}\}$, donde $j - i + 1 \geq 2$ es par, y $g(i, j) \in \{0, 1, 2\}$ es el costo mínimo de emparejar los caracteres $s[i]$ y $s[j]$.

    NOTA: Propuesto originalmente por Michel Mizrahi.

    TOMADO: 2012C1P1 19-MAY-2012.

32. Se tiene una expresión lógica formada por n constantes lógicas, con $n - 1$ operadores lógicos intercalados entre ellas. Cada una de las constantes es True (verdadero) o False (falso). La cantidad de operadores distintos que aparece en la expresión es acotada, y para cada uno de ellos se conoce su tabla de verdad. El resultado de evaluar la expresión, en general depende del orden en que se aplican los operadores. Por ejemplo, si la expresión es True $\vee$ False $\Rightarrow$ True $\wedge$ False, dos formas posibles de evaluarla son

    $$\begin{aligned} \text{True} \vee ((\text{False} \Rightarrow \text{True}) \wedge \text{False}) &= \text{True}\ , \\ (\text{True} \vee \text{False}) \Rightarrow (\text{True} \wedge \text{False}) &= \text{False}\ . \end{aligned}$$

    Diseñar un algoritmo eficiente que decida si existe un orden de aplicación de los operadores que hace que el resultado de evaluar la expresión sea True. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n^3)$.

    SOLUCIÓN: Es muy parecido al que se da de ejemplo en la Teórica, en el cual hay que minimizar la cantidad de operaciones cuando se calcula la productoria de varias matrices. En este caso se pueden definir dos matrices T y F tales que $T[i,j]$ indica si es posible obtener TRUE en la subexpresión que va de la i-ésima a la j-ésima constante, mientras que $F[i,j]$ es análoga para FALSE. Cada posición se calcula en función de las posiciones $[i,k]$ y $[k+1,j]$ de las mismas matrices, utilizando las tablas de verdad de los operadores ($k = i, i+1, \dots j-1$).

NOTA: Propuesto originalmente por Federico Lebrón.

TOMADO: 2012C2P1 13-OCT-2012.

33. Sea s una cadena de caracteres de longitud n. Sea d (diccionario) una lista de p cadenas de caracteres (palabras) distintas entre sí, cada una de las cuales tiene longitud no nula y $O(1)$. Diseñar un algoritmo que

Variante 1: decida si se puede partir s en subcadenas que están en d. Ejemplos:

- $s =$ "`TikZistkeinZeichenprogramm`" se puede partir en subcadenas que están en $d =$ ("`TikZ`", "`Zeichenprogramm`", "`ist`", "`kein`").
- $s =$ "`adorador`" se puede partir en subcadenas que están en $d =$ ("`ador`", "`noseusa`").
- $s =$ "`adorador`" *no* se puede partir en subcadenas que están en $d =$ ("`adora`", "`dorador`").

Variante 2: determine la cantidad de maneras distintas de partir s en subcadenas que están en d. Ejemplos:

- Para $s =$ "`adorador`" y $d =$ ("`ador`", "`noseusa`") el resultado es 1.
- Para $s =$ "`adorador`" y $d =$ ("`adora`", "`dorador`") el resultado es 0.
- Para $s =$ "`abracadabra`" y $d =$ ("`a`", "`abra`", "`bra`", "`cad`", "`cada`") el resultado es 6 ("`a-bra-cad-a-bra`", "`a-bra-cad-abra`", "`a-bra-cada-bra`", "`abra-cad-a-bra`", "`abra-cad-abra`", y "`abra-cada-bra`").
- Para s formada por n repeticiones del caracter "`a`" y $d =$ ("`a`", "`aa`") el resultado es el n-ésimo número de Fibonacci.

El algoritmo debe tener complejidad $O(np)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SUGERENCIA: Resolver para cada prefijo de s.

SOLUCIÓN:

Variante 1: Sea $f(i)$ la solución restringida a las primeras i posiciones de s. La solución es $f(n)$. Se cumple $f(0) = \text{TRUE}$; $f(i) = \text{IF}(\text{existe } t \in d \text{ que es sufijo de } s_1 s_2 \cdots s_i; f(i-|t|); \text{FALSE})$ para $i = 2, 3, \dots n$. La existencia de t puede decidirse en $O(p)$ verificando cada palabra de d, por lo que la complejidad total es $O(np)$. También es posible decidir la existencia de t en $O(1)$ si al comienzo del algoritmo se almacenan las palabras invertidas en un trie; en tal caso la complejidad total es $O(p+n)$.

Variante 2: Sea $f(i)$ la solución restringida a las primeras i posiciones de s. La solución es $f(n)$. Se cumple $f(0) = 1$; $f(i) = \sum_{t \in d} \text{IF}(t \text{ es sufijo de } s_1 s_2 \cdots s_i; f(i-|t|); 0)$ para $i = 2, 3, \dots n$. La complejidad de $f(i)$ es $O(p)$ ya que las palabras tienen longitud $O(1)$, por lo que la complejidad total es $O(np)$.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2019C1P1 08-MAY-2019 (Variante 1), 2019C1R1 12-JUL-2019 (Variante 2).

34. Dadas dos cadenas de caracteres $s = (s[1], s[2], \dots s[p])$ y $t = (t[1], t[2], \dots t[q])$, se dice que t es una subsecuencia de s si y sólo si existen q valores $i_1 < i_2 \cdots < i_q$ tales que $t[j] = s[i_j]$ para $j = 1, 2, \dots q$. Por ejemplo, "`hoyeselfindelmundo`" tiene como subsecuencias a "`oye`", "`feudo`" y "", pero no a "`nifles`".

Un palíndromo es una cadena de caracteres capicúa, como por ejemplo "`rallar`", "`SMS`", "`@`" y "".

Sea s una cadena de caracteres de longitud n. Diseñar un algoritmo eficiente que indique la longitud de la subsecuencia más larga de s que sea un palíndromo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n^2)$, y está basado en programación dinámica.

SUGERENCIA: Considerar los palíndromos en la porción de la cadena $(s[i], s[i+1], \dots s[j])$.

SOLUCIÓN: Definir $f(i,j)$ como la longitud de la subsecuencia más larga que sea un palíndromo en $(s[i], s[i+1], \ldots s[j])$. Queremos $f(1,n)$. Se cumple $f(i,j) = \max\{\mathrm{IF}(s[i] = s[j]; f(i+1, j-1)+2; -\infty), f(i+1,j), f(i,j-1)\}$.

NOTA: Propuesto originalmente por Michel Mizrahi. El Ejercicio 35 pide subcadena en vez de subsecuencia.

TOMADO: 2012C2R1 12-DIC-2012.

35. Una subcadena de una cadena de caracteres es una porción de la cadena formada por posiciones consecutivas de la misma. Por ejemplo, "`abracadabra`" tiene como subcadenas a "`abra`", "`racad`" y "", pero no a "`arba`" ni "`aa`".

    Un palíndromo es una cadena de caracteres capicúa, como por ejemplo "`rallar`", "`SMS`", "`@`" y "".

    Sea s una cadena de caracteres de longitud n. Diseñar un algoritmo de complejidad $O(n^2)$ que indique la longitud de la subcadena más larga de s que sea un palíndromo. Por ejemplo, para las cadenas "`SMS`", "`abcd`" y "`wiki`", los resultados deben ser respectivamente 3, 1 y 3. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n)$, aunque es muy complicado.

    SOLUCIÓN: Basta encontrar todas las subcadenas que son palíndromos y quedarse con la más larga. Puede hacerse en $O(m+n) = O(m) = O(n^2)$ según el Ejercicio 36, donde m es la cantidad de subcadenas no nulas de s que son palíndromos.

    NOTA: El algoritmo $O(n)$ es el de Manacher. El Ejercicio 34 pide subsecuencia en vez de subcadena.

    TOMADO: 2016C2P1 01-OCT-2016.

36. Una subcadena de una cadena de caracteres es una porción de la cadena formada por posiciones consecutivas de la misma. Por ejemplo, "`abracadabra`" tiene como subcadenas a "`abra`", "`racad`" y "", pero no a "`arba`" ni "`aa`".

    Un palíndromo es una cadena de caracteres capicúa, como por ejemplo "`rallar`", "`SMS`", "`@`" y "".

    Sea s una cadena de caracteres de longitud n. Diseñar un algoritmo de complejidad $O(n^2)$ que indique la cantidad de subcadenas no nulas de s que son palíndromos. Por ejemplo, para las cadenas "`SMS`", "`abcd`" y "`wiki`", los resultados deben ser respectivamente 4, 4 y 5. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(m+n) = O(m)$, donde m es la cantidad buscada.

    SOLUCIÓN: Basta pararse en cada posición y en cada interposición de la cadena, y mover dos punteros hacia los extremos mientras los caracteres por los que se va pasando son iguales.

    TOMADO: 2014C2P1 11-OCT-2014.

37. Un palíndromo es una cadena de caracteres capicúa, como por ejemplo "`rallar`", "`SMS`", "`@`" y "".

    Sea s una cadena de caracteres de longitud n. Diseñar un algoritmo de complejidad $O(n^3)$ que indique la mínima cantidad de porciones en las que es necesario dividir a s para que cada porción sea un palíndromo. Por ejemplo, para las cadenas "`SMS`", "`abcd`" y "`wiki`", los resultados deben ser respectivamente 1, 4 y 2. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(m+n) = O(m)$, donde m es la cantidad de porciones no nulas de s que son palíndromos.

    SOLUCIÓN: Sea $f(i,j)$ que indica el valor buscado para la subcadena que va de la posición i a la j. Se cumple $f(i,j) = 1$ si la subcadena es un palíndromo (en particular, si $i = j$ que es el caso base), y $f(i,j) = \min_{i \le k < j}\{f(i,k) + f(k+1,j)\}$ en caso contrario, lo que produce un algoritmo de complejidad $O(n^3)$.

    Se logra mejor complejidad notando que en una división óptima, la primera subcadena (o si se prefiere, la última) es un palíndromo, a la vez que el resto debe estar dividido de manera óptima. Sea EsPalindromo(i,j) que vale 1 si la subcadena que va de la posición i a la j es un palíndromo, y $+\infty$ en caso contrario. Dicha función puede precomputarse en $O(n^2)$ para todos los valores posibles de i y j, ya que encontrar todas las subcadenas no nulas que son palíndromos puede hacerse en $O(m+n) = O(m) = O(n^2)$ según el Ejercicio 36. De este modo podemos definir $f(1,0) = 0$; $f(1,j) = \min_{0 \le k < j}\{f(1,k) + \mathrm{EsPalindromo}(k+1,j)\}$ para $j = 1, 2, \ldots n$. Alternativamente podemos definir $f(n+1,n) = 0$; $f(i,n) = \min_{i \leq k \leq n}\{\text{EsPalindromo}(i,k) + f(k+1,n)\}$ para $i = n, n-1, \ldots 1$. En ambos casos resulta un algoritmo de complejidad $O(n^2)$.

El segundo algoritmo equivale a calcular un camino mínimo (en un DAG) que tiene un nodo por cada interposición de la cadena, y hay un eje de peso 1 entre dos nodos si y sólo si la subcadena entre ellos es un palíndromo. Podemos calcular el camino mínimo desde el extremo izquierdo hacia el derecho (con los ejes dirigidos de izquierda a derecha), o a la inversa (con los ejes dirigidos de derecha a izquierda). El algoritmo implícito por la función recursiva parece ser el del orden topológico para DAGs, de complejidad $O(m+n)$, donde m es la cantidad de ejes. De hecho, puede implementarse tal algoritmo o también puede usarse BFS (ya que todos los ejes tienen el mismo peso), en ambos casos con complejidad $O(m+n)$. Aparentemente el segundo algoritmo de programación dinámica puede adaptarse para que tenga complejidad $O(m+n)$, tratando solamente los pares de i y j para los cuales $\text{EsPalindromo}(i,j) < +\infty$.

NOTA: Propuesto originalmente por Michel Mizrahi.

TOMADO: 2014C1R1 16-JUL-2014.

38. Juan y Pinchame se quieren comunicar de forma segura. Juan "inventó" la siguiente codificación: `A` $\to$ `1`, `B` $\to$ `2`, ... `Z` $\to$ `27`. Para codificar una palabra se concatenan los códigos de cada una de sus letras. Por ejemplo, como los códigos de "`H`", "`O`", "`L`" y "`A`" son respectivamente "`8`", "`16`", "`12`" y "`1`", entonces el código de "`HOLA`" es "`816121`". Pinchame le dijo a Juan que la idea era pésima, no sólo porque esa codificación es muy insegura, sino también porque la decodificación es ambigua, ya que palabras distintas producen el mismo código. En el ejemplo anterior, al decodificar "`816121`" se podría obtener "`HAFABA`", "`HAFAT`", "`HAFLA`", "`HOABA`", "`HOAT`" o efectivamente "`HOLA`". Juan se defendió diciendo que tal ambigüedad no se iba a presentar casi nunca. Entonces Pinchame le exhibió un algoritmo que dada una secuencia de dígitos, indicaba cuántas palabras tienen a esa secuencia como código. Al ver los resultados, Juan se enojó y se fue.

    Diseñar un algoritmo eficiente basado en programación dinámica que dada una secuencia de n dígitos decimales determine la cantidad de palabras que tienen a esa secuencia como código. Ejemplos: para la secuencia "`816121`" el resultado es 6, para "27" es 2, para "28" es 1, y para "30" es 0. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n)$ y espacial $O(1)$ (adicional a los datos).

    SOLUCIÓN: Sea $(d_1, d_2, \ldots d_n)$ la secuencia de dígitos. Sea $\text{EsLetra}(\cdot)$ una función que devuelve 1 si su argumento es código de alguna letra, y 0 en caso contrario; por ejemplo $\text{EsLetra}(\text{“1”}) = \text{EsLetra}(\text{“2”}) = \text{EsLetra}(\text{“27”}) = 1$, mientras que $\text{EsLetra}(\text{“02”}) = \text{EsLetra}(\text{“28”}) = \text{EsLetra}(\text{“30”}) = 0$. Definir $f(i)$ como la solución restringida a las primeras i posiciones de la secuencia. Se cumple $f(0) = 1$; $f(1) = \text{EsLetra}(d_1)$; $f(i) = f(i-1) \times \text{EsLetra}(d_i) + f(i-2) \times \text{EsLetra}(d_{i-1}d_i)$ para $i = 2, 3, \ldots n$. Esto funciona ya que las palabras que tienen a la secuencia como código pueden dividirse de manera disjunta entre aquellas en las que el dígito actual codifica por sí solo una letra, y aquellas en las que el dígito actual codifica una letra en combinación con el dígito anterior.

    NOTA: Propuesto originalmente por Federico Lebrón.

    TOMADO: 2014C1P1 17-MAY-2014.

39. Diseñar un algoritmo que dado $v \in \mathbb{R}^n$, determine la máxima suma de elementos en posiciones consecutivas de v. Ejemplos: para $v = (1, -1, 1, 2, -1, 4, -1)$ el resultado es 6, para $v = (1, -1, 1, 2, -5, 4, -1)$ el resultado es 4, para $v = (-1)$ el resultado es 0. El algoritmo debe tener complejidad $O(n)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n)$ y espacial $O(1)$ (adicional a los datos).

    SUGERENCIA: Calcular para cada posición del vector la máxima suma que termina en esa posición.

    SUGERENCIA: Tratar cada elemento del vector exactamente una vez.

    SOLUCIÓN: El algoritmo es el de Kadane.

    NOTA: Existe un algoritmo basado en dividir y conquistar de complejidad temporal $O(n \log n)$, que los alumnos podrían haber visto en Algo II.

    TOMADO: 2015C1P1 16-MAY-2015.

40. 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.

Variante 1: 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).

Variante 2: En una partida reciente Rolando eligió a sus personajes en orden de manera tal que cada uno dominara al anterior. Eso fue aburrido porque durante el juego siempre le convenía utilizar a su personaje más dominante. Para que el juego sea más interesante, esta vez Rolando desea que dentro del subconjunto de personajes que elija, ninguno domine a ningún otro.

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.

SOLUCIÓN: Ordenar los personajes, digamos por resistencia. Definir $f(i)$ como la solución restringida a los primeros i personajes, y tal que incluye al i-ésimo personaje. Se cumple $f(1) = 1$; $f(i) = 1 + \max_{j \in \text{Compatibles}(i)} f(j)$ para $i = 2, 3, \ldots n$, donde suponemos que el máximo vale 0 si $\text{Compatibles}(i) = \emptyset$. Finalmente, tomar el máximo de todos los $f(i)$.

Para la Variante 1 es $\text{Compatibles}(i) = \{j < i \text{ tales que } s(j) < s(i)\}$. Esto funciona porque los subconjuntos deseables deben estar ordenados de manera creciente por el segundo atributo, y si j es compatible con i entonces todos los personajes considerados por $f(j)$ pueden estar en el mismo subconjunto que el i-ésimo personaje.

Para la Variante 2 es $\text{Compatibles}(i) = \{j < i \text{ tales que } s(j) > s(i)\}$. Esto funciona porque los subconjuntos deseables deben estar ordenados de manera decreciente por el segundo atributo, y si j es compatible con i entonces todos los personajes considerados por $f(j)$ pueden estar en el mismo subconjunto que el i-ésimo personaje.

En definitiva la solución es ordenar por un atributo y buscar la subsecuencia creciente o decreciente más larga (según la variante). El problema de encontrar la subsecuencia es bastante conocido y puede pensarse en términos de grafos, como aparece en el Ejercicio 219.

Ojo que no anda definir $f(i)$ como la solución restringida a los primeros i personajes y suponer $f(i) = \max\{f(i-1), 1 + \max_{j \in \text{Compatibles}(i)} f(j)\}$ para $i = 2, 3, \ldots n$, que representaría las dos opciones posibles de no elegir y elegir al i-ésimo personaje. La respuesta sería directamente $f(n)$, pero no anda porque si $f(j)$ no considera al j-ésimo personaje, entonces la compatiblidad de j no asegura la compatibilidad de los personajes considerados por $f(j)$.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez (Variante 2).

TOMADO: 2015C2P1 03-OCT-2015 (Variante 1), 2015C2R1 11-DIC-2015 (Variante 2).

41. Sea $v = (v_1, v_2, \ldots v_n)$ un vector de números enteros. Diseñar un algoritmo que indique la mínima cantidad de números que hay que eliminar del vector para que cada número que permanezca sea múltiplo del anterior (excepto el primero). Por ejemplo, para los vectores $(-5, 5, 0)$, $(0, 5, -5)$ y $(0, 5, -5, 2, 15, 15)$, los resultados deben ser respectivamente 0, 1 y 2. El algoritmo debe tener complejidad $O(n^2)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar.

SOLUCIÓN: Es casi igual al Ejercicio 219.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2017C1P1 13-MAY-2017.

42. Astro Boy ha crecido, y ha cambiado su nombre a Astro Void. Para ganarse la vida se dedica a la compraventa de asteroides. Sea $p \in \mathbb{N}^n$ tal que p_i es el precio de un asteroide el i-ésimo día en una secuencia de n días. Astro Void quiere comprar y vender asteroides durante esos n días de manera tal de obtener la mayor ganancia neta posible. Debido a las dificultades que existen en el almacenaje y transporte de asteroides, Astro Void comienza sin asteroides, puede comprar a lo sumo un asteroide cada día, y puede vender a lo sumo un asteroide cada día.

Variante 1: Además, el Ente Regulador Asteroidal prohíbe que Astro Void venda un asteroide antes de haberlo comprado, o que al final de los n días tenga algún asteroide.

Variante 2: El Ente Regulador Asteroidal permite que Astro Void venda un asteroide antes de haberlo comprado; sin embargo, ningún día la cantidad de asteroides vendidos pero no comprados puede exceder a 3, y al final de los n días todos los asteroides vendidos deben haber sido comprados.

Diseñar un algoritmo eficiente basado en programación dinámica que calcule la máxima ganancia neta que puede obtener Astro Void respetando las restricciones indicadas. Por ejemplo, para los vectores $p = (2, 2, 2, 1, 2, 1, 1, 1)$, $p = (2, 2, 2, 2, 1, 1, 1, 1)$ y $p = (2, 7, 1, 8, 2)$, los resultados deben ser respectivamente

Variante 1: 1, 0 y 12.

Variante 2: 4, 3 y 12.

Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n^2)$ y espacial $O(n)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

SOLUCIÓN:

Variante 1: Armar un digrafo con un nodo por cada estado del problema, y un arco desde un estado a otro si y sólo si es posible pasar del primero al segundo con la compra o venta de asteroides. Los estados son pares ordenados (c, d) representando que Astro Void tiene c asteroides al terminar el día d, para $c = 0, 1, \ldots n$ y $d = 0, 1, \ldots n$. Hay un arco de longitud $-p_d$ desde el estado $(c-1, d-1)$ a (c, d) representando la compra de un asteroide durante el día d. Hay un arco de longitud p_d desde el estado $(c+1, d-1)$ a (c, d) representando la venta de un asteroide durante el día d. Hay un arco de longitud 0 desde el estado $(c, d-1)$ a (c, d) representando la compra y venta de un asteroide durante el día d. Lo que queremos es la longitud de un camino máximo del estado $(0, 0)$ al estado $(0, n)$. Los nodos tienen grado $O(1)$, de modo que la cantidad de ejes es lineal en la cantidad de nodos, y por lo tanto el algoritmo de camino máximo en DAGs tiene complejidad lineal en la cantidad de nodos, que es $O(n^2)$ en este caso. El algoritmo se puede ejecutar sin construir explícitamente el grafo, pero pensado en estos términos se requiere cierto cuidado para obtener la complejidad espacial indicada.

Una formulación equivalente basada en programación dinámica consiste en definir $f(c, d)$ como la máxima ganancia neta terminando con c asteroides el día d. Queremos $f(0, n)$. Se cumple $f(0, 0) = 0$, $f(c, 0) = -\infty$ para $c = 1, 2, \ldots n$; $f(c, d) = \max\{f(c, d-1), \mathrm{IF}(c > 0; f(c-1, d-1)-p_d; -\infty), \mathrm{IF}(c < n; f(c+1, d-1)+p_d; -\infty)\}$ para $c = 0, 1, \ldots n$ y $d = 1, 2, \ldots n$. Se puede calcular en una matriz por columnas, o manteniendo sólo dos columnas a la vez, lo que permite obtener las complejidades requeridas.

Notar que la solución óptima no utiliza los estados con más de $n/2$ asteroides, ya que no es posible comprarlos cuando pasaron pocos días, y no es posible venderlos cuando faltan pocos días para terminar. En general, c debe mantenerse entre d y $n - d$, lo cual podría utilizarse para calcular menos estados, aunque sin mejorar la complejidad.

Variante 2: Es similar a la Variante 1. Las diferencias son que la cantidad de asteroides toma valores $c = -3, -2, \ldots n$, y que queremos calcular $\max_{c \geq 0} f(c, n)$. No obstante, es suficiente con calcular $f(0, n)$, ya que nunca conviene terminar con algún asteroide debido a que los precios no son negativos.

Otras: Se consideró una variante similar a la Variante 2, pero sin límite para la cantidad de asteroides vendidos pero no comprados. En tal caso el orden original de los precios se vuelve irrelevante, y es más un conjunto de precios que una lista de precios. La ganacia óptima puede obtenerse si se ordenan los precios para tomar los $\lfloor n/2 \rfloor$ precios más baratos y los $\lfloor n/2 \rfloor$ precios más caros, restando el primer grupo del segundo. Tal solución tiene complejidad $O(n \log n)$, y no requiere programación dinámica sino que es golosa.

Se consideró una variante con capital, el cual aumentaba con las ventas y disminuía con las compras, pero pareció muy complicada.

Se consideró una variante en la que podía comprarse y venderse cualquier cantidad de asteroides cada día mientras nunca se tuvieran más de n. En tal caso la solución óptima no compra y vende el mismo día (sería equivalente a comprar o vender el neto), no compra dos días distintos sin vender entre ellos (es mejor o igual comprar al precio mínimo de los dos días), no vende dos días distintos sin comprar entre ellos (es mejor o igual vender al precio máximo de los dos días), cuando compra lleva el stock al máximo posible (para ganar más en la siguiente venta), y cuando vende lleva el stock al mínimo posible (análogo). De tal modo para cada día hay sólo dos estados posibles (stock máximo o sin stock), y existe una solución que compra al inicio de cada subsecuencia ascendente y vende al final, siempre cantidades máximas. Dicha solución es golosa y tiene complejidad $O(n)$.

NOTA: Propuesto originalmente por Guido Tagliavini Ponce.

TOMADO: 2016C1P1 07-MAY-2016 (Variante 1), 2016C1R1 11-JUL-2016 (Variante 2).

43. En una materia de la facultad están desarrollando un nuevo videojuego para enseñar programación dinámica. El videojuego es una adaptación del clásico Strikers 1945, que por algún motivo se va a llamar Dijkstrers 1930. Cada nivel del juego está dado por una cantidad de disparos disponibles y una matriz. El jugador maneja un avión propio que inicialmente se ubica en la posición inferior izquierda de la matriz. En cada una de las otras posiciones puede haber un avión enemigo.

Variante 1: Cada segundo el avión propio se mueve una fila hacia arriba, y el jugador puede elegir que se mantenga en la misma columna, o que simultáneamente se mueva una columna a la izquierda o a la derecha, siempre dentro de los límites de la matriz. Si el avión propio y un avión enemigo ocupan la misma posición, el jugador puede destruir al avión enemigo efectuando un disparo; caso contrario los dos aviones colisionan y se pierde el nivel. Para completar el nivel, el avión propio debe llegar a la fila superior sin haber colisionado. El objetivo es completar el nivel habiendo destruído la máxima cantidad posible de aviones enemigos con los disparos disponibles. Diseñar un algoritmo eficiente que decida si el nivel puede ser completado y en tal caso indique esa cantidad. La entrada del algoritmo es la cantidad $d \geq 1$ de disparos disponibles y la matriz de f filas y c columnas. El algoritmo debe tener complejidad $O(dfc)$. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. En los siguientes ejemplos el avión propio se representa con $\triangle$ y cada avión enemigo con $\bigtriangledown$.

$d = 1$

$\bigtriangledown$	$\bigtriangledown$	$\bigtriangledown$
$\bigtriangledown$	$\bigtriangledown$	$\bigtriangledown$
$\triangle$	$\bigtriangledown$	$\bigtriangledown$

imposible

$d = 1$

$\bigtriangledown$	$\bigtriangledown$	
$\triangle$	$\bigtriangledown$	

max $= 1$

$d = 1$

	$\bigtriangledown$	
	$\bigtriangledown$	
$\triangle$		

max $= 1$

$d = 314$

		$\bigtriangledown$
	$\bigtriangledown$	
$\triangle$		

max $= 2$

$d = 271$

	$\bigtriangledown$	
		$\bigtriangledown$
$\triangle$		

max $= 1$

SUGERENCIA: Definir $g(i, j, k)$ como la máxima cantidad de aviones enemigos destruídos cuando el avión propio está en la posición (i, j) de la matriz y le quedan k disparos disponibles, o $-\infty$ si eso no es posible.

Variante 2: Cada segundo cada avión enemigo se mueve una fila hacia abajo, y el jugador puede elegir que el avión propio se mantenga quieto, o que simultáneamente se mueva una columna a la izquierda o a la derecha, siempre dentro de los límites de la matriz. Si el avión propio y un avión enemigo ocupan la misma posición, el jugador puede destruir al avión enemigo efectuando un disparo, aunque no está obligado a hacerlo. Si un avión enemigo llega a la fila inferior, en el paso siguiente desaparece del juego. Para completar el nivel, todos los aviones enemigos deben haber desaparecido del juego. El objetivo es completar el nivel habiendo destruído la máxima cantidad posible de aviones enemigos con los disparos disponibles. Diseñar un algoritmo eficiente que indique esa cantidad. La entrada del algoritmo es la cantidad $d \geq 1$ de disparos disponibles y la matriz de f filas y c columnas. El algoritmo debe tener complejidad $O(dfc)$. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(fc)$, lo cual es necesario para obtener puntaje máximo en este ejercicio. En los siguientes ejemplos el avión propio se representa con $\triangle$ y cada avión enemigo con $\triangledown$.

$d = 1$

$\triangledown$	$\triangledown$	$\triangledown$
$\triangledown$	$\triangledown$	$\triangledown$
$\triangle$	$\triangledown$	$\triangledown$

$\max = 1$

$d = 314$

		$\triangledown$
	$\triangledown$	
$\triangle$		

$\max = 2$

$d = 271$

	$\triangledown$	
		$\triangledown$
$\triangle$		

$\max = 1$

SOLUCIÓN:

Variante 1: Numerar las filas de la matriz desde 1 hasta f, y las columnas desde 1 hasta c, empezando (por ejemplo) por la posición inicial del avión propio. Armar un digrafo con un nodo por cada estado del juego, y un arco de un estado a otro si y sólo si es posible pasar del primero al segundo conforme avanza el juego. Los estados son ternas (i, j, k) representando que el avión propio está en la posición (i, j) y le quedan k disparos disponibles, para $i = 1, 2, \ldots f$, $j = 1, 2, \ldots c$ y $k = 0, 1, \ldots d$. El estado inicial es el $(1, 1, d)$, mientras que los estados finales son los que cumplen $i = f$. Si un arco llega a un estado que indica una posición en la cual hay un avión enemigo, el arco tiene longitud 1, y caso contrario tiene longitud 0. Lo que queremos es la longitud de un camino máximo del estado inicial a un estado final, ya que la longitud de un camino entre esos estados es la cantidad de aviones enemigos destruídos. Basta calcular la longitud de un camino máximo a cada estado final, y quedarse con el máximo sobre todos los estados finales (o se pueden agrupar todos los estados finales agregando un nodo final ficticio). Los nodos tienen grado $O(1)$, de modo que la cantidad de arcos es lineal en la cantidad de nodos, y por lo tanto el algoritmo de camino máximo en DAGs tiene complejidad lineal en la cantidad de nodos, que es $O(dfc)$ en este caso. El algoritmo se puede ejecutar sin construir explícitamente el grafo.

Una formulación equivalente basada en programación dinámica consiste en definir la función que aparece en la sugerencia y calcular su máximo para $i = f$. Se puede calcular en una matriz tridimensional, o con dos matrices comunes manteniendo sólo dos valores de i a la vez.

Variante 2: Es equivalente que los aviones enemigos bajen y desaparezcan por la parte inferior de la matriz, a que el avión propio suba y salga por la parte superior. Esto último es equivalente a que el avión propio llegue a la fila superior, ya que no hay más aviones enemigos que enfrentar a partir de ese momento. Por lo tanto, respecto de los movimientos de los aviones, podemos suponer que estamos en la situacíon de la Variante 1. Lo único diferente es entonces que no se pierde el nivel si el avión propio y un avión enemigo ocupan la misma posición, por lo que lo ideal es que el avión propio pase por la máxima cantidad posible de posiciones con aviones enemigos, destruyendo a todos los que pueda con los disparos disponibles. En consecuencia, es suficiente con determinar la máxima cantidad de posiciones con aviones enemigos que un recorrido del avión propio puede tener, y quedarse con el mínimo entre ese valor y d. Para ello armar un grafo con un nodo por cada posición de la matriz, y un arco de un nodo a otro si y sólo si es posible pasar de una posición a la otra. El nodo inicial es la posición inicial del avión propio, mientras que los nodos finales son las posiciones de la fila superior. Si un arco llega a un nodo que indica una posición en la cual hay un avión enemigo, el arco tiene longitud 1, y caso contrario tiene longitud 0. Lo que queremos es la longitud de un camino máximo del nodo inicial a un nodo final, ya que la longitud de un camino entre esos nodos es la cantidad de aviones enemigos que el camino tiene. El resto es similar a la Variante 1. El algoritmo de camino máximo en DAGs tiene complejidad $O(fc)$ en este caso, y luego calcular el mínimo respecto de d es $O(1)$.

Para calcular el máximo se puede plantear una formulación equivalente basada en programación dinámica. Consiste en definir $g(i, j)$ como la máxima cantidad de posiciones con aviones enemigos en los recorridos del avión propio que van de su posición inicial a la posición

(i, j). Se puede calcular en una matriz o con dos vectores que mantienen sólo dos filas de la misma.

NOTA: Propuesto originalmente por Brian Bokser (Variante 1).

TOMADO: 2017C2P1 04-OCT-2017 (Variante 1), 2017C2R1 11-DIC-2017 (Variante 2).

44. Investring es un juego para un jugador. Al inicio del juego el jugador elige dos cadenas de n caracteres $s = s_1 s_2 \cdots s_n$ y $t = t_1 t_2 \cdots t_n$. El juego se desarrolla en n pasos. En el i-ésimo paso ($i = 1, 2, \ldots n$) el jugador elige entre invertir o no la subcadena $s_i s_{i+1} \cdots s_n$. Por ejemplo, si $s =$ "`esa`", luego del primer paso se tendrá $s =$ "`ase`" si el jugador elige invertir, y luego del segundo paso se tendrá $s =$ "`aes`" si el jugador nuevamente elige invertir. El objetivo del juego es que al terminar los n pasos se cumpla $s = t$. Diseñar un algoritmo de complejidad $O(n^2)$ que indique la mínima cantidad de inversiones necesarias para ganar el juego. Si es imposible ganar, el algoritmo debe informarlo. Ejemplos:

   - Para $s =$ "`esa`" y $t =$ "`aes`", el resultado es 2.
   - Para $s =$ "`esa`" y $t =$ "`eas`", el resultado es 1.
   - Para $s =$ "`esa`" y $t =$ "`sea`", es imposible ganar el juego.

   Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

   SUGERENCIA: Definir $f(i, j)$ como la mínima cantidad de inversiones necesarias para ganar el juego si las cadenas iniciales son $s' = s_i s_{i+1} \cdots s_j$ y $t' =$ últimos $j - i + 1$ caracteres de t (o infinito si es imposible ganar con esas cadenas iniciales). Análogamente definir $g(i, j)$ para las cadenas iniciales $s' = s_j s_{j-1} \cdots s_i$ y $t' =$ últimos $j - i + 1$ caracteres de t.

   SOLUCIÓN: La solución utiliza programación dinámica. El enunciado pide calcular $f(1, n)$, donde el valor infinito significa que es imposible ganar. Sea $d(k, \ell)$ que vale 1 si s_k coincide con la primera posición de $t' =$ últimos ℓ caracteres de t, e infinito en caso contrario. Se cumple $f(i, i) = g(i, i) = d(i, 1)$ para $i = 1, 2, \ldots n$; $f(i, j) = \min\{f(i+1, j) \times d(i, j-i+1), [1 + g(i, j-1)] \times d(j, j-i+1)\}$ y $g(i, j) = \min\{g(i, j-1) \times d(j, j-i+1), [1 + f(i+1, j)] \times d(i, j-i+1)\}$ para $i = 1, 2, \ldots n-1$ y $j = i+1, i+2, \ldots n$.

   NOTA: Propuesto originalmente por Guido Tagliavini Ponce.

   Similar al Problema G del Torneo Argentino de Programación 2015.

   http://torneoprogramacion.com.ar/wp-content/uploads/2015/09/pruebaTAP2015.pdf

   http://www.spoj.com/problems/TAP2015G/

   TOMADO: 2016C2R1 12-DIC-2016.

45. Viajante de comercio bitónico: Se tienen $n \geq 2$ puntos en el plano $p_1, p_2, \ldots p_n$, ordenados de manera creciente de acuerdo a su coordenada x (no hay dos puntos con la misma coordenada x). Un recorrido bitónico de los puntos es un recorrido que comienza en p_1, recorre de manera creciente en x algunos de los puntos hasta llegar a p_n (ida), y finalmente recorre de manera decreciente en x algunos de los puntos hasta volver a p_1 (vuelta); el recorrido pasa exactamente una vez por cada punto.

   Para $n = 2$ hay un único recorrido bitónico, que es p_1, p_2, p_1. Para $n = 3$ hay dos recorridos bitónicos. Uno es p_1, p_2, p_3, p_1, mientras que el otro es p_1, p_3, p_2, p_1. Sin embargo, ambos recorridos tienen la misma longitud, ya que lo que recorre uno a la ida, lo recorre el otro a la vuelta, y viceversa. En general, dado cualquier recorrido bitónico, existe uno "simétrico" con la ida y la vuelta intercambiadas.

   Diseñar un algoritmo que encuentre la mínima longitud que puede tener un recorrido bitónico. El algoritmo debe tener complejidad $O(n^2)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n^2)$ y espacial $O(n)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

   SUGERENCIA: Para $i \neq j$, sea $f(i, j) = f(j, i)$ la longitud mínima para completar hasta p_1 un recorrido bitónico a partir de una porción que va de p_i a p_n, vuelve de p_n a p_j, y pasa exactamente una vez por cada punto $p_{\min\{i,j\}}, p_{\min\{i,j\}+1}, \ldots p_n$. La solución es $f(n-1, n) + d(n-1, n)$, donde $d(i, j) = d(j, i)$ es la distancia (en el plano) entre p_i y p_j.

SUGERENCIA: Para $i \neq j$, sea $g(i,j) = g(j,i)$ la longitud mínima para completar hasta p_n un recorrido bitónico a partir de una porción que va de p_1 a p_i, vuelve de p_j a p_1, y pasa exactamente una vez por cada punto $p_1, p_2, \ldots p_{\max\{i,j\}}$. La solución es $g(1,2) + d(1,2)$, donde $d(i,j) = d(j,i)$ es la distancia (en el plano) entre p_i y p_j.

SOLUCIÓN: Para la primera sugerencia sea $m = \min\{i,j\}$. En la parte óptima que completa la porción existente, p_{m-1} debe aparecer inmediatamente a la izquierda de alguna de las dos ramas, y el resto de esa parte óptima debe ser también óptimo. De tal forma $f(i,j)$ es $f(m-1,j)+d(m-1,i)$ si p_{m-1} aparece inmediatamente a la izquierda de p_i, o $f(i,m-1) + d(m-1,j)$ si p_{m-1} aparece inmediatamente a la izquierda de p_j. Como no sabemos cuál opción es mejor, tomamos el mínimo de ambas, es decir $f(i,j) = \min\{f(m-1,j) + d(m-1,i), f(i,m-1) + d(m-1,j)\}$. Dado que $f(i,j) = f(j,i)$ es suficiente calcular la función para $i < j$. En tal caso resulta $m = i$ y por lo tanto $f(i,j) = \min\{f(i-1,j) + d(i-1,i), f(i-1,i) + d(i-1,j)\}$. Los casos base son aquellos con $i = 1$; esos valores son sencillos de calcular ya que $f(1,j) = d(1,j)$. Si usamos una matriz para almacenar los valores de la función, estamos calculando la parte estrictamente triangular superior de la misma. Teniendo en cuenta las dependencias es posible completar la matriz por filas, manteniendo sólo dos filas en memoria a la vez, en $O(n^2)$ tiempo y $O(n)$ espacio. La solución es la indicada en la sugerencia, ya que podemos suponer sin pérdida de generalidad que p_{n-1} aparece inmediatamente a la izquierda de p_n en la ida, y el costo de esa porción a completar es $d(n-1,n)$.

De manera similar, para la segunda sugerencia sea $m = \max\{i,j\}$. En la parte óptima que completa la porción existente, p_{m+1} debe aparecer inmediatamente a la derecha de alguna de las dos ramas, y el resto de esa parte óptima debe ser también óptimo. De tal forma $g(i,j) = \min\{g(m+1,j) + d(i,m+1), g(i,m+1)+d(j,m+1)\}$. Como es suficiente calcular la función para $i < j$ resulta $m = j$ y por lo tanto $g(i,j) = \min\{g(j,j+1) + d(i,j+1), g(i,j+1) + d(j,j+1)\}$. Los casos base son $g(i,n) = d(i,n)$. Podemos almacenar los valores de la función en la parte estrictamente triangular superior de una matriz, y completar la misma por columnas de derecha a izquierda manteniendo sólo dos columnas en memoria a la vez. La solución es la indicada en la sugerencia, ya que podemos suponer sin pérdida de generalidad que p_2 aparece inmediatamente a la derecha de p_1 en la vuelta, y el costo de esa porción a completar es $d(1,2)$.

NOTA: Propuesto originalmente por Gonzalo Ciruelos. Aparece en el libro de Cormen.

TOMADO: 2018C1P1 09-MAY-2018, 2018C1R1 13-JUL-2018.

46. Diseñar un algoritmo que dado $v \in \mathbb{N}^n$ con $n \geq 3$, determine la máxima suma que puede obtenerse eligiendo algunos de esos n valores, con la condición de que no se pueden elegir 3 valores en posiciones consecutivas. Ejemplos: para $v = (1,6,1,8)$ el resultado es 15, para $v = (2,7,1,8,2,8)$ el resultado es 25, para $v = (3,14,15,9,26)$ el resultado es 55. El algoritmo debe tener complejidad $O(n)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n)$ y espacial $O(1)$ (adicional a los datos).

    SUGERENCIA: Definir $f(i)$ como la solución restringida a las primeras i posiciones de v.

    SOLUCIÓN: $f(1) = v_1$; $f(2) = v_1 + v_2$; $f(3) = \max\{v_1 + v_2, v_1 + v_3, v_2 + v_3\}$ (alguno de los 3 no lo podemos elegir); $f(i) = \max\{f(i-1), v_i + f(i-2), v_i + v_{i-1} + f(i-3)\}$ para $i = 4,5,\ldots n$ (no elegimos v_i, elegimos v_i pero no v_{i-1}, elegimos ambos). Si los valores pudieran ser negativos, hay que cambiar levemente los casos base.

    NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

    TOMADO: 2018C2P1 03-OCT-2018.

Grafos conexos, bipartitos, generalidades, etc.

47. Sean d_{out} y d_{in} dos vectores, cada uno de ellos formado por n enteros

Variante 1: en el intervalo $[0, n-1]$.

Variante 2: no negativos.

Además, ambos vectores tienen igual suma. Se desea encontrar un digrafo tal que el i-ésimo vértice tenga grado de salida $d_{\text{out}}(i)$ y grado de entrada $d_{\text{in}}(i)$ para $i = 1, 2, \ldots n$.

Variante 1: El digrafo buscado no debe tener arcos de un vértice a sí mismo, ni más de un arco con igual dirección entre el mismo par de vértices (aunque sí puede tener dos arcos con direcciones opuestas).

Variante 2: El digrafo buscado puede tener arcos de un vértice a sí mismo, y varios arcos con igual o distinta dirección entre el mismo par de vértices.

¿Es cierto que siempre existe un digrafo como el descripto? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

Variante 1: La cota superior es inútil ya que siempre puede ser burlada agregando una cantidad suficiente de elementos nulos en ambos vectores a la vez, los cuales necesariamente van a producir nodos aislados. Con eso en mente es fácil conseguir un contraejemplo que permite afirmar que no siempre existe un digrafo simple. Por ejemplo, basta tomar $d_{\text{out}} = d_{\text{in}} = (0, 1)$.

Variante 2: Siempre existe. Se puede demostrar por inducción en la suma de d_{out} (o de d_{in}). En el caso base el grafo es $K_n^c = nK_1$. En el paso inductivo restamos 1 a cualquier $d_{\text{out}}(i) > 0$ y cualquier $d_{\text{in}}(j) > 0$, armamos un grafo para los vectores modificados, y agregamos el arco (v_i, v_j).

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2012C1R1 18-JUL-2012 (Variante 1).

48. (a) Sea G un grafo. Demostrar que G tiene un subgrafo que es un ciclo simple impar si y sólo si tiene un subgrafo inducido que es un ciclo simple impar.

    (b) ¿Sigue valiendo la propiedad del punto anterior para ciclos simples pares en vez de impares? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

    SOLUCIÓN:

    (a) La ida se demuestra utilizando las mismas ideas que en el Ejercicio 62 y el Ejercicio 314.
    La vuelta es trivial.

    (b) La vuelta es cierta pero la ida no. Un posible contraejemplo es $K_4 - e$ (diamante), que tiene un ciclo simple par, pero no inducido. Hay que elegir los cuatro nodos para poder tener un ciclo simple par, pero si se los elige no inducen un ciclo simple.

    NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

    TOMADO: 2018C2R1 12-DIC-2018.

49. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

    (a) Demostrar que para todo $d \in \mathbb{N}_0$ existe un grafo d-regular.
    (b) Sea G un grafo de n vértices. Demostrar que G es d-regular si y sólo si G^c es $(n-1-d)$-regular.
    (c) Demostrar que un grafo es 0-regular si y sólo si cada una de sus componentes conexas es K_1.
    (d) Demostrar que un grafo es 1-regular si y sólo si cada una de sus componentes conexas es K_2.
    (e) Demostrar que un grafo es 2-regular si y sólo si cada una de sus componentes conexas es un circuito simple.

    SOLUCIÓN:

    (a) K_{d+1}.
    (b) Para la ida, basta notar que en G^c cada vértice es adyacente a los vértices a los cuales no es adyacente en G, con excepción de sí mismo. La cantidad de esos vértices es $n - 1 - d$, que por definición es el grado del vértice en G^c.
    La vuelta se deduce de la ida ya que $(G^c)^c = G$ y $n - 1 - (n - 1 - d) = d$.
    (c) Fácil.
    (d) Fácil.

(e) Para la ida, basta demostrar que cada componente conexa del grafo es un circuito simple. Consideremos entonces una componente conexa del grafo, y sea C un camino simple máximo (y por lo tanto maximal) en ella. El camino C debe tener al menos 3 vértices porque caso contrario no habría un nodo de grado 2 en el grafo. Sean v y w los extremos de C. Estos vértices no pueden tener adyacentes fuera de C porque en tal caso el camino no sería maximal. Tampoco pueden tener adyacencias a vértices internos de C (salvo las de C mismo) porque en tal caso esos vértices internos no tendrían grado 2. Sin embargo, v y w tienen grado 2, de modo que hay un eje entre ellos, que junto con C forma un circuito simple. Los nodos del circuito no tienen adyacentes fuera del mismo porque en tal caso no tendrían grado 2.

Para la vuelta, cada circuito aporta 2 ejes a cada nodo del mismo. Como no hay nodos ni ejes fuera de los circuitos, el grado de cada nodo es 2.

TOMADO: 2015C1P1 16-MAY-2015 (dos últimos puntos), 2018C2P1 03-OCT-2018 (tres primeros puntos).

50. Sea G un grafo de m ejes.

(a) Demostrar que si $m < 10$ entonces existe un vértice v tal que $d(v) < 4$.

(b) Demostrar que si $m < 15$ entonces existe un vértice v tal que $d(v) < 5$.

(c) Demostrar que para todo $d \in \mathbb{N}$, si $m < d \times (d+1)/2$ entonces existe un vértice v tal que $d(v) < d$.

SOLUCIÓN: Basta demostrar el tercer punto, ya que los otros son los casos particulares $d = 4$ y $d = 5$. Sea n la cantidad de vértices. Por absurdo suponemos que para todo vértice $d(v) \geq d$. Si tomamos cualquier vértice v se cumple $n \geq d(v)+1 \geq d+1$. Además $2m = \sum_{i=1}^{n} d(v_i) \geq \sum_{i=1}^{n} d = nd \geq d(d+1)$, y por lo tanto $m \geq d(d+1)/2$, lo cual es absurdo.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2014C1P1 17-MAY-2014 (dos primeros puntos), 2018C1P1 09-MAY-2018 (último punto).

51. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, se define su grafo junta como $G_1 + G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$, es decir, el grafo que contiene a G_1 y a G_2 como subgrafos, y además contiene un eje entre cada vértice de G_1 y cada vértice de G_2. Se dice que un grafo J es un grafo junta si y sólo si existen grafos G_1 y G_2 tales que $J = G_1 + G_2$.

Sea G un grafo.

(a) Demostrar que si G es un grafo junta entonces G es conexo. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

(b) Demostrar que si G es un grafo junta entonces G^c es no conexo. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

(c) ¿Puede ocurrir que G y G^c sean ambos grafos junta? En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar.

SUGERENCIA: Usar que si un grafo no es conexo entonces su complemento lo es.

SOLUCIÓN:

(a) Sean $v_1 \in V_1$ y $v_2 \in V_2$. Sean v y w dos vértices de G. Si $v, w \in V_1$, entonces existe el camino v, v_2, w. Si $v, w \in V_2$, entonces existe el camino v, v_1, w. Si uno pertenece a V_1 y el otro a V_2, entonces existe el camino v, w. Como en todos los casos existe un camino entre v y w, G es conexo.

Alternativamente, por el segundo punto G^c es no conexo, de modo que G es conexo según la sugerencia.

La recíproca es falsa, por ejemplo para K_1. Otros ejemplos pueden encontrarse en el Ejercicio 52.

(b) Por el Ejercicio 5.6 de la Práctica, basta con encontrar una partición de los vértices de G^c para la cual no haya ejes entre las partes. Una posible partición es (V_1, V_2). Es una partición porque cumple la definición de tal. Si tomamos cualquier par de vértices $v_1 \in V_1$ y $v_2 \in V_2$, el eje (v_1, v_2) no está en G^c porque sí está en G.

La recíproca es cierta. Por el Ejercicio 5.6 de la Práctica, existe una partición (V_1, V_2) de los vértices de G^c para la cual no hay ejes entre las partes. Si tomamos cualquier par de vértices $v_1 \in V_1$ y $v_2 \in V_2$, el eje (v_1, v_2) está en G porque no está en G^c. Eso equivale a decir que G es el grafo junta del subgrafo inducido por V_1 y del subgrafo inducido por V_2.

(c) No puede ocurrir. Si G y G^c fueran grafos junta, por el segundo punto G^c y G serían no conexos, lo cual no puede ocurrir según la sugerencia.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

TOMADO: 2017C2R1 11-DIC-2017 (las idas), 2018C1R1 13-JUL-2018 (segundo punto).

52. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, se define su grafo junta como $G_1 + G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$, es decir, el grafo que contiene a G_1 y a G_2 como subgrafos, y además contiene un eje entre cada vértice de G_1 y cada vértice de G_2. Se dice que un grafo J es un grafo junta si y sólo si existen grafos G_1 y G_2 tales que $J = G_1 + G_2$.

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

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) $K_{p,q}$

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

SOLUCIÓN: Por definición, el grafo trivial no es grafo junta, de modo que en lo que sigue sólo se analizan grafos no triviales.

K_n: Para $n \geq 2$ es grafo junta ya que $K_n = K_{n-1} + K_1$.

C_n: Por el primer punto $C_3 = K_3$ es grafo junta. Por el cuarto punto $C_4 = K_{2,2}$ es grafo junta. Para $n \geq 5$ no es grafo junta; ya que si lo fuera, como no hay vértices universales, G_1 y G_2 deberían tener al menos dos nodos, y entonces su junta tendría a C_4 como subgrafo.

Alternativamente para $n \geq 5$ no es grafo junta. Si en G_1 o en G_2 hay al menos 3 vértices, entonces en $G_1 + G_2$ habría vértices de grado al menos 3, lo cual no ocurre en C_n. Por lo tanto G_1 y G_2 tienen a lo sumo 2 vértices, lo que implica $n \leq 4$.

P_n: Por el primer punto $P_2 = K_2$ es grafo junta. Por el punto siguiente $P_3 = K_{1,2}$ es grafo junta. Para $n \geq 4$ no es grafo junta, por el mismo argumento del punto anterior.

Alternativamente para $n \geq 4$ no es grafo junta. El mismo argumento alternativo presentado para C_n permite demostrar para P_n que G_1 y G_2 tienen a lo sumo 2 vértices. Pero si los dos tuvieran 2 vértices, entonces en $G_1 + G_2$ no habría vértices de grado menor que 2, que están presentes en P_n. Por lo tanto G_1 o G_2 tienen un vértice, y dado que ya vimos que G_1 y G_2 tienen a lo sumo 2 vértices, necesariamente $n \leq 3$.

$K_{p,q}$: Todos, ya que $K_{p,q} = pK_1 + qK_1$.

ABC h: Para $h = 1$ el árbol es $K_{1,2}$ que por el punto anterior es grafo junta. Para $h \geq 2$ no es grafo junta por el mismo argumento utilizado para C_n.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. No se incluye A n (variable), GBC n m (variable), ni W_n (trivial). De acuerdo al Ejercicio 51, las respuestas a este ejercicio son las complementarias de las respuestas al Ejercicio 93.

TOMADO: 2018C1P1 09-MAY-2018.

53.

Variante 1: Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, se define su grafo junta como $G_1 + G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$, es decir, el grafo que contiene a G_1 y a G_2 como subgrafos, y además contiene un eje entre cada vértice de G_1 y cada vértice de G_2. Se dice que un grafo J es un grafo junta si y sólo si existen grafos G_1 y G_2 tales que $J = G_1 + G_2$.

Sea G un grafo de n vértices y m ejes. Diseñar un algoritmo eficiente que decida si $G = G_1 + G_2$ con G_1 bipartito y G_2 completo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(m + n)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

Variante 2: En el Torneo Tricolor compiten 3 equipos: azul, amarillo y rojo. Las reglas del torneo aparecen a continuación.

- No debe haber amigos dentro del equipo azul.
- No debe haber amigos dentro del equipo amarillo.
- Los miembros del equipo rojo deber ser amigos de todos los que participan en el torneo.
- Cada equipo debe tener al menos un miembro.

Diseñar un algoritmo eficiente basado en grafos que dada la lista de amigos dentro de un grupo de personas, decida si es posible armar con ellas los 3 equipos para el Torneo Tricolor. La entrada del algoritmo es la cantidad p de personas, la cantidad r de relaciones de amistad entre ellas, y para cada amistad las dos personas que son amigas (identificadas por enteros entre 1 y p). La relación de amistad se supone simétrica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(p+r)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

SUGERENCIA: Resolver primero sin tener en cuenta la última regla.

Variante 3: En el Torneo Tricolor compiten 3 equipos: azul, amarillo y rojo. Las reglas del torneo aparecen a continuación.

- Todos los miembros del equipo azul deben ser amigos entre sí.
- Todos los miembros del equipo amarillo deben ser amigos entre sí.
- Ningún miembro del equipo rojo puede tener amigos dentro del torneo.
- Cada equipo debe tener al menos un miembro.

Diseñar un algoritmo eficiente basado en grafos que dada la lista de amigos dentro de un grupo de personas, decida si es posible armar con ellas los 3 equipos para el Torneo Tricolor. La entrada del algoritmo es la cantidad p de personas, la cantidad r de relaciones de amistad entre ellas, y para cada amistad las dos personas que son amigas (identificadas por enteros entre 1 y p). La relación de amistad se supone simétrica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(p^2)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

SUGERENCIA: Resolver primero sin tener en cuenta la última regla.

SOLUCIÓN: Propuesto originalmente por Michel Mizrahi (Variante 3).

NOTA:

Variante 1: Si $n < 3$ devolver que no se puede. El problema consiste en determinar si G está formado por algunos nodos universales y el resto bipartito. En general va a convenir poner en G_2 todos los nodos universales, para luego tener más posibilidades de armar con el resto un grafo bipartito G_1. Poner todos los nodos universales en G_2 (si no hay nodos universales, devolver que no se puede). Si G_2 tiene $n-1$ o n nodos, recuperar respectivamente 1 o 2 nodos de G_2 y armar G_1 con un nodo en cada parte. Si por el contrario G_2 tiene a lo sumo $n-2$ nodos, determinar con BFS si los nodos no universales forman un grafo bipartito (usar una versión modificada de BFS que ignore completamente los nodos universales).

Variante 2: Armar un grafo G con un nodo por cada persona y un eje entre cada par de personas que son amigas. Aplicar a G la solución de la Variante 1.

Variante 3: Armar un grafo G con un nodo por cada persona y un eje entre cada par de personas que *no* son amigas. Aplicar a G la solución de la Variante 1.

TOMADO: 2019C1P1 08-MAY-2019 (Variante 1), 2019C1R1 12-JUL-2019 (Variante 2).

54. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, se define su grafo junta como $G_1 + G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$, es decir, el grafo que contiene a G_1 y a G_2 como subgrafos, y además contiene un eje entre cada vértice de G_1 y cada vértice de G_2. Se dice que un grafo J es un grafo junta si y sólo si existen grafos G_1 y G_2 tales que $J = G_1 + G_2$.

    Sea G_n el grafo de $n \geq 2$ vértices definido inductivamente de la siguiente manera: $G_2 = K_2$ y $G_n = G_{n-1}^{c} + K_1$ para $n > 2$.

    (a) Demostrar que G_n es conexo y G_n^{c} es no conexo.

(b) Demostrar que G_n tiene un único par de vértices de igual grado, y que lo mismo vale para G_n^c.
SUGERENCIA: Demostrar ambas cosas a la vez.

SOLUCIÓN:

(a) Es un caso particular del Ejercicio 51. Alternativamente, G_n es conexo porque tiene un vértice universal, mientras que G_n^c es no conexo porque es no trivial y tiene un vértice aislado.

(b) Inducción en n. El caso base se cumple trivialmente. En el paso inductivo, basta demostrar que para $n > 2$ el grafo G_n tiene un único par de vértices de igual grado, ya que los grados en un complemento son iguales si y sólo si lo son en el grafo original. Como G_{n-1}^c es no conexo y no trivial, entonces no tiene vértices universales, de modo que los grados de sus vértices toman valores en $0, 1, \ldots n-3$. Al agregarle un vértice universal para formar G_n, cada vértice que ya existía en G_{n-1}^c incrementa en 1 su grado, por lo que los grados de esos vértices ahora toman valores en $1, 2, \ldots n-2$. Además, por hipótesis inductiva G_{n-1}^c tiene un único par de vértices de igual grado, lo cual sigue ocurriendo al incrementar los grados. El grado del vértice universal agregado es $n-1$, que no coincide con el de ninguno de los vértices que ya existían.

Otra posibilidad es probar (por inducción en n) que en G los grados de los vértices toman todos los valores en $1, 2, \ldots n-1$, mientras que en G^c toman todos los valores en $0, 2, \ldots n-2$. Como en cada caso son $n-1$ valores y hay n vértices, entonces hay un grado que se repite, pero no más que uno porque en tal caso los grados no tomarían $n-1$ valores diferentes.

Notar que por el Ejercicio 5.4 de la Práctica todo grafo tiene al menos un par de vértices de igual grado, de modo que sólo restaría probar que en G y G^c tales pares son únicos.

NOTA: Propuesto originalmente por Leopoldo Taravilse. De acuerdo al Ejercicio 55 los grafos G_n y G_n^c son los únicos que cumplen lo que dice el enunciado. Aquí se demuestra la existencia de tales grafos.

TOMADO: 2017C1R1 14-JUL-2017.

55. (a) Sea G_n un grafo *conexo* de n vértices que tiene un único par de vértices de igual grado (notar que $n \geq 2$). Demostrar que G_n tiene al menos un vértice *universal*, y que si $n \geq 3$ entonces tal vértice es único.
SUGERENCIA: Para la existencia usar que G_n no puede tener vértices aislados, y suponer por absurdo que tampoco tiene vértices universales. Para la unicidad suponer por absurdo que tiene al menos dos vértices universales, lo cual implica que todo vértice tiene grado al menos 2.
SUGERENCIA: Por absurdo concluir que hay $n-2$ valores posibles para los n grados ($1, 2, \ldots n-2$ para la existencia y $2, 3, \ldots n-1$ para la unicidad).

(b) Sea H_n un grafo *no conexo* de n vértices que tiene un único par de vértices de igual grado. Demostrar que H_n tiene al menos un vértice *aislado*, y que si $n \geq 3$ entonces tal vértice es único.

(c) Demostrar que salvo isomorfismos, existe a lo sumo un grafo conexo de n vértices que tiene un único par de vértices de igual grado, y a lo sumo un grafo no conexo de n vértices que tiene un único par de vértices de igual grado.
SUGERENCIA: Inducción con absurdo en el paso inductivo.

SOLUCIÓN:

(a) Como G_n es conexo y no trivial, entonces no tiene vértices aislados. Por absurdo supongamos que además no tiene vértices universales. En tal caso hay $n-2$ valores posibles para los n grados ($1, 2, \ldots n-2$). Eso implica que hay al menos dos grados distintos repetidos, o al menos tres vértices del mismo grado, y en cualquier caso se produce un absurdo.

Por absurdo supongamos que G_n tiene más de un vértice universal. Como $n \geq 3$, entonces todo vértice tiene grado al menos 2. De tal modo hay $n-2$ valores posibles para los n grados ($2, 3, \ldots n-1$), lo cual es absurdo como en la parte anterior.

(b) Es similar al primer punto.

Como H_n es no conexo y no trivial, entonces no tiene vértices universales. Por absurdo supongamos que además no tiene vértices aislados. En tal caso hay $n-2$ valores posibles para los n grados ($1, 2, \ldots n-2$), lo cual es absurdo.

Por absurdo supongamos que H_n tiene más de un vértice aislado. Como $n \geq 3$, entonces todo vértice tiene grado a lo sumo $n-3$. De tal modo hay $n-2$ valores posibles para los n grados $(0, 1, \ldots n-3)$, lo cual es absurdo.

(c) Inducción en n.

El caso base es trivial, ya que hay un único grafo conexo de 2 vértices, y un único grafo no conexo de 2 vértices.

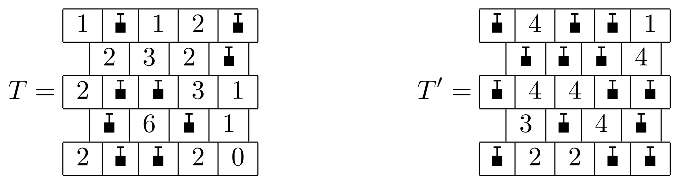
En el paso inductivo, por absurdo supongamos que existen al menos dos grafos conexos no isomorfos de $n \geq 3$ vértices que tienen un único par de vértices de igual grado. Sean A_n y B_n dos de ellos. Por el primer punto cada uno de esos grafos tiene un único vértice universal, y si lo removemos de cada grafo obtenemos grafos A_{n-1} y B_{n-1} de $n-1$ vértices que tienen un único par de vértices de igual grado (por unicidad del grado del vértice removido). Por hipótesis inductiva debe ser uno conexo y el otro no conexo. Supongamos sin pérdida de generalidad que A_{n-1} es conexo. Por el primer punto A_{n-1} tiene al menos un vértice universal, lo cual es absurdo porque entonces A_n tendría al menos dos.

Para los grafos no conexos, por absurdo supongamos que existen al menos dos grafos no conexos no isomorfos de $n \geq 3$ vértices que tienen un único par de vértices de igual grado. Sean A_n y B_n dos de ellos. Sus complementos A_n^c y B_n^c tienen un único par de vértices de igual grado, y según el Ejercicio 97b son conexos, lo cual no puede ocurrir por la primera parte. Alternativamente, remover el único vértice aislado de A_n y B_n para obtener A_{n-1} y B_{n-1} de $n-1$ vértices que tienen un único par de vértices de igual grado (por unicidad del grado del vértice removido). Por hipótesis inductiva debe ser uno conexo y el otro no conexo. Supongamos sin pérdida de generalidad que A_{n-1} es no conexo. Por el primer punto A_{n-1} tiene al menos un vértice aislado, lo cual es absurdo porque entonces A_n tendría al menos dos.

NOTA: Propuesto originalmente por Leopoldo Taravilse. De acuerdo al Ejercicio 54 existen grafos que cumplen lo que dice el enunciado. Aquí se demuestra la unicidad de tales grafos.

TOMADO: 2017C1P1 13-MAY-2017.

56. Se tiene un tablero T formado por n casillas, cada una de las cuales contiene o bien una bomba, o bien un número. A partir de T se construye otro tablero T' que tiene sus casillas dispuestas de la misma manera que en T. La diferencia es que cada casilla de T' contiene un número si la correspondiente casilla de T contiene una bomba, y viceversa. Tanto en T como en T', cuando una casilla contiene un número, esa es la cantidad de bombas en las casillas del mismo tablero que son vecinas a la dada. Demostrar mediante grafos que la suma de los números en las casillas de T es igual a la suma de los números en las casillas de T'. En la siguiente figura se muestra un posible par de tableros T y T'.



SUGERENCIA: Modelar T con un grafo o digrafo G que tiene un vértice por cada casilla del tablero.

SOLUCIÓN: Modelar T mediante un grafo G que tiene un vértice por cada casilla, y en eje entre cada par de casillas vecinas tales que una de ellas contiene una bomba y la otra un número. El grafo resulta bipartito, donde una de las partes, digamos V_1, representa a las casillas que contienen bombas, y la otra parte, digamos V_2, representa a las casillas que contienen números. La suma de los números en las casillas de T es la suma de los grados en V_2, que al ser G bipartito es la cantidad de ejes de G. Modelar T' mediante un grafo G' que tiene un vértice por cada casilla, y en eje entre cada par de casillas vecinas tales que una de ellas contiene una bomba y la otra un número. El grafo resulta isomorfo a G, y por lo tanto bipartito, pero ahora V_1 representa a las casillas que contienen números, y V_2 a las que contienen bombas. La suma de los números en las casillas de T' es la suma de los grados de V_1, que al ser G' bipartito es la cantidad de ejes de G', coincidente con la cantidad de ejes de G debido al isomorfismo.

Alternativamente, modelar T mediante un digrafo G que tiene un vértice por cada casilla, y un eje dirigido desde cada casilla que contiene un número hacia cada casilla vecina que contiene una bomba. Notar que el grado de salida de cada casilla que contiene una bomba es 0, mientras que el grado de salida de cada casilla que contiene un número es justamente ese número. Por lo tanto, la suma de los grados de salida coincide con la suma de los números en las casillas de T. Modelar T' mediante un digrafo G' que tiene un vértice por cada casilla, y un eje dirigido desde cada casilla que contiene una bomba hacia cada casilla vecina que contiene un número. Notar que el grado de entrada de cada casilla que contiene una bomba es 0, mientras que el grado de entrada de cada casilla que contiene un número es justamente ese número. Por lo tanto, la suma de los grados de entrada coincide con la suma de los números en las casillas de T'. Como los contenidos de las casillas están intercambiados en T y T', los grafos G y G' resultan isomorfos. Dado que en todo digrafo la suma de los grados de entrada coincide con la suma de los grados de salida, entonces las sumas en ambos tableros coinciden.

NOTA: Encontrado originalmente en el apunte de Lucas Andisco para la Selección para la 27ª Olimpíada Matemática del Cono Sur (2016).

TOMADO: 2016C2P1 01-OCT-2016.

57. Un polígono es una figura plana limitada por una secuencia finita de segmentos rectos llamados lados. Por ejemplo, un cuadrado es un polígono de 4 lados iguales y perpendiculares. Un poliedro es un cuerpo geométrico cuyas caras son polígonos. Por ejemplo, un cubo es un poliedro de 6 caras que son cuadrados.

Demostrar mediante grafos que si todas las caras de un poliedro tienen una cantidad impar de lados, entonces el poliedro tiene una cantidad par de caras.

SOLUCIÓN: Se puede modelar el poliedro con un grafo que tiene un vértice por cada cara, y un eje por cada lado compartido por dos caras. De tal modo, el grado de cada vértice es la cantidad de lados de la cara correspondiente. Si la cantidad de caras fuera impar, la suma de los grados del grafo sería una suma impar de valores impares, y por lo tanto impar, lo cual no puede ocurrir porque la suma de grados es siempre par.

Alternativamente, se puede modelar el poliedro con un grafo planar que tiene un vértice por cada vértice del poliedro, y un eje por cada lado entre dos vértices. De tal modo, la cantidad de ejes en el borde de cada región es la cantidad de lados de la cara correspondiente. Si la cantidad de caras fuera impar, la suma de los bordes sería una suma impar de valores impares, y por lo tanto impar, lo cual no puede ocurrir porque la suma de bordes es siempre par.

Notar que el primer grafo es el grafo dual del segundo grafo.

NOTA: Encontrado originalmente en el apunte de Lucas Andisco para la Selección para la 27ª Olimpíada Matemática del Cono Sur (2016).

TOMADO: 2016C1P1 07-MAY-2016.

58. Sea G un grafo. Sean $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ dos componentes conexas de G. Demostrar que $G_1 = G_2$ o $V_1 \cap V_2 = \emptyset$.

SOLUCIÓN: Hay varias formas parecidas de demostrarlo. Si $V_1 \cap V_2 = \emptyset$ no hay nada más que decir. Supongamos entonces $V_1 \cap V_2 \neq \emptyset$. Sean $v_1 \in V_1$, $v_2 \in V_2$, $v \in V_1 \cap V_2$. Como G_1 es conexo, existe un camino de v_1 a v; análogamente, existe un camino de v a v_2. Por lo tanto, existe un camino entre cualquier par de nodos de V_1 y V_2, pasando por v. Como G_1 es una componente conexa, y todos los nodos de V_2 son alcanzables desde V_1, resulta $V_2 \subseteq V_1$; análogamente, $V_1 \subseteq V_2$. De eso se deduce $V_1 = V_2$, lo que implica $G_1 = G_2$ ya que las componentes conexas son subgrafos inducidos.

NOTA: Propuesto originalmente por Elisa Orduna.

TOMADO: 2013C1P1 18-MAY-2013.

59. Sea G un grafo. Sean v y w dos vértices de G tales que $G - v$ es conexo, $G - w$ es conexo, y $G - \{v, w\}$ no es conexo. Demostrar que existe un ciclo simple que contiene a v y w.

SOLUCIÓN: Sea $H = G - \{v, w\}$ que por hipótesis es no conexo. Sea H_1 una componente conexa de H, y sea H_2 el resto de H. Sea v_1 un nodo en H_1, y sea v_2 un nodo en H_2. En H no hay caminos entre v_1 y v_2, pero sí en $G - v$; sea C_w uno de tales caminos, que suponemos simple, el cual pasa necesariamente por w y no por v. Análogamnete, sea C_v un camino simple entre v_1 y v_2 en $G - w$, el cual pasa necesariamente por v y no por w. Ambos caminos tienen extremos coincidentes. Si nos paramos en v y vamos por C_v hacia v_1 hasta tocar a C_w, y luego hacia v_2 hasta tocar a C_w, las porciones de ambos caminos entre los nodos coincidentes forman un circuito simple que contiene a v y w.

NOTA: Propuesto originalmente por Marina Groshaus.

TOMADO: 2012C2R1 12-DIC-2012.

60. (a) Sea G un grafo de $n \geq 3$ vértices. Demostrar que todo subgrafo inducido por 3 vértices de G tiene una cantidad par de ejes si y sólo si G no tiene ejes o es bipartito completo.

    (b) Para $n \geq 3$, sea $f(n)$ la cantidad de grafos no isomorfos de n vértices en los que todo subgrafo inducido por 3 vértices tiene una cantidad par de ejes. Por ejemplo, $f(3) = 2$ (los grafos son $3K_1$ y $K_{1,2}$), mientras que $f(4) = 3$ (los grafos son $4K_1$, $K_{1,3}$ y $K_{2,2}$). Hallar una fórmula explícita para $f(n)$. Justificar.

    SOLUCIÓN:

    (a) Para la ida, la implicación es cierta si G no tiene ejes. Si G tiene ejes, sea (v_1, v_2) cualquiera de ellos. Cualquier otro nodo w de G tiene que ser adyacente a exactamente uno de los extremos de ese eje, ya que caso contrario el subgrafo inducido por v_1, v_2 y w no tendría una cantidad par de ejes. Podemos poner entonces en una parte a los nodos adyacentes a v_2 (incluyendo v_1), y en la otra a los nodos adyacentes a v_1 (incluyendo v_2). Dentro de cada parte los nodos no son adyacentes entre sí, y cada v_i es adyacente a todos los nodos de la parte opuesta. Para probar que el grafo es bipartito completo, sólo falta argumentar que cualquier par de nodos en partes opuestas son adyacentes. Supongamos que no. Sea w_1 en la parte de v_1, que no es adyacente a w_2 en la parte de v_2. Como v_i es adyacente a todos los nodos de la otra parte, se cumple $w_i \neq v_i$. Es fácil comprobar que el subgrafo inducido por w_1, w_2 y v_1 tiene exactamente un eje, lo cual es absurdo.

    Para la vuelta, si G no tiene ejes todo subgrafo inducido de G tiene 0 ejes, que es par. Si G es bipartito completo, los subgrafos inducidos por 3 nodos que estén en la misma parte tiene 0 ejes, mientras que los subgrafos inducidos por un nodo en una parte y los otros dos en la otra tienen 2 ejes.

    (b) Cada bipartito completo queda caracterizado por la cantidad de nodos en la parte con menos nodos, que puede ir de 1 a $\lfloor n/2 \rfloor$. Como también hay que contar al grafo sin ejes, resulta $f(n) = 1 + \lfloor n/2 \rfloor$.

    NOTA: Propuesto originalmente por Leopoldo Taravilse.

    TOMADO: 2016C2P1 01-OCT-2016.

61. El grafo Q_d, también llamado hipercubo de orden d, se define inductivamente de la siguiente manera: $Q_0 = K_1$, y Q_d con $d \in \mathbb{N}$ es el grafo que se obtiene al tomar dos copias de Q_{d-1} y agregar un eje entre cada vértice de una copia y su vértice correspondiente en la otra copia. Por ejemplo, $Q_1 = K_2$ y $Q_2 = C_4$ (ciclo simple de 4 vértices).

    (a) Determinar la cantidad de vértices y la cantidad de ejes de Q_d. Expresar los resultados en función de d. Justificar.

    (b) Demostrar que Q_d es trivial o bipartito.

    SOLUCIÓN:

    (a) Informalmente, $n(Q_0) = 1 = 2^0$ y conforme crece d la cantidad de nodos se duplica, de modo que $n(Q_d) = 2^d$ (la prueba formal es por inducción en d). En cualquier grafo la suma de los grados es dos veces la cantidad de ejes. Como Q_d es d-regular (prueba formal por inducción) resulta $m(Q_d) = d \times n(Q_d)/2 = d2^d/2 = d2^{d-1}$.

    (b) Q_0 es trivial, y Q_d con $d \geq 1$ no es trivial porque tiene más de un nodo. Por lo tanto basta demostrar que si $d \geq 1$ entonces Q_d es bipartito, lo cual se puede demostrar por inducción en d. El caso base es $d = 1$, que es inmediato ya que $Q_1 = K_2$ es bipartito. Para el paso inductivo, sea (V_1, V_2) una partición de los nodos de Q_{d-1}, y sea (W_1, W_2) la copia de la partición en la copia de Q_{d-1} que se usa para construir Q_d. Consideremos la partición de los nodos de Q_d definida por $(V_1 \cup W_2, V_2 \cup W_1)$. No hay ejes dentro de $V_1 \cup W_2$ porque por hipótesis inductiva en Q_{d-1} no los hay dentro de V_1 ni dentro de W_2, y los únicos ejes nuevos en Q_d van de un conjunto de nodos a su copia. Análogamente no hay ejes dentro de $V_2 \cup W_1$. Otra forma de hacer el paso inductivo es demostrar que Q_d no tiene ciclos impares, que para grafos no triviales es equivalente a ser bipartito. Sea C un ciclo cualquiera de Q_d; queremos ver que C es par. Lo podemos probar por inducción en la cantidad de veces que el ciclo pasa de una copia de Q_{d-1} a la otra. Sea k esa cantidad (que debe ser par). El caso base es $k = 0$, lo que significa que C se queda en una copia de Q_{d-1}, de modo que C es par por hipótesis inductiva (de la inducción sobre d). Para el paso inductivo ($k \geq 2$), podemos empezar a recorrer C desde cualquiera de las copias de Q_{d-1}. En algún momento pasaremos a la otra copia con un eje e (ya que $k > 0$), y en algún momento volveremos a la primera copia con un eje f (ya que C debe cerrarse). Modifiquemos C eliminando e y f, y recorriendo lo que está entre ellos dentro de la primera copia de Q_{d-1} (es decir, pasando por los nodos originales en vez de por sus copias). Obtenemos un circuito C' que tiene la misma paridad que C, y pasa $k-2$ veces de una copia a la otra. Por hipótesis inductiva (de la inducción sobre k) el circuito C' es par, y por lo tanto también lo es C.

TOMADO: 2014C1R1 16-JUL-2014.

62. Sea $G = (V, E)$ un grafo de n vértices.

 (a) Para $n \geq 3$ impar, demostrar que para todo $v \in V$ se cumple que $G - v$ es bipartito si y sólo si G es bipartito o un ciclo simple (impar).

 (b) Para $n \geq 4$ par, demostrar que para todo $v \in V$ se cumple que $G - v$ es bipartito si y sólo si G es bipartito.

 SOLUCIÓN: Sabemos que G es bipartito o trivial si y sólo si no tiene ciclos simples impares.

 (a) Para la ida, si G es bipartito no hay nada más que probar. Si no lo es, veamos que es un ciclo simple impar. Como G no es bipartito ni trivial, tiene un ciclo simple impar C. Si hay algún nodo v fuera de C, al sacarlo subsistiría C, por lo que $G - v$ no sería bipartito. Si C tiene alguna cuerda, uno de los dos circuitos que forma la cuerda con C tiene que ser impar, y al sacar un vértice v del otro circuito $G - v$ no sería bipartito (alternativamente, elegir como C a un circuito simple impar de longitud mínima).

 Para la vuelta, como $n \geq 3$ se cumple que $G - v$ es no trivial, de modo que basta ver que no tiene ciclos simples impares. Si G es bipartito entonces no tiene ciclos simples impares por lo que el subgrafo $G - v$ tampoco los tiene. Si G es C_n entonces $G - v$ es P_{n-1} que no tiene ciclos simples.

 (b) Similar al caso impar (levemente más simple).

 NOTA: Propuesto originalmente por Guido Tagliavini Ponce. Es parecido al Ejercicio 314. Está relacionado con el Ejercicio 48.

 TOMADO: 2016C2R1 12-DIC-2016.

63. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

 Sea C_n el grafo que es un ciclo simple de $n \geq 3$ vértices. Sean p y q enteros tales que $p, q \geq 3$.

 (a) Demostrar que si p y q son pares, entonces C_p es homomorfo a C_q y C_q es homomorfo a C_p.

 (b) Demostrar que si p y q son impares, entonces C_p es homomorfo a C_q si y sólo si $p \geq q$.

 (c) Demostrar que si p es par y q es impar, entonces C_p es homomorfo a C_q pero C_q no es homomorfo a C_p.

 SOLUCIÓN:

 (a) Basta ver que C_p es homomorfo a C_q. Al ser p par, C_p es bipartito, por lo que es homomorfo a K_2 y a cualquier supergrafo del mismo, en particular C_q.

(b) Sean $v_1, v_2, \ldots v_p$ los vértices de C_p numerados siguiendo el ciclo. Análogamente, sean $w_1, w_2, \ldots w_q$ los vértices de C_q. Para la vuelta, supongamos $p \geq q$. Consideremos la función $f(v_i) = w_i$ para $i = 1, 2, \ldots q$ y $f(v_i) = w_{q-(i-q \bmod 2)}$ para $i = q+1, q+2, \ldots p$. Es decir, los vértices excedentes de C_p van a parar por f alternadamente a los vértices w_{q-1} y w_q. Se puede comprobar fácilmente que f es un homormorfismo de C_p a C_q. Para la ida, supongamos que C_p es homomorfo a C_q y sin embargo $p < q$. Sea f un homomorfismo de C_p a C_q. Se puede comprobar fácilmente que todo homomorfismo debe mandar un ciclo en un ciclo de la misma longitud. Como C_p es un ciclo, el ciclo correspondiente en C_q no puede recorrer C_q completamente porque $p < q$. Por lo tanto, el ciclo en C_q está incluído en un camino simple de C_q, lo cual implica que tiene longitud par porque un camino simple no tiene circuitos impares (es bipartito). Esto es absurdo ya que C_p es un circuito impar.

(c) Al ser p par, C_p es bipartito, por lo que es homomorfo a K_2 y a cualquier supergrafo del mismo, en particular C_q. Se puede comprobar fácilmente que todo homomorfismo debe mandar un ciclo en un ciclo de la misma longitud. Por lo tanto C_q no es homomorfo a C_p, ya que C_p es bipartito y por lo tanto no tiene circuitos impares como lo es C_q.

TOMADO: 2013C1P1 18-MAY-2013.

64. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

Sea G un grafo. Demostrar que G es bipartito o trivial si y sólo si es homomorfo a K_2.

SOLUCIÓN: Para la ida, consideremos una partición $\{V_1, V_2\}$ del conjunto de nodos tal que no hay ejes dentro de cada parte, o $V_2 = \emptyset$ si G es trivial; definamos una función que manda todos los nodos de V_1 a uno de los nodos de K_2, y todos los nodos de V_2 al otro; como no hay ejes dentro de V_1 ni dentro de V_2, la función resulta un homomorfismo. Para la vuelta, sea V_1 la preimagen de uno de los nodos de K_2, y V_2 la preimagen del otro; no puede haber ejes dentro de V_1 ni dentro de V_2 porque entonces la función no sería un homorfismo; por lo tanto, G es bipartito, o trivial si V_1 o V_2 son vacíos.

NOTA: Es esencialmente el Ejercicio 309.

TOMADO: 2012C2R1 12-DIC-2012.

65. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

(a) ¿Existe algún grafo G tal que para todo grafo H se cumple que G es homomorfo a H?

(b) ¿Existe algún grafo H tal que para todo grafo G se cumple que G es homomorfo a H?

(c) ¿Existe algún grafo H tal que para todo grafo bipartito G se cumple que G es homomorfo a H?

En caso afirmativo caracterizar en términos simples todos los grafos que cumplen lo pedido y justificar; en caso negativo demostrar.

SOLUCIÓN:

(a) Como tiene que valer en particular para H sin ejes, entonces G tampoco puede tenerlos. La respuesta es entonces que G es cualquier grafo sin ejes.

(b) Tiene que valer en particular cuando G es completo, y todos sus nodos deben mapearse a nodos diferentes (y adyacentes) en el H que cumpla lo pedido. Por lo tanto H debe contener un completo de una cantidad arbitraria de nodos, lo cual no es posible.

(c) Cualquier grafo con ejes. Dado cualquier grafo bipartito, los nodos de una parte se mandan a un extremo de un eje, y los de la otra parte al otro extremo de ese mismo eje.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2012C2P1 13-OCT-2012.

66. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

    Sea G un grafo.

    (a) ¿Es cierto que todo subgrafo de G es *homomorfo* a G?
    (b) ¿Es cierto que todo subgrafo de G es *isomorfo* a G?

    En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

    SOLUCIÓN: Para homomorfo es cierto, tomando f como la identidad. Para isomorfo no es cierto, ya que el subgrafo debería tener, por ejemplo, la misma cantidad de nodos que el grafo, lo cual en general no es cierto.

    TOMADO: 2011C2R1 14-DIC-2011.

67. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

    Sean G y H dos grafos, y f un homomorfismo de G a H. Sean $S_G = (v_1, v_2, \ldots v_k)$ una secuencia de vértices (no necesariamente distintos) de G, y $S_H = (f(v_1), f(v_2), \ldots f(v_k))$ la secuencia correspondiente de vértices de H. ¿Es cierto que...

    (a) si S_G es un camino (no necesariamente simple) en G, entonces S_H es un camino en H?
    (b) si S_G es un camino simple en G, entonces S_H es un camino simple en H?
    (c) si S_H es un camino (no necesariamente simple) en H, entonces S_G es un camino en G?
    (d) si S_H es un camino simple en H, entonces S_G es un camino simple en G?

    En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

    SOLUCIÓN:

    (a) Que es camino es trivial, por definición de homomorfismo.
    (b) Puede no ser simple, por ejemplo si se define un homomorfismo de P_4 a P_2.
    (c) Puede no ser camino, por ejemplo si no hay ejes en G pero sí en H.
    (d) Ídem punto anterior.

    NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Se puede remplazar camino por ciclo, con resultados similares.

    TOMADO: 2012C1P1 19-MAY-2012.

68. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

    Demostrar que el homomorfismo es una relación reflexiva y transitiva, pero no simétrica ni antisimétrica. Es decir, demostrar los siguientes hechos:

    (a) Todo grafo G es homomorfo a sí mismo.
    (b) Para toda terna de grafos F, G y H, si F es homomorfo a G y G es homomorfo a H, entonces F es homomorfo a H.
    (c) Existen grafos G y H tales que G es homomorfo a H pero H no es homomorfo a G.
    (d) Existen grafos G y H no isomorfos tales que G es homomorfo a H, y H es homomorfo a G.

    SOLUCIÓN:

    (a) Tomar f como la identidad.
    (b) Componer las dos funciones.
    (c) Por ejemplo $G = 2K_1$ y $H = K_2$. G es homomorfo a H porque no hay en G adyacencias que conservar al pasar a H. H no es homomorfo a G porque no es posible conservar sus adyacencias en G sin ejes.

(d) Por ejemplo P_2 y P_n con $n > 2$.

NOTA: También es cierto que no es una relación total, es decir, existen grafos G y H tales que G no es homomorfo a H, y H no es homomorfo a G. Por ejemplo si G es cualquier grafo perfecto con $\chi(G) = \omega(G) = p \geq 3$ (digamos K_3), y $H = M_q$ con $q \geq 4$ (el grafo de Mycielski número q) que tiene $\chi(H) = q$ y $\omega(H) = 2$. Según el Ejercicio 342, G no es homomorfo a H ya que debería ser $\omega(G) \leq \omega(H)$. Según el Ejercicio 308, H no es homomorfo a G ya que debería ser $\chi(H) \leq \chi(G)$. Hay familias de grafos (llamados incomparables) tales que no hay homomorfismo entre ningún par de grafos distintos dentro de la familia (según comentario de Federico Lebrón).

TOMADO: 2012C1R1 18-JUL-2012.

69. Dado un digrafo, un orden $v_1, v_2, \ldots$ de sus vértices se dice topológico si y sólo si para todo eje (v_i, v_j) del digrafo se cumple que $i < j$ (es decir, si hay un eje dirigido de v_i a v_j, entonces v_i aparece antes que v_j en el orden).

Sea G un digrafo. Demostrar que G no tiene ciclos dirigidos si y sólo si existe un orden topológico de sus vértices.

SOLUCIÓN: Para la ida, según el Ejercicio 81, G tiene un vértice con grado de entrada 0, el cual puede comenzar el orden topológico. Si removemos ese vértice del grafo, obtenemos un nuevo grafo sin ciclos dirigidos. Podemos repetir el argumento para obtener un segundo vértice, y así sucesivemente hasta obtener el orden completo. Esto puede formalizarse con inducción en la cantidad de vértices.

Para la vuelta, si G tuviera un ciclo dirigido, ninguno de los vértices del ciclo podría preceder a los otros el en orden, ya que habría un eje hacia atrás.

NOTA: Es una propiedad conocida.

70. Un grafo funcional es un (pseudo) grafo dirigido en el cual todo vértice v cumple que $d_{\text{out}}(v) = 1$, de modo tal que representa a alguna función de un conjunto en sí mismo.

Para cada uno de los ítems que aparecen a continuación, exhibir un grafo funcional que cumpla lo indicado, o demostrar que tal grafo no existe. Justificar.

(a) Existe un vértice v tal que $d_{\text{in}}(v) = 0$.
(b) Existe un vértice v tal que $d_{\text{in}}(v) = 1$.
(c) Existe un vértice v tal que $d_{\text{in}}(v) \geq 2$.
(d) Todo vértice v cumple que $d_{\text{in}}(v) = 0$.
(e) Todo vértice v cumple que $d_{\text{in}}(v) = 1$.
(f) Todo vértice v cumple que $d_{\text{in}}(v) \geq 2$.

SOLUCIÓN: En todo digrafo se cumple $m = \sum_{v \in V} d_{\text{in}}(v) = \sum_{v \in V} d_{\text{out}}(v)$; como en este caso $d_{\text{out}}(v) = 1$, todo lo anterior es igual a n.

(a) El grafo del quinto punto, con un vértice adicional que tiene como sucesor a un vértice del ciclo.
(b) El grafo del quinto punto.
(c) El grafo del primer punto.
(d) No existe, porque en tal caso $n = \sum_v d_{\text{in}}(v) = \sum_v 0 = 0$, es decir, $n = 0$.
Alternativamente se puede argumentar que si todos los grados de entrada son nulos entonces no hay ejes, y deberían ser nulos también todos los grados de salida, lo cual no ocurre en grafos funcionales.
(e) Un ciclo simple dirigido, o también un único vértice con un autoeje, o dos vértices formando un ciclo.
(f) No existe, porque en tal caso $n = \sum_v d_{\text{in}}(v) \geq \sum_v 2 = 2n$, es decir, $n \geq 2n$ que equivale a $n \leq 0$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2018C2P1 03-OCT-2018.

71. Un grafo funcional es un (pseudo) grafo dirigido en el cual todo vértice v cumple que $d_{\text{out}}(v) = 1$, de modo tal que representa a alguna función de un conjunto en sí mismo. En la siguiente figura aparecen todos los grafos funcionales de hasta dos vértices.

Decidir para cada $d \in \mathbb{N}_0$ si existe un grafo funcional tal que todo vértice v cumple $d_{\text{in}}(v)+d_{\text{out}}(v) = d$. En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar.

SOLUCIÓN: En todo digrafo se cumple $m = \sum_{v \in V} d_{\text{in}}(v) = \sum_{v \in V} d_{\text{out}}(v)$; como en este caso $d_{\text{out}}(v) = 1$, todo lo anterior es igual a n. Por lo tanto, para que ocurra $d_{\text{in}}(v) + d_{\text{out}}(v) = d$ debemos tener $2n = \sum_{v \in V} d_{\text{in}}(v) + \sum_{v \in V} d_{\text{out}}(v) = \sum_{v \in V} (d_{\text{in}}(v) + d_{\text{out}}(v)) = \sum_{v \in V} d = dn$. Es decir $2n = dn$, que sólo ocurre para $d = 2$. Un grafo funcional que cumple lo pedido para $d = 2$ es un ciclo con todos los arcos dirigidos hacia el mismo lado (incluso un único nodo con un arco a sí mismo).

NOTA: Propuesto originalmente por Guido Tagliavini Ponce.

TOMADO: 2016C2R1 12-DIC-2016.

72. Un grafo funcional es un (pseudo) grafo dirigido en el cual todo vértice v cumple que $d_{\text{out}}(v) = 1$, de modo tal que representa a alguna función de un conjunto en sí mismo. En la siguiente figura aparecen todos los grafos funcionales de hasta dos vértices.

Sea G un grafo funcional.

(a) Demostrar que los ciclos de G no comparten vértices.

(b) Demostrar que cada componente débilmente conexa de G tiene un único ciclo.

SOLUCIÓN:

(a) Por absurdo supongamos que dos ciclos de G tienen algún vértice en común. Sea $v_1, v_2, \ldots v_p$ uno de los ciclos y $w_1, w_2, \ldots w_q$ el otro. Supongamos sin pérdida de generalidad que $v_1 = w_1$, y sea $j \geq 2$ el mínimo i tal que $v_i \neq w_i$. Como v_j y w_j son dos sucesores distintos entre sí de $v_{j-1} = w_{j-1}$, se cumple $d_{\text{out}}(v_{j-1}) \geq 2$, lo cual es absurdo.

(b) Si la componente tiene autoejes entonces tiene al menos un ciclo, mientras que si no tiene autoejes entonces tiene al menos un ciclo por el Ejercicio 81.

Respecto de la unicidad, supongamos por absurdo que alguna componente tiene al menos dos ciclos C_1 y C_2, los cuales por el primer punto no comparten vértices. Consideremos un camino no dirigido desde cualquier nodo de C_1 a cualquier nodo de C_2, el cual existe porque la componente es débilmente conexa. Dicho camino en algún momento debe abandonar definitivamente C_1 y luego llegar a C_2. Sea $v_1, v_2, \ldots v_k$ esa porción del camino. El primer eje del subcamino tiene dirección a v_1, porque caso contrario se tendría $d_{\text{out}}(v_1) \geq 2$. Análogamente el último eje tiene dirección a v_k. Sea (v_j, v_{j+1}) el primer eje con dirección a v_k, el cual existe porque hay ejes en esa dirección. Además, resulta $j \geq 2$ debido a que el primer eje tiene dirección a v_1. Como tomamos el primer eje con dirección a v_k, el eje anterior tiene dirección a v_1, de modo que $d_{\text{out}}(v_j) \geq 2$, lo cual no ocurre en grafos funcionales.

Alternativamente se puede demostrar utilizando propiedades de árboles sobre el grafo que se obtiene al remover las direcciones. Como dicho grafo puede ser un pseudo o multi grafo, para transformarlo en un grafo simple se pueden agregar nodos en los autoejes o ejes múltiples, y argumentar que eso no modifica la cantidad de ciclos. O considerar esos casos por separado.

NOTA: Propuesto originalmente por Guido Tagliavini Ponce (primer punto). Se usa en el Ejercicio 73.

TOMADO: 2019C1P1 08-MAY-2019.

73. Un grafo funcional es un (pseudo) grafo dirigido en el cual todo vértice v cumple que $d_{\text{out}}(v) = 1$, de modo tal que representa a alguna función de un conjunto en sí mismo. En la siguiente figura aparecen todos los grafos funcionales de hasta dos vértices.

Cada componente débilmente conexa de un grafo funcional tiene un único ciclo.

Sea G un grafo funcional.

(a) Demostrar que existe un vértice v tal que $d_{\text{in}}(v) = 0$ si y sólo si existe un vértice w tal que $d_{\text{in}}(w) \geq 2$.

(b) Demostrar que son equivalentes:

i. G es fuertemente conexo.

ii. G es débilmente conexo y para todo vértice v se cumple que $d_{\text{in}}(v) = 1$ (esto último equivale a decir que la función es biyectiva).

iii. G es un ciclo dirigido.

SOLUCIÓN: En todo digrafo se cumple $m = \sum_{v \in V} d_{\text{in}}(v) = \sum_{v \in V} d_{\text{out}}(v)$; como en este caso $d_{\text{out}}(v) = 1$, todo lo anterior es igual a n.

(a) Para la ida, supongamos por absurdo que para todo vértice w se cumple que $d_{\text{in}}(w) \leq 1$. En tal caso tenemos $n = \sum_{w \in V} d_{\text{in}}(w) = d_{\text{in}}(v) + \sum_{w \neq v} d_{\text{in}}(w) \leq 0 + \sum_{w \neq v} 1 = n - 1$, es decir, $n \leq n - 1$.

Para la vuelta, supongamos por absurdo que para todo vértice v se cumple que $d_{\text{in}}(v) \geq 1$. En tal caso tenemos $n = \sum_{v \in V} d_{\text{in}}(v) = d_{\text{in}}(w) + \sum_{v \neq w} d_{\text{in}}(v) \geq 2 + \sum_{w \neq v} 1 = n + 1$, es decir, $n \geq n + 1$.

(b)

(73(b)i) $\Rightarrow$ (73(b)ii): Como G es fuertemente conexo, según el Ejercicio 5.10.a de la Práctica el grafo subyacente es conexo. Por lo tanto, G sin las direcciones también es conexo, ya que tiene los mismos nodos que el grafo subyacente, y no tiene menos ejes. Es decir, G es débilmente conexo por definición.

Respecto de los grados, por absurdo supongamos que existe v tal que $d_{\text{in}}(v) \neq 1$, es decir, $d_{\text{in}}(v) = 0$ o $d_{\text{in}}(v) \geq 2$. Por el primer punto ambas situaciones son equivalentes, de modo que basta analizar la situación $d_{\text{in}}(v) = 0$. Por ese mismo punto existe un vértice w tal que $d_{\text{in}}(w) \geq 2$. Como tienen distintos grados de entrada se tiene $v \neq w$, y al ser $d_{\text{in}}(v) = 0$ no existe camino dirigido de w a v, lo cual contradice que G sea fuertemente conexo.

(73(b)i) $\Rightarrow$ (73(b)iii): Por el mismo argumento usado en (73(b)i) $\Rightarrow$ (73(b)ii) sabemos que G es débilmente conexo, lo cual implica que tiene un único ciclo según el enunciado (aunque se prueba en el Ejercicio 72). Si hubiera algún nodo fuera del ciclo, como G es fuertemente conexo habría algún camino dirigido al ciclo, y el primer vértice v del ciclo al que se llega siguiendo ese camino tendría $d_{\text{in}}(v) \geq 2$; por un punto anterior eso equivale a que exista un vértice w tal que $d_{\text{in}}(w) = 0$; tal vértice w no sería alcanzable desde v, lo cual contradice que G sea fuertemente conexo. Si hubiera ejes adicionales entre los nodos de ciclo, habría algún vértice v tal que $d_{\text{out}}(v) \geq 2$, lo cual no ocurre en grafos funcionales.

(73(b)ii) $\Rightarrow$ (73(b)i): Pendiente, aunque innecesario.

(73(b)ii) $\Rightarrow$ (73(b)iii): Como G es débilmente conexo tiene un único ciclo según el enunciado (aunque se prueba en el Ejercicio 72). Si hubiera algún nodo fuera del ciclo, al ser G débilmente conexo habría algún camino al ciclo ignorando las direcciones, y el primer vértice v del ciclo al que se llega siguiendo ese camino tendría $d_{\mathrm{in}}(v) \geq 2$ que no ocurre por hipótesis, o $d_{\mathrm{out}}(v) \geq 2$ que no ocurre en grafos funcionales. Si hubiera ejes adicionales entre los nodos de ciclo, habría algún vértice v tal que $d_{\mathrm{out}}(v) \geq 2$, también una contradicción.

(73(b)iii) $\Rightarrow$ (73(b)i): Se puede ir de cualquier vértice a cualquier otro siguiendo el ciclo.

(73(b)iii) $\Rightarrow$ (73(b)ii): El grafo es débilmente conexo porque se puede ir de cualquier vértice a cualquier otro siguiendo el ciclo.
Respecto de los grados, por hipótesis el grafo está formado por n vértices $v_1, v_2, \ldots v_n$ tales que cada vértice es antecesor del siguiente, v_n es antecesor de v_1 y no hay otros ejes. Como cada vértice tiene un único antecesor, su grado de entrada es 1.

NOTA: Propuesto originalmente por Guido Tagliavini Ponce (implicación (73(b)i) $\Rightarrow$ (73(b)ii).

TOMADO: 2019C1R1 12-JUL-2019 (primer punto).

74. Decidir si las siguientes secuencias son gráficas. En caso afirmativo decidir si existe más de un grafo no isomorfo que tenga la misma secuencia de grados y justificar; en caso negativo demostrar.

    (a) $D_1 = (0, 0, 1, 1, 0, 0)$.
    (b) $D_2 = (0, 1, 2, 2, 3, 0)$.
    (c) $D_3 = (1, 2, 3, 3, 4, 5)$.
    (d) $D_4 = (1, 2, 3, 4, 5)$.
    (e) $D_5 = (1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 5)$.
    (f) $D_6 = (2, 2, 3, 3, 4, 4)$.
    (g) $D_7 = (1, 2, 1, 1, 2, 1)$.
    (h) $D_8 = (0, 0)$.

    SOLUCIÓN:

    (a) Es único: $4K_1 + K_2$.
    (b) Es único: en el grafo del punto anterior dos vértices dejan de estar aislados.
    (c) Es único: en el grafo del punto anterior dos vértices dejan de estar aislados.
    (d) No es gráfica: la suma no es par.
    (e) Hay dos no isomorfos: uno es formando K_3 con los vértices de grado 2, y otra componente conexa con el resto; otro es formando K_4 con los vértices de grado 3, y otra componente conexa con el resto.
    (f) Hay dos no isomorfos.
    (g) Hay dos no isomorfos: $2K_{1,2}$ y $K_2 + P_4$.
    (h) Es único: $2K_1$.

    TOMADO: 2011C2P1 15-OCT-2011.

75. Sea G un grafo de n vértices, A su matriz de adyacencia, y $B = A^2$.

    (a) Demostrar que B_{ij} es la cantidad de vértices que son adyacentes tanto al vértice i como al vértice j.
    (b) Demostrar que K_3 es subgrafo de G si y sólo si existen vértices i y j tales que $A_{ij} \neq 0$ y $B_{ij} \neq 0$.
    (c) Se desea decidir si K_3 es subgrafo de G, y en caso afirmativo encontrar los 3 vértices que forman el subgrafo. Es sencillo resolver este problema en $O(n^3)$, analizando todas las ternas posibles de vértices. Diseñar un algoritmo de menor complejidad que la mencionada. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n^{2.3728596})$, y utiliza el hecho de que si $M \in \mathbb{R}^{n \times n}$ entonces es posible calcular M^2 en $O(n^{2.3728596})$ (demostrado en 2020 por Alman y Williams).

SOLUCIÓN:

(a) Sabemos que B_{ij} es la cantidad de caminos de longitud 2 entre i y j. Hay una biyección entre esos caminos y los vértices adyacentes tanto a i como a j.

(b) Para la ida, si K_3 está formado por los nodos u, v y w, entonces u y v son adyacentes, lo que implica $A_{uv} \neq 0$; además, w es adyacente a ambos, y usando el punto anterior resulta $B_{uv} \neq 0$. Para la vuelta, $A_{ij} \neq 0$ implica que i y j son adyacentes, y $B_{ij} \neq 0$ implica que existe un nodo adyacente a ambos.

(c) Dada A, elevarla al cuadrado para obtener B, recorrer todos los pares i y j para encontrar lo indicado en el primer punto. Por la equivalencia K_3 es subgrafo si y sólo si se encuentra lo pedido. En caso de que se encuentre, recorrer las filas i y j de A buscando un tercer nodo adyacente a ambos.

NOTA: Basado en lo visto en la materia PGTC.

TOMADO: 2007C1R1 17-JUL-2007.

76. Sea $G = (V, E)$ un digrafo de n vértices y m ejes. Existen varias estructuras de datos que permiten representar a G. Si se utilizan *listas de sucesores*, para cada vértice $v \in V$ se dispone de una lista donde aparecen todos los vértices $w \in V$ tales que $(v, w) \in E$. De manera similar, en las *listas de predecesores*, para cada vértice $v \in V$ se tiene una lista donde aparecen todos los vértices $w \in V$ tales que $(w, v) \in E$.

Diseñar un algoritmo para cada uno de los problemas indicados, que tenga la complejidad mencionada en cada caso. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

(a) Dado G representado con listas de sucesores, representarlo con listas de predecesores, con complejidad $O(m + n)$.

(b) Dado G representado con listas de sucesores, invertir todos los ejes y representar el resultado con listas de sucesores, con complejidad $O(m + n)$.

(c) Dado G representado con listas de sucesores, invertir todos los ejes y representar el resultado con listas de predecesores, con complejidad estrictamente mejor que $\Theta(m + n)$.

SOLUCIÓN: Para los dos primeros puntos, recorrer las listas y por cada entrada que representa un arco, agregar la entrada correspondiente en la otra representación. Para el tercer punto no es necesario hacer nada; el algoritmo resulta $O(1)$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Se pueden obtener variantes cambiando “sucesores” por “predecesores” o viceversa. Relacionado con el Ejercicio 77.

TOMADO: 2018C1P1 09-MAY-2018.

77. Sea $G = (V, E)$ un digrafo de n vértices y m ejes, donde cada vértice se identifica por un entero distinto entre 1 y n. Existen varias estructuras de datos que permiten representar a G. Si se utilizan *listas de sucesores*, para cada vértice $v \in V$ se dispone de una lista donde aparecen todos los vértices $w \in V$ tales que $(v, w) \in E$. De manera similar, en las *listas de predecesores*, para cada vértice $v \in V$ se tiene una lista donde aparecen todos los vértices $w \in V$ tales que $(w, v) \in E$. En ambas representaciones, los elementos en las listas podrían aparecer de cualquier manera o estar ordenados.

Diseñar un algoritmo de complejidad $O(m+n)$ para cada uno de los problemas indicados. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

(a) Dado G representado con listas de sucesores, representarlo con listas ordenadas de predecesores.

(b) Dado G representado con listas de predecesores, representarlo con listas ordenadas de sucesores.

(c) Dado G representado con listas de sucesores, representarlo con listas ordenadas de sucesores.

SOLUCIÓN:

(a) Recorrer los nodos en orden, y para cada v recorrer su lista original (sucesores). Por cada w de la lista original de v, agregar a v en la lista deseada (predecesores) de w.

(b) El mismo algoritmo del primer punto sirve, ya que las listas tienen los significados invertidos.

(c) Basta aplicar el algoritmo del primer punto, y luego el del segundo punto. En esencia, dos veces el mismo algoritmo.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez (grafos no dirigidos). Se pueden obtener variantes cambiando "sucesores" por "predecesores" o viceversa. Relacionado con el Ejercicio 76.

78. (a) Sea G un digrafo de $n \geq 2$ vértices y m arcos. Demostrar que si G *es* fuertemente conexo entonces $m \geq n$. Demostrar que si G *no es* fuertemente conexo entonces $m \leq (n-1)^2 = n(n-1) - (n-1)$.

SUGERENCIA: Para la segunda parte demostrar que a G le "faltan" al menos $n-1$ arcos.

(b) Exhibir una familia infinita de digrafos fuertemente conexos tal que el n-ésimo digrafo tenga n vértices y n arcos ($n \geq 2$). Justificar.

(c) Exhibir una familia infinita de digrafos no fuertemente conexos tal que el n-ésimo digrafo tenga n vértices y $(n-1)^2$ arcos ($n \geq 2$). Justificar.

(d) Se quiere demostrar la equivalencia de $n \geq 2$ predicados $p_1, p_2, \ldots p_n$. Para ello se planea demostrar implicaciones del tipo $p_i \Rightarrow p_j$. Mostrar que es necesario demostrar al menos n y a lo sumo $1 + (n-1)^2$ de dichas implicaciones.

SOLUCIÓN:

(a) Si G es fuertemente conexo, sea H su grafo subyacente, el cual es conexo por el Ejercicio 5.10.a de la Práctica, y por lo tanto se cumple $m(H) \geq n(H) - 1$. Por absurdo supongamos que para G tenemos $m(G) \leq n(G) - 1$. El digrafo G no puede tener menos ejes que el grafo subyacente H, y ambos grafos coinciden en cantidad de nodos. En consecuencia tenemos $n(G)-1 \geq m(G) \geq m(H) \geq n(H)-1 = n(G)-1$, lo que implica que estamos en presencia de igualdades. Como H es conexo y $m(H) = n(H) - 1$, el grafo es un árbol, de modo que para todo par de nodos existe un único camino simple entre ellos. Como $m(G) = m(H)$, el digrafo G no tiene más ejes, por lo que el camino que existe en G puede ser de ida, o de vuelta, o ninguno, pero no ambos.

Si G no es fuertemente conexo, existen vértices v_{n-1} y v_n sin camino dirigido del primero al segundo. Eso implica que el arco (v_{n-1}, v_n) no está en G, como así tampoco al menos uno de cada par de arcos (v_{n-1}, v_i) y (v_i, v_n), donde $v_1, v_2, \ldots v_{n-2}$ son los restantes vértices de G. En total hay al menos $n-1$ arcos que no existen en G. Como un digrafo completo de n vértices tiene $n(n-1)$ arcos, entonces G tiene a lo sumo lo que dice el enunciado.

(b) Ciclos dirigidos de n vértices.

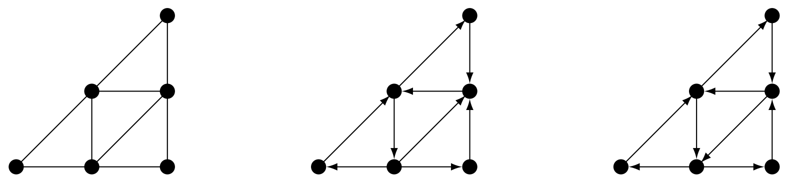
(c) Completo de $n-1$ vértices (cada par conectado en ambos sentidos), y arcos desde todos esos vértices hacia el vértice restante. La cantidad de arcos es $(n-1)(n-2) + n - 1 = (n-1)^2$. El grafo no es fuertemente conexo porque no se puede salir del último vértice.

(d) Surge de los dos primeros puntos.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2017C2R1 11-DIC-2017.

79. Un triángulo de un grafo es un circuito simple de 3 vértices. Una orientación de los ejes de un grafo es triangular si y sólo si todo arco pertenece a algún triángulo orientado con todos sus arcos en el mismo sentido. Por ejemplo, si el grafo es el de la izquierda, la orientación del centro no es triangular, mientras que la de la derecha sí lo es.



(a) Demostrar que al orientar de manera triangular un grafo conexo, el digrafo que se obtiene es fuertemente conexo.

(b) Exhibir un grafo conexo tal que todo eje pertenezca al menos a un triángulo, pero que no se pueda orientar de manera triangular. Justificar.

(c) ¿Puede orientarse el grafo del punto anterior (obviamente, de manera no triangular) para que el digrafo obtenido sea fuertemente conexo? Justificar.

SOLUCIÓN:

(a) Hay que demostrar que existe un camino dirigido entre cualquier par de vértices. Sean u y v dos vértices. Por ser el grafo conexo, existe un camino de u a v. Consideremos cada eje de ese camino. En el digrafo, podemos movernos de un extremo al otro de cada uno de esos ejes, usando el triángulo dirigido que contiene al eje, y de esa manera ir avanzando de u a v.

(b) Cualquier W_n (rueda de n rayos) con $n \geq 3$ impar sirve, aunque para $n \geq 5$ es más fácil de analizar porque cada eje que no es rayo pertenece a un único triángulo. Para ver que ese grafo sirve, hacemos un análisis de casos. Conviene ir orientando en orden cada eje que no es rayo, junto con el triángulo al cual pertenece.

(c) El grafo necesariamente va a poder orientarse de manera tal que el digrafo obtenido sea fuertemente conexo, ya que todo eje pertencese a algún circuito simple (algún triángulo), y por el Ejercicio 5.10.b de la Práctica, esto es condición suficiente.

NOTA: Propuesto originalmente por Flavia Bonomo.

TOMADO: 2002C1P1 18-MAY-2002, 2013C1R1 17-JUL-2013.

80. Un torneo (también llamado digrafo completamente conexo) es una orientación de un grafo completo. Más precisamente, un torneo es un digrafo que se obtiene asignando una dirección a cada eje de un grafo completo no dirigido.

Demostrar que cualquier torneo tiene a lo sumo un vértice con grado de entrada 0, y a lo sumo un vértice con grado de salida 0.

SOLUCIÓN: Por absurdo, considerar que tiene 2 o más, y ver que en cualquier par de ellos, uno tiene grado de salida positivo, y el otro tiene grado de entrada positivo, ya que existe el menos un arco que va del primero al segundo.

TOMADO: 2002C1P1 18-MAY-2002, 2016C2P1 01-OCT-2016.

81. Sea G un digrafo tal que todo vértice tiene grado de entrada positivo. Demostrar que G contiene un ciclo simple dirigido. Ídem si todo vértice tiene grado de salida positivo.

SOLUCIÓN: Para grado de entrada, considerar un camino simple dirigido de longitud máxima. Sea v el primer nodo del camino, y w cualquiera de sus predecesores. El nodo w debe estar en el camino, porque caso contrario el camino no sería de longitud máxima. De tal forma la porción del camino que va de v a w, junto con el arco que va de w a v, forman un ciclo simple dirigido.

Alternativamente, hay al menos un ciclo porque se puede elegir cualquier vértice y retroceder tantas veces como se quiera siguiendo los predecesores. Como la cantidad de vértices es finita, en algún momento se repite un vértice y se forma un ciclo simple.

Para grado de salida, la demostración es análoga, o pueden invertirse todos los arcos de G y usar el resultado anterior.

NOTA: Aparece en mi apunte de control de proyectos (la contrarrecíproca).

TOMADO: 2003C1R1 18-JUL-2003.

82. Sea G un digrafo sin ciclos dirigidos, y sea v un vértice de G. Demostrar que v es el único vértice con grado de entrada 0 si y sólo si para todo vértice $w \neq v$ de G existe un camino dirigido de v a w.

SOLUCIÓN: Para la ida, sea w cualquier nodo de G, a partir del cual nos podemos mover repetidamente a uno de sus predecesores; como no hay circuitos, en algún momento llegamos a un nodo con grado de entrada 0, que por hipótesis debe ser v; recorriendo los nodos en sentido inverso, obtenemos el camino buscado. Para la vuelta, por absurdo supongamos que v no es el único nodo con grado de entrada 0; si v no tiene grado de entrada 0, al no haber circuitos sus predecesores no son alcanzables desde v; si hay otro nodo con grado de entrada 0, ese nodo no es alcanzable desde v.

NOTA: Aparece en mi apunte de control de proyectos.

TOMADO: 2010C2P1 16-OCT-2010.

83. Sea G un grafo tal que para todo vértice v se cumple $d(v) \geq k$.

 (a) Demostrar que G tiene un camino simple de k o más ejes.

 (b) Demostrar que si $k \geq 2$ entonces G tiene un circuito simple de $k+1$ o más ejes.

 SOLUCIÓN: Para el primer punto, considerar un camino simple de longitud máxima. El camino debe contener al menos $k+1$ nodos, porque caso contrario cualquiera de sus extremos sería adyacente al menos a un nodo fuera del camino, y el camino no sería de longitud máxima. Para el segundo punto, considerar también un camino simple de longitud máxima. Por el punto anterior, este camino contiene al menos $k{+}1$ nodos. Sean $v_1, v_2, \ldots v_j$ los nodos del camino, con $j \geq k{+}1 \geq 3$. Si v_j es adyacente a alguno de los nodos $v_1, v_2, \ldots v_{j-k}$, se forma un circuito simple del largo requerido. Caso contrario, v_j es adyacente a lo sumo a los $k-1$ nodos $v_{j-k+1}, v_{j-k+2}, \ldots v_{j-1}$. Por lo tanto, es adyacente a otro nodo fuera del camino, lo cual es absurdo porque el camino no sería de longitud máxima.

 NOTA: También puede demostrarse constructivamente.

 TOMADO: 2003C1RX 11-AGO-2003 (con k en vez de $k+1$ para circuito simple).

84. Sea G un grafo de n vértices tal que para todo vértice v se cumple que $d(v) \geq (n-1)/2$. Demostrar que G es conexo.

 SOLUCIÓN: Hay varias formas de demostrarlo. La más fácil parece ser suponer que G no es conexo, tomar un nodo de una de sus componentes conexas, otro de otra, y deducir que en cada una de las dos componentes conexas debe haber por lo menos $1 + (n-1)/2$ nodos (el nodo elegido junto con sus adyacentes), dando un total de $n+1$ nodos sólo para esas dos componentes conexas.

 TOMADO: 2001C2R1 19-DIC-2001, 2017C2P1 04-OCT-2017.

85.

Variante 1: Demostrar que un grafo de n vértices que tiene más de $(n-2)(n-3)/2$ ejes tiene a lo sumo 2 componentes conexas.

Variante 2: Demostrar que un grafo de n vértices ($n \geq 2$) que tiene más de $(n-c)(n-c-1)/2$ ejes tiene a lo sumo c componentes conexas ($c = 1, 2, \ldots n-1$).
SUGERENCIA: Inducción en c.

 SOLUCIÓN:

Variante 1: Suponer por absurdo que el grafo tiene 3 o más componentes conexas. Sea k la cantidad de vértices de cualquiera de ellas; la cantidad de ejes de esa componente es a lo sumo $k(k-1)/2$. El resto del grafo tiene $n-k$ nodos y 2 o más componentes conexas, por lo que k está entre 1 y $n-2$. Por el Ejercicio 5.7 de la Práctica (que dice que un grafo de n nodos que tiene más de $(n-1)(n-2)/2$ ejes es conexo), el resto del grafo tiene a lo sumo $(n-k-1)(n-k-2)/2$ ejes. Por lo tanto, el número total del ejes del grafo es $k(k-1)/2 + (n-k-1)(n-k-2)/2$, que es una parábola creciente en k, por lo que su máximo está en $k = 1$ o $k = n-2$. En ambos valores de k la función vale $(n-2)(n-3)/2$, lo cual es absurdo porque por hipótesis la cantidad de ejes del grafo es mayor que ese valor.

Variante 2: El caso base de la inducción es justamente el Ejercicio 5.7 de la Práctica. El paso inductivo es similar a la solución de la Variante 1 (por absurdo se supone que hay más componentes conexas, se toma una de ellas por un lado y el resto del grafo por el otro, se acota el número de ejes en la componente conexa elegida, y por la hipótesis inductiva se acota el número de ejes en el resto del grafo). En este caso las cuentas se complican un poco al trabajar con un c genérico, pero la mecánica es la misma.

 TOMADO: 2004C1P1 22-MAY-2004 (Variante 1), 2004C1R1 21-JUL-2004 (Variante 2).

86. Sea G un grafo formado por k componentes conexas.

(a) Demostrar que si se elimina un eje de G resulta un grafo formado por k o por $k+1$ componentes conexas.

¿Se puede ser más preciso si G es un árbol? Dar el valor más exacto posible y justificar.

(b) Demostrar que si se agrega un eje a G resulta un grafo formado por k o por $k-1$ componentes conexas.

¿Se puede ser más preciso si los extremos del eje agregado pertenecen a distintas componentes conexas de G? Dar el valor más exacto posible y justificar.

SOLUCIÓN: Para el primer punto, si el eje eliminado era un puente de su componente conexa, se forman 2 componentes conexas a partir de esa, y caso contrario no pasa nada; si G es un árbol, $k = 1$ y todo eje es puente, de modo tal que pasa a tener 2 componentes conexas. Para el segundo punto, si el eje conecta 2 componentes conexas, esas pasan a formar una sola, y caso contrario no pasa nada.

TOMADO: 2005C1R1 22-JUL-2005.

87. Decidir si las siguientes afirmaciones son verdaderas o no. En caso afirmativo describir un grafo H que cumpla lo indicado y justificar; en caso negativo demostrar.

(a) Existe un grafo H tal que para todo grafo G, H tiene un subgrafo isomorfo a G.

(b) Existe un grafo H tal que para todo grafo G de $n \leq 2718$ vértices, H tiene un subgrafo isomorfo a G.

(c) Existe un grafo H tal que para todo grafo G, G tiene un subgrafo isomorfo a H.

(d) Existe un grafo H tal que para todo grafo G de $n \leq 314159$ vértices, G tiene un subgrafo isomorfo a H.

SOLUCIÓN:

(a) Falso. Si tal H existiera, tendría cierta cantidad n de nodos, todos sus subgrafos tendrían a lo sumo n nodos, y cualquier grafo G de más de n nodos tendría demasiados nodos como para poder ser isomorfo a un subgrafo de H.

(b) Verdadero. Basta que H tenga una componente conexa por cada grafo posible de $n \leq 2718$ vértices. Tales grafos son muchos, pero una cantidad finita.

(c) Verdadero. $H = K_1$ es subgrafo de cualquier grafo.

(d) Verdadero. $H = K_1$ es subgrafo de cualquier grafo.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

TOMADO: 2017C1R1 14-JUL-2017.

88. Sean G y H dos grafos. Demostrar que G y H son isomorfos si y sólo si G^c y H^c son isomorfos.

SOLUCIÓN: Para la vuelta, podemos aplicar la ida a G^c y H^c, de modo que $(G^c)^c = G$ y $(H^c)^c = H$ son isomorfos. Para la ida es como sigue. Sea f un isomorfismo de G en H, que existe por ser G y H isomorfos. Como f es una biyección de $V(G)$ en $V(H)$, y además $V(G) = V(G^c)$ y $V(H) = V(H^c)$, f es también una biyección de $V(G^c)$ en $V(H^c)$. Falta ver que preserva adyacencias en los complementos, para lo cual vamos a usar que las preserva en los grafos originales. Sean g_1 y g_2 dos vértices de G^c. Sean $h_1 = f(g_1)$ y $h_2 = f(g_2)$ sus respectivas imágenes en H^c. Si g_1 y g_2 son adyacentes en G^c, entonces no lo son en G; luego h_1 y h_2 tampoco lo son en H, y por lo tanto sí lo son en H^c. Si g_1 y g_2 no son adyacentes en G^c, entonces sí lo son en G; luego h_1 y h_2 también lo son en H, y por lo tanto no lo son en H^c.

TOMADO: 2014C1R1 16-JUL-2014.

89. Sea $G = (V, E)$ un grafo de n vértices. Demostrar que si G tiene al menos r componentes conexas entonces

$$\sum_{v \in V} d_{G^c}(v) \geq n \times (r-1) \ ,$$

donde $d_{G^c}(v)$ es el grado de v en G^c.

SOLUCIÓN: Sea $c \geq r$ la cantidad de componentes conexas de G. En G^c cada nodo es adyacente a todos los nodos que en G están en otras componentes conexas. Como cada componente conexa de G tiene al menos un nodo, en G^c cada nodo es adyacente al menos a $c-1$ nodos, y entonces $d_{G^c}(v) \geq c-1 \geq r-1$.

NOTA: Propuesto originalmente por Pablo Grassi.

TOMADO: 2003C1P1 17-MAY-2003.

90. Diseñar un algoritmo eficiente que dado un grafo $G = (V, E)$, decida si existen subconjuntos V_1 y V_2 de V tales que

    - $V_1 \cup V_2 = V$;
    - $V_1 \cap V_2 = \emptyset$;
    - $V_i \neq \emptyset$ para $i \in \{1, 2\}$; y
    - el subgrafo inducido por V_i es completo para $i \in \{1, 2\}$.

    Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n^2)$, donde $n = |V|$.

    SOLUCIÓN: Equivale a decidir si G^c es bipartito que puede hacerse con BFS luego de complementar G.

    NOTA: Es una parte de una de las variantes del Ejercicio 53.

    TOMADO: 2014C2R1 12-DIC-2014.

91. Exhibir en cada caso todos los grafos no isomorfos G, tales que G y G^c son ambos:

    (a) circuitos simples;
    (b) caminos simples;
    (c) bipartitos;
    (d) completos.

    En todos los casos justificar las respuestas.

    SOLUCIÓN: Para el primer punto, G y G^c tienen que tener n ejes, es decir $n = n(n-1)/2 - n$, de donde resulta $n = 5$, por lo que $G = G^c = C_5$; otra forma es notar que cada nodo debe tener grado 2 tanto en G como en G^c, por lo que $2 = (n-1) - 2$, de donde también resulta $n = 5$. Para el segundo punto, G y G^c tienen que tener $n-1$ ejes, es decir $n-1 = n(n-1)/2 - (n-1)$, de donde resulta $n = 1$ o $n = 4$, por lo que $G = G^c = P_1 = K_1$ o $G = G^c = P_4$. Para el tercer punto, en cada parte tiene que haber a lo sumo 2 nodos, ya que caso contrario en G^c habría un K_3; al analizar casos resulta que G puede ser $2K_1$, K_2, $K_1 \cup K_2$, P_3, $2K_2$, P_4 o C_4 (casi lo mismo que en el tercer punto del Ejercicio 305). Para el cuarto punto, G debe ser completo pero no puede tener ejes (porque faltarían en G^c), de modo que debe ser $G = G^c = K_1$.

    NOTA: Propuesto originalmente por Pablo Heiber.

    TOMADO: 2010C2P1 16-OCT-2010.

92. Caracterizar la familia de grafos que tienen la propiedad de que todo subgrafo inducido es conexo. Justificar.

    TOMADO: 2001C1R1 17-JUL-2001.

93. Determinar para qué valores de n, p, q y h los siguientes grafos tienen complemento conexo. Justificar.

    (a) K_n
    (b) C_n (ciclo simple de $n \geq 3$ vértices)
    (c) P_n (camino simple de n vértices)
    (d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)
    (e) $K_{p,q}$
    (f) árbol binario completo de altura $h \geq 0$

    SOLUCIÓN:

K_n: $K_n^c = nK_1$, que es conexo si y sólo si $n = 1$.

C_n: $C_3^c = K_3^c = 3K_1$ y $C_4^c = 2K_2$, que no son conexos. Para $n \geq 5$ se puede demostrar que C_n^c es conexo. Basta ver que existe un camino entre un vértice dado y cada uno de los otros. Sea v cualquier vértice de C_n, sean v_{-1} y v_1 sus adyacentes, sea v_{-2} el adyacente a v_{-1} que no es v, y sea v_2 el adyacente a v_1 que no es v. Estos vértices son todos distintos ya que $n \geq 5$. En el complemento existe un eje entre v y cada uno de los otros vértices, excepto v_{-1} y v_1, de modo que sólo falta encontrar un camino entre v y esos dos vértices. Para v_{-1} el camino se obtiene usando como vértice intermedio a v_2, mientras que para v_1 se obtiene usando como vértice intermedio a v_{-2}. Otra forma de demostrar que C_n^c es conexo para $n \geq 5$ es como sigue. Sean u y v dos vértices cualesquiera de C_n. Si no son adyacentes, lo son en C_n^c, de modo que existe un camino entre ellos. Si son adyacentes, sea w cualquier vértice de C_n que no es adyacente a ninguno de ellos (existe por $n \geq 5$). En C_n^c el vértice w es adyacente tanto a u como a v, de modo que también en este caso existe un camino entre u y v.

P_n: $P_1^c = K_1^c = K_1$, que es conexo. $P_2^c = K_2^c = 2K_1$, que no es conexo. P_3^c no es conexo porque tiene un nodo aislado. $P_4^c = P_4$, que es conexo. Para $n \geq 5$, P_n^c tiene un eje más que C_n^c, que es conexo, por lo tanto también lo es P_n^c.

ABC h: Los argumentos ad hoc son complicados. Una forma sencilla de resolver la cuestión es como sigue. Por el Ejercicio 6.7 de la Práctica el complemento de un árbol es conexo o es un vértice aislado junto con un subgrafo completo. Por lo tanto, la única forma de que el complemento de un árbol no sea conexo, es que el árbol sea el complemento de un nodo aislado junto con un subgrafo completo, es decir que sea una estrella. La única estrella que a la vez es un árbol binario completo de altura h es cuando $h = 1$, de modo que los complementos son conexos si y sólo si $h \neq 1$.

NOTA: No se incluye A n (variable), ni GBC n m (variable). De acuerdo al Ejercicio 51, las respuestas a este ejercicio son las complementarias de las respuestas al Ejercicio 52.

TOMADO: 2014C1P1 17-MAY-2014 (excepto W_n y $K_{p,q}$).

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

Un puente de un grafo es un eje del mismo tal que al removerlo se obtiene un grafo con más componentes conexas.

(a) ¿Existe un grafo con algún punto de corte y ningún puente?

(b) ¿Existe un grafo con algún puente y ningún punto de corte?

En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar.

SOLUCIÓN: Un ejemplo para el primer punto es un grafo formado por dos triángulos unidos por un vértice. Para el segundo punto un ejemplo es K_2.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2008C1P1 17-MAY-2008.

95. 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 y sea v uno de sus vértices. Demostrar que si v es extremo de un camino simple maximal entonces v no es punto de corte. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.
SUGERENCIA: No vale.

(b) Sea G un grafo. Usar el primer punto para demostrar que G tiene al menos un vértice que no es punto de corte.

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

SOLUCIÓN:

(a) Si v es un vértice aislado, no es punto de corte y no hay nada que probar. Si no es aislado, sea C un camino simple maximal que lo tiene a v como extremo. Como C es maximal, todo vértice adyacente a v está en $C - v$, de modo que cualquier camino que en G pasa por v, en $G - v$ puede usar nodos de $C - v$.

La recíproca no vale. Un ejemplo es $K_{p,p+1}$ con $p \geq 2$. Los vértices de la parte que tiene tamaño p no son puntos de corte ni extremos de caminos simples maximales.

(b) Todo grafo tiene un camino simple máximo, el cual es maximal, y tiene al menos un extremo, el cual no es punto de corte por el primer punto.

(c) Notar que un vértice es punto de corte si y sólo si lo es en su componente conexa. Si todas las componentes conexas de G son triviales, el resultado es inmediato tomando dos vértices cualesquiera de G. Caso contrario, sea H una componente conexa de G que no sea trivial. El grafo H tiene un camino simple máximo, el cual es maximal, y tiene al menos dos extremos distintos, los cuales no son puntos de corte por el primer punto.

NOTA: El tercer punto también se puede demostrar usando un árbol generador (ver Ejercicio 144b).

TOMADO: 2003C1RX 11-AGO-2003 (parecido), 2009C1R1 19-AGO-2009 (parecido), 2014C2P1 11-OCT-2014.

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

Sea G un grafo y sea v uno de sus vértices. ¿Es cierto que…

(a) si v está en un ciclo simple entonces no es punto de corte?

(b) vale la recíproca del punto anterior?

(c) si $d(v) \leq 1$ entonces v no es punto de corte?

(d) vale la recíproca del punto anterior?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Falso; un contraejemplo es C_n ($n \geq 3$) con una hoja agregada; el padre de la hoja está en un ciclo pero es punto de corte.

(b) Falso; un contraejemplo es P_n; los extremos no son puntos de corte pero no están en ningún ciclo.

(c) Verdadero; si $d(v) = 0$ al removerlo sólo desaparece su componente conexa, de modo que la cantidad no puede aumentar; en general, si $d(v) \leq 1$ al removerlo no puede desaparecer ningún camino entre nodos alcanzables desde v pero distintos de v, porque ninguno de esos caminos pasa por v.

(d) Falso; un contraejemplo es C_n ($n \geq 3$); cada nodo v no es punto de corte pero $d(v) > 1$.

TOMADO: 2015C1R1 13-JUL-2015.

97. Sea G un grafo.

(a) ¿Puede ocurrir que G y G^c sean ambos *conexos*?

(b) ¿Puede ocurrir que G y G^c sean ambos *no conexos*?

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

¿Puede ocurrir que G y G^c tengan ambos a un mismo vértice v como punto de corte?

En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar.

SOLUCIÓN:

(a) Si. Todos los autocomplementarios (por ejemplo K_1, P_4, C_5, C_3 con dos patas agregadas), y otros que no son autocomplementarios (por ejemplo C_4 con una pata agregada).

(b) No, por el Ejercicio 98.

Alternativamente, si G no es conexo, en G^c los nodos de componentes conexas distintas de G se conectan directamente, y los nodos de una misma componente conexa se conectan yendo a y volviendo de otra componente conexa.

(c) No, usando el punto anterior y que $(G - v)^c = G^c - v$. Si v es punto de corte de $G \Rightarrow G - v$ no es conexo $\Rightarrow (G - v)^c$ es conexo $\Rightarrow G^c - v$ es conexo $\Rightarrow v$ no es punto de corte de G^c.

TOMADO: 2001C2P1 20-OCT-2001.

98. Sea G un grafo. Demostrar que si G no es conexo entonces G^c tiene un subgrafo generador bipartito completo. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN: Sea V_1 el conjunto de nodos de cualquier componente conexa de G, y sea V_2 el conjunto de los nodos restantes. Dado que G no es conexo, ni V_1 ni V_2 son vacíos. Como en G no hay ejes que vayan de V_1 a V_2, todos esos ejes están en G^c. Si tomamos todos los nodos y esos ejes, obtenemos el subgrafo de G^c que buscamos. La recíproca es cierta.

NOTA: Sirve para demostrar el Ejercicio 97b.

TOMADO: 2001C2RX 29-DIC-2001, 2005C1P1 21-MAY-2005.

99. Sea G un grafo.

(a) Demostrar que si G no es conexo entonces G^c es conexo.

(b) Demostrar que si G es autocomplementario entonces es conexo.

(c) Demostrar que si G es no trivial y autocomplementario entonces no tiene vértices aislados ni universales.

SOLUCIÓN:

(a) Ver Ejercicio 97b.

(b) Por absurdo supongamos que G no es conexo.

Por el primer punto G^c es conexo, lo que implica que no es isomorfo a G.

Alternativamente, sea v un vértice de grado máximo en G, el cual está en una componente conexa C que tiene cierta cantidad k de vértices, de modo que $\Delta(G) \leq k - 1$. Sea w un vértice de otra componente conexa. En el complemento, w es adyacente a todos los vértices de C, por lo que su grado es al menos k, de modo que $\Delta(G^c) \geq k$. Eso es absurdo ya que $\Delta(G) = \Delta(G^c)$.

(c) No tiene vértices aislados porque entonces no sería conexo. No tiene vértices universales porque en G^c (que es isomorfo a G) serían aislados (o argumentar usando que G^c también es autocomplementario).

TOMADO: 2008C1R1 18-JUL-2008, 2017C2R1 11-DIC-2017 (excepto el primer punto).

100. (a) Se puede demostrar que la función $f(G) = G^c$ es biyectiva, de lo cual se deduce que tiene inversa biyectiva. ¿Cuál es tal inversa? Justificar.

(b) Decimos que un grafo G es anti-isomorfo a un grafo H si y sólo si G es isomorfo a H^c. Demostrar que si G es anti-isomorfo a G, entonces G no tiene 271 ni 314 vértices.

SOLUCIÓN:

(a) La inversa es f.

(b) Que G sea anti-isomorfo a G significa por definición que G es isomorfo a G^c, es decir, que G es autocomplementario. Por el Ejercicio 5.21.b de la Práctica la cantidad de vértices de un grafo autocomplementario debe ser congruente con 0 o 1 módulo 4, lo cual no cumplen ni 271 ni 314.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2015C1R1 13-JUL-2015.

101. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

(a) Sea G un grafo de n vértices, d-regular y autocomplementario. Demostrar que $n = 2d + 1$ y d es par.

SUGERENCIA: Usar que un grafo es regular si y sólo si su complemento lo es.

(b) Sea un entero positivo $n \not\equiv 1 \pmod 4$. Demostrar que hay una cantidad par de grafos regulares de n vértices.

(c) Para cada entero positivo $n \leq 7$ indicar cuántos grafos regulares de n vértices hay, y cuántos de ellos son autocomplementarios. Justificar.

SUGERENCIA: Para cada n considerar cada posible d y armar una tabla.

SOLUCIÓN:

(a) Por el Ejercicio 49 el grafo G^c es $(n-1-d)$-regular. Dado que este grafo es isomorfo a G debe ser $d = n - 1 - d$, que equivale a $n = 2d + 1$, lo que implica además que n es impar.

Por el Ejercicio 5.21.b de la Práctica debe ser $n = 4k$ o $n = 4k + 1$ para algún entero k. La alternativa $n = 4k$ queda descartada porque n sería par. De tal modo $2d+1 = 4k+1$ (ya que ambos miembros son iguales a n), lo que equivale a $d = 2k$.

(b) Por el punto anterior si un grafo de n vértices es regular y autocomplementario entonces $n = 4k + 1$, es decir, $n \equiv 1 \pmod 4$. Por contrarrecíproca, si $n \not\equiv 1 \pmod 4$ entonces no hay grafos regulares y autocomplementarios de n vértices.

Como complementar es una función biyectiva, podemos aparear a cada grafo regular de n vértices con su complemento, y por lo dicho los dos grafos de cada par son distintos entre sí. Es así que la cantidad de grafos regulares de n vértices es dos veces la cantidad de tales pares, y por lo tanto un número par.

(c) Una forma ordenada de proceder es calcular para cada n y cada d (entre 0 y $n-1$) cuántos grafos d-regulares de n vértices hay. Para $d \leq 2$, la cantidad de grafos d-regulares y $(n-1-d)$-regulares surge del Ejercicio 49. Además, como la sumatoria de grados es par, si tanto n como d son impares entonces no hay grafos d-regulares de n vértices. En la tabla siguiente aparecen las cantidades para cada combinación posible de n y d, así como el total para cada n.

		d							
		0	1	2	3	4	5	6	cualquiera
n	1	1	–	–	–	–	–	–	1
	2	1	1	–	–	–	–	–	2
	3	1	–	1	–	–	–	–	2
	4	1	1	1	1	–	–	–	4
	5	1	–	1	–	1	–	–	3
	6	1	1	2	2	1	1	–	8
	7	1	–	2	–	2	–	1	6

Por el primer punto si un grafo de n vértices es d-regular y autocomplementario entonces $n \equiv 1 \pmod 4$ y $d = (n-1)/2$. Para $n \leq 7$ las únicas posibilidades son entonces $n = 1$ y $d = 0$, o $n = 5$ y $d = 2$. Podemos ver en la tabla que para cada una de esas combinaciones hay un único grafo, que es K_1 para $n = 1$, y C_5 para $n = 5$. Ambos son autocomplementarios, y no hay grafos regulares autocomplementarios para los otros valores de n.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

TOMADO: 2018C1R1 13-JUL-2018.

102. Un vértice de un grafo se dice simplicial si y sólo si es un vértice aislado o sus adyacentes forman un subgrafo completo.

(a) Exhibir un grafo tal que todos sus vértices sean simpliciales, pero sin vértices aislados. Justificar.

(b) Exhibir un grafo tal que todos sus vértices sean no simpliciales. Justificar.

(c) Sea G un grafo y v uno de sus vértices. Demostrar que v es simplicial si y sólo si pertenece a un único subgrafo completo maximal de G.

SOLUCIÓN:

(a) K_n, con $n \geq 2$.

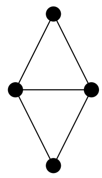
(b) C_n, con $n \geq 4$.

(c) Para la ida, si v es aislado es inmediato; si v no es aislado, y perteneciera a más de un subgrafo completo maximal, habría un nodo en cada uno de esos subgrafos que no serían adyacentes entre sí, lo cual es absurdo porque ambos son adyacentes a v. Para la vuelta, por absurdo supongamos que v tiene dos adyacentes w_1 y w_2 que no son adyacentes entre sí; si tomamos a v junto con w_i y agregamos nodos hasta formar subgrafos completos maximales, obtenemos dos subgrafos completos maximales distintos, porque w_1 y w_2 no son adyacentes.

NOTA: Basado en lo visto en la materia PGTC.

TOMADO: 2007C1P1 19-MAY-2007 (aunque el primer punto pedía algún vértice).

103. Si a K_4 se le saca cualquiera de sus ejes, se obtiene un grafo llamado diamante.



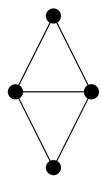
Sea G un grafo. Demostrar que el diamante es subgrafo inducido de G si y sólo si algún eje de G pertenece a más de un subgrafo completo maximal de G.

SOLUCIÓN: Para la ida, elegir un diamante, tomar los dos triángulos, y expandir cada uno de ellos hasta que se convierta en un subgrafo completo maximal; como ambas puntas del diamante no pueden pertenecer al mismo subgrafo completo, se obtienen dos subgrafos completos maximales distintos. Para la vuelta, elegir un eje que pertenezca a más de un subgrafo completo maximal, y elegir dos de esos subgrafos; por ser maximales, ningún subgrafo está contenido en el otro, de modo que existe un nodo que está en un subgrafo y no en el otro, y viceversa; los nodos del eje elegido, junto con esos otros dos nodos, inducen un diamante.

NOTA: Basado en lo visto en la materia PGTC. El ejercicio pide demostrar la contrarrecíproca de las dos implicaciones de: G es diamante-free si y sólo si todo eje de G es uniclical.

TOMADO: 2007C1R1 17-JUL-2007.

104. Si a K_4 se le saca cualquiera de sus ejes, se obtiene un grafo llamado diamante.



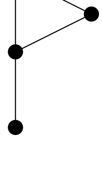
Sea G un grafo. Demostrar que si el diamante no es subgrafo inducido de G, entonces G^c tiene a lo sumo 2 componentes conexas, o todas las componentes conexas de G^c tienen exactamente un nodo.

SOLUCIÓN: Por absurdo supongamos que G^c tiene al menos 3 componentes conexas, al menos una de ellas con más de un nodo. Tomemos 2 nodos adyacentes de una componente conexa, y 2 nodos más de otras 2 componentes conexas. El subgrafo inducido por esos nodos tiene un sólo eje, por lo que en el G esos nodos inducen un diamante.

NOTA: Propuesto originalmente por Francisco Soulignac.

TOMADO: 2005C2R1 15-DIC-2005.

105. El siguiente grafo es llamado *paw* (hacha).



Sea G un grafo. Demostrar que si el grafo paw no es subgrafo inducido de G, entonces G^c es conexo o todas las componentes conexas de G^c son completas.

SOLUCIÓN: Por absurdo supongamos que G^c tiene al menos 2 componentes conexas, al menos una de ellas no completa. Cada componente conexa no completa debe tener al menos 3 nodos, porque si tuviera menos sería completa o no sería conexa. Más aún, una componente de este tipo contiene un camino simple de 3 nodos como subgrafo inducido. Para encontrarlo, basta tomar 2 nodos no adyacentes, y buscar un camino mínimo entre ellos, el cual contiene al camino de 3 nodos. Tomemos un camino como el descripto de una componente conexa no completa, y un nodo de otra componente conexa. El subgrafo inducido por esos nodos es un camino simple de 3 nodos junto con un nodo aislado, por lo que en G esos nodos inducen un paw.

NOTA: Propuesto originalmente por Francisco Soulignac.

TOMADO: 2008C1R1 18-JUL-2008.

106. (a) Sea G un grafo. Demostrar que cada componente conexa de G es un subgrafo completo si y sólo si G no contiene como subgrafo inducido a un camino simple de 3 vértices.
 (b) Diseñar un algoritmo eficiente que decida si un grafo G tiene como subgrafo inducido a un camino simple de 3 vértices. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

 SOLUCIÓN:

 (a) Para la ida, notar que todo subgrafo inducido o bien es completo o bien no es conexo, mientras que P_3 es no completo y conexo. Para la vuelta, basta ver la contrarrecíproca, es decir, que si alguna componente conexa de G no es un subgrafo completo, entonces G contiene a P_3 como subgrafo inducido. Sea C una componente conexa de G que no es un subgrafo completo. Esa componente tiene al menos 3 nodos (porque los grafos conexos de menos nodos son completos). Además, como no es un subgrafo completo, hay al menos un par de nodos de ella que no son adyacentes. El camino mínimo entre ellos tiene al menos 3 nodos, y el subgrafo inducido por los primeros 3 nodos desde alguno de sus extremos es P_3.
 (b) Por la primera parte se puede hacer BFS o DFS contando la cantidad de nodos y ejes de cada componente conexa, y en base a eso determinar si son subgrafos completos. Eso tiene la complejidad de BFS o DFS, es decir $O(m + n)$ con listas de adyacentes.

107. Sea G un grafo. Decimos que G es un cografo si y sólo si al complementar cada componente conexa de G por separado, y repetir eso un número finito de veces, se obtiene un grafo con todos los vértices de grado 0.

 Demostrar que si G es un cografo entonces no contiene como subgrafo inducido a un camino simple de 4 nodos.

 SOLUCIÓN: Por absurdo, suponer que G contiene a P_4 como subgrafo inducido. Es claro que hay vértices de G que no tienen grado 0. Al complementar la componente conexa que contiene a P_4 como subgrafo inducido, los nodos que formaban P_4 van a seguir formando P_4, de modo que nunca se va a llegar a tener todos vértices de grado 0.

 NOTA: Basado en lo visto en la materia PGTC. La recíproca también es cierta.

 TOMADO: 2005C2P1 15-OCT-2005.

108. Sea G un grafo que no contiene como subgrafo inducido a un circuito simple de 4 nodos, ni a un camino simple de 4 nodos.

 (a) Demostrar que si G es conexo, entonces tiene al menos un vértice que es adyacente a todos los otros.

 SUGERENCIA: Demostrar por el absurdo usando BFS a partir de un vértice de grado máximo.
 (b) Demostrar que si G no es trivial, entonces G no es conexo o G^c no es conexo.

 SOLUCIÓN:

(a) Sea u un vértice de grado máximo de G. Consideremos un árbol de BFS que se obtiene a partir de u. Notar que en G los nodos de niveles no consecutivos no son adyacentes. Como G es conexo, el árbol contiene a todos los nodos de G. Si el árbol tiene altura 0 o 1, es claro que u es adyacente a todos los nodos de G. Caso contrario, sea w un nodo a distancia 2 (en el árbol) de u, y sea v su padre. Notar que u no tiene grado 1 en G, dado que v tiene grado por lo menos 2, y u es de grado máximo. Notar también que v no puede ser adyacente en G a todos los otros nodos adyacentes a u, dado que en ese caso v tendría grado mayor que u, ya que v es también adyacente a w. Sea entonces x un nodo adyacente a u pero no a v. El subgrafo inducido por u, v, w y x es un circuito simple de 4 nodos si w y x son adyacentes, y es un camino simple de 4 nodos si no lo son.

(b) Si G no es conexo, listo. Si G es conexo, por el punto anterior tiene un vértice adyacente a todos los otros, de modo que en G^c ese vértice tiene grado 0, y como el grafo no es trivial, ese nodo se encuentra aislado.

NOTA: Propuesto originalmente por Flavia Bonomo. Para el primer punto no en necesario usar explícitamente BFS en la argumentación.

TOMADO: 2003C1R1 18-JUL-2003.

109. Determinar el mayor número de ejes que puede tener un grafo de n vértices no conexo. Justificar.

SOLUCIÓN: De acuerdo al Ejercicio 5.7 de la Práctica, si un grafo tiene más de $(n-1)(n-2)/2$ ejes entonces es conexo. Por lo tanto, si un grafo no es conexo tiene a lo sumo esa cantidad de ejes. Como existe un grafo no conexo que tiene esa cantidad de ejes ($K_{n-1} \cup K_1$), entonces esa cantidad es el máximo. Otra forma de demostrarlo es notar que el grafo debe tener dos componentes conexas, cada una de ellas un subgrafo completo, por lo que la solución es el máximo de la función $k(k-1)/2 + (n-k)(n-k-1)/2$, con k entero entre 1 y $n-1$.

TOMADO: 2002C1R1 17-JUL-2002.

110. Determinar el mayor número de ejes que puede tener un grafo de n vértices bipartito. Justificar.

SOLUCIÓN: Debe ser bipartitio completo. La cantidad de ejes es $k(n-k)$ con k entre 0 y $n-1$. Es una parábola en k cuyo máximo está en $k = n/2$. Si n es par la cantidad de ejes es entonces $k(n-k) = n^2/4$. Si n es impar hay que tomar el entero más próximo $k = (n \pm 1)/2$ y la cantidad de ejes resulta $(n^2-1)/4$.

TOMADO: 2001C1P1 19-MAY-2001, 2010C2R1 15-DIC-2010.

111. Sea G un grafo bipartito de n vértices. Demostrar que $\delta(G) + \Delta(G) \leq n$, donde $\delta(G)$ y $\Delta(G)$ son los grados mínimo y máximo de los vértices de G, respectivamente.

SOLUCIÓN: Consideremos una partición de los nodos de G en los conjuntos V_1 y V_2. Supongamos sin pérdida de generalidad que $|V_1| \leq |V_2|$. Se cumple que $\delta(G) \leq |V_1|$, ya que cualquier nodo de V_2 tiene grado a lo sumo $|V_1|$. Se cumple que $\Delta(G) \leq |V_2|$, ya que todos los nodos de V_1 tienen grado a lo sumo $|V_2|$, y todos los nodos de V_2 tienen grado a lo sumo $|V_1| \leq |V_2|$. Por lo tanto $\delta(G) + \Delta(G) \leq |V_1| + |V_2| = n$.

Otra forma más sencilla es argumentar que $\delta(G)$ y $\Delta(G)$ no decrecen conforme se agregan ejes a G, por lo que son máximos cuando G es bipartito completo, en cuyo caso $\delta(G) = |V_1|$ y $\Delta(G) = |V_2|$, y por lo tanto $\delta(G) + \Delta(G) = n$.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2007C1P1 19-MAY-2007.

112. Sea G un grafo no trivial. ¿Es cierto que...

(a) si G es bipartito, entonces todo subgrafo no trivial de G también es bipartito?

(b) si G es bipartito, entonces todo subgrafo propio no trivial de G también es bipartito?

(c) si todo subgrafo no trivial de G es bipartito, entonces G también es bipartito?

(d) si todo subgrafo propio no trivial de G es bipartito, entonces G también es bipartito?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar. (Un subgrafo propio de G es un subgrafo de G que es distinto de G.)

SOLUCIÓN:

(a) Verdadero; cualquiera de esos subgrafos es no trivial por definición, y no tiene circuitos de longitud impar porque G no los tiene.

(b) Verdadero por el punto anterior, ya que un subgrafo propio es en particular un subgrafo.

(c) Verdadero, ya que G es subgrafo de sí mismo.

(d) Falso; un contraejemplo es K_3.

TOMADO: 2004C1P1 22-MAY-2004.

113. Determinar para qué valores de n los siguientes grafos son bipartitos. Justificar.

 (a) K_n
 (b) C_n (ciclo simple de $n \geq 3$ vértices)
 (c) P_n (camino simple de n vértices)
 (d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

 SOLUCIÓN:

 K_n: $n = 2$; para $n = 1$ una de las partes sería vacía, y para $n \geq 3$ hay ciclos impares.

 C_n: n par; caso contrario, hay ciclos impares.

 P_n: $n \geq 2$; no hay ciclos impares, pero para $n = 1$ una de las partes sería vacía.

 NOTA: No se incluye A n (no aporta), GBC n m (trivial), $K_{p,q}$ (trivial), ni ABC h (no aporta).

 TOMADO: 2003C1RX 11-AGO-2003 (excepto W_n), 2015C1P1 16-MAY-2015 (excepto W_n).

114. Sea G un grafo bipartito conexo. Sea $\{V_1, V_2\}$ una partición de los vértices de G tal que todos los ejes tienen un extremo en V_1 y otro en V_2. Ídem $\{V_1', V_2'\}$. Demostrar que $V_1 = V_1'$ o $V_1 = V_2'$.

 SOLUCIÓN: Sea $v_2 \in V_2$. Como G es conexo, para todo vértice existe un camino desde v_2. Más aún, V_1 contiene todos los nodos a distancia (cantidad de ejes) impar de v_2, mientras que V_2 contiene todos los nodos a distancia par de v_2. Si $v_2 \in V_2'$, lo mismo pasa respectivamente con V_1' y V_2', por lo que $V_1 = V_1'$ (y $V_2 = V_2'$). Si $v_2 \in V_1'$, lo mismo pasa respectivamente con V_2' y V_1', por lo que $V_1 = V_2'$ (y $V_2 = V_1'$).

 NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

 TOMADO: 2013C1R1 17-JUL-2013.

115. Dada una colección de intervalos sobre la recta real, se define su grafo de intervalos como 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.

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

 (a) K_n
 (b) C_n (ciclo simple de $n \geq 3$ vértices)
 (c) P_n (camino simple de n vértices)
 (d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)
 (e) $K_{p,q}$

 NOTA: No se incluye A n (variable), GBC n m (variable), ni ABC h (Ejercicio 130d).

 TOMADO: 2004C1R1 21-JUL-2004 (excepto W_n y $K_{p,q}$).

116. Dada una colección de intervalos sobre la recta real, se define su grafo de intervalos como 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. Decimos que G es un grafo de intervalos propios si y sólo si existe una tal colección en la que además ningún intervalo está incluido en otro.

 Demostrar que $K_{1,3}$ es un grafo de intervalos pero no es un grafo de intervalos propios.

SOLUCIÓN: Los 3 nodos independientes deben ser 3 intervalos separados en la recta real. El intervalo del cuarto nodo debe tocar a los otros 3 para que el grafo sea de intervalos, pero en tal caso ese intervalo necesariamente va a contener al del medio del grupo de 3.

NOTA: Basado en lo visto en la materia PGTC.

TOMADO: 2009C1P1 16-MAY-2009.

117. 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.

    SOLUCIÓN: Definir S_i como el conjunto de los ejes que inciden sobre el i-ésimo vértice.

    NOTA: Basado en lo visto en la materia PGTC.

    TOMADO: 2013C2P1 12-OCT-2013.

118. (a) Demostrar que para todo grafo $G = (V, E)$ se cumple que $\delta(G) \leq 2|E|/|V| \leq \Delta(G)$, donde $\delta(G)$ y $\Delta(G)$ son los grados mínimo y máximo de los vértices de G, respectivamente.

    (b) Exhibir grafos G_1, G_2, G_3 y G_4 tales que
        i. $\delta(G_1) < 2|E_1|/|V_1| = \Delta(G_1)$,
        ii. $\delta(G_2) < 2|E_2|/|V_2| < \Delta(G_2)$,
        iii. $\delta(G_3) = 2|E_3|/|V_3| = \Delta(G_3)$, y
        iv. $\delta(G_4) = 2|E_4|/|V_4| < \Delta(G_4)$,

        o demostrar que no existen. Justificar.

    SOLUCIÓN:

    (a) Sean $n = |V|$, $m = |E|$, v_i los vértices. Se cumple $n\delta(G) = \sum_{i=1}^{n} \delta(G) \leq \sum_{i=1}^{n} d(v_i) \leq \sum_{i=1}^{n} \Delta(G) = n\Delta(G)$. Como la suma de los grados es $2m$, tenemos $n\delta(G) \leq 2m \leq n\Delta(G)$, y dividiendo por n resulta lo pedido.

    (b) Con distintos signos no hay porque el igual indica que todos los vértices deben tener el mismo grado, por lo que el mínimo coincide con el máximo. Para los otros dos casos se puede tomar como G_2 a cualquier grafo no regular, y como G_3 a cualquier grafo regular.

    NOTA: Propuesto originalmente por Leandro Montero.

    TOMADO: 2009C1R1 19-AGO-2009.

119. 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.

    (a) Demostrar que para cada entero $n \geq 1$ existe un grafo G_n tal que $K_n = L(G_n)$.

    (b) Exhibir para cada entero $n \geq 1$ todos los grafos G_n no isomorfos sin vértices aislados tales que $K_n = L(G_n)$. Justificar.

    SOLUCIÓN:

    (a) $G_n = K_{1,n}$ (estrella de n rayos).

    (b) Tomar cualquier eje de los n que tiene G_n, y luego analizar todas las formas posibles de agregar los otros respetando las adyacencias. Resultan los grafos mencionados en el primer punto y además puede ser $G_3 = K_3$ (el único que tiene dos).

    TOMADO: 2011C2P1 15-OCT-2011.

120. 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.

    Para cada entero $n \geq 2$, sea $K_n - e$ el grafo que se obtiene si se le saca a K_n cualquiera de sus ejes.

(a) Demostrar que para cada $n \in \{2, 3, 4\}$ existe un grafo G_n tal que $K_n - e = L(G_n)$.

(b) Exhibir para cada $n \in \{2, 3, 4\}$ todos los grafos G_n no isomorfos sin vértices aislados tales que $K_n - e = L(G_n)$. Justificar.

(c) Demostrar que para cada entero $n \geq 5$ no existe un grafo G_n tal que $K_n - e = L(G_n)$.
SUGERENCIA: Eventualmente demostrar primero para $n = 5$.

SOLUCIÓN:

(a) $G_2 = 2K_2$, $G_3 = P_4$, G_4 es un triángulo con un eje adyacente al mismo.

(b) Tomar cualquier eje de los $n - 1$ que tiene G_n, y luego analizar todas las formas posibles de agregar los otros respetando las adyacencias. Resultan los grafos mencionados en el primer punto.

(c) Tomar los dos ejes no adyacentes de G_n, que por ahora es $2K_2$. Es necesario agregar al menos dos ejes adiconales (ya que $n \geq 5$) los cuales son adyacentes a todos los anteriores. Un eje se agrega sin problema a G_n, que por ahora es P_4. El otro eje no se puede agregar de manera que quede adyacente a los otros tres.

NOTA: Propuesto originalmente por Federico Lebrón ($n = 5$).

TOMADO: 2011C2R1 14-DIC-2011.

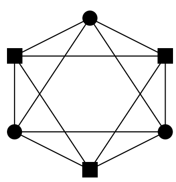
121. Dado un grafo con ejes G, se define su grafo de líneas $L(G)$ como el grafo que tiene un vértice por cada eje de G, y tal que dos vértices de $L(G)$ son adyacentes si y sólo si los ejes correspondientes de G tienen un extremo en común.

Sea G un grafo con ejes. Demostrar que $K_{1,3}$ no es subgrafo inducido de $L(G)$.

SOLUCIÓN: Si lo fuera, el nodo central sería un eje e en G adyacente a los correspondientes a los otros 3 nodos, pero esos ejes no pueden ser adyacentes entre sí, de modo que no es posible ubicarlos ya que e tiene sólo dos extremos.

TOMADO: 2012C1P1 19-MAY-2012.

122. 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. En la siguiente figura aparece $T(K_3)$; los vértices correspondientes a vértices de K_3 son círculos, mientras que los vértices correspondientes a ejes de K_3 son cuadrados.



(a) Demostrar que si G es conexo entonces $T(G)$ es conexo. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

(b) Demostrar que si G no tiene ejes entonces G y $T(G)$ son isomorfos. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

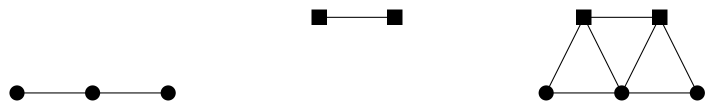
SOLUCIÓN:

(a) Dados dos vértices de $T(G)$, si corresponden a vértices en G el mismo camino que los une en G los une en $T(G)$. Si son dos ejes de G, basta tomar un extremo de cada uno en G, buscar un camino entre esos vértices en G, y los ejes que lo forman son un camino de vértices en G, cuyos extremos son adyacentes a los vértices originales. Si uno es un eje y otro un vértice, se puede tomar un extremo y el mismo vértice, y proceder de manera similar. La recíproca es cierta, ya que los nodos de $T(G)$ que no estén en G pueden ser remplazados por ejes de G.

(b) Si G no tiene ejes, por definición resulta isomorfo a $T(G)$. La recíproca es cierta, porque $T(G)$ tiene $m + n$ vértices, y la única forma de que eso sea igual a n es si $m = 0$.

TOMADO: 2013C2R1 16-DIC-2013.

123. 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 $T(G)$.

SOLUCIÓN:

(a) El eje $e = (v_i, v_j)$ de G tiene $d_i - 1$ ejes adyacentes en el extremo v_i, y $d_j - 1$ ejes adyacentes en el extremo v_j. Por lo tanto, $d(e)$ en $L(G)$ es $d_i + d_j - 2$.

(b) La cantidad de vértices de $L(G)$ es m por definición. La cantidad de ejes es la mitad de la suma de los grados de sus vértices, que son los ejes de G. Como cada vértice v_i de G aporta $d_i - 1$ a sus ejes incidentes, que son d_i, la suma de los grados de los vértices de $L(G)$ es $\sum_{i=1}^{n} d_i(d_i - 1) = \sum_{i=1}^{n} d_i^2 - d_i = -2m + \sum_{i=1}^{n} d_i^2$. La cantidad de ejes de $L(G)$ es la mitad de eso, que coincide con lo indicado en el enunciado.

(c) Cada vértice $e = (v_i, v_j)$ de $T(G)$ (correspondiente a un eje de G) tiene grado $d_i + d_j$, mientras que cada vértice v_i de $T(G)$ (correspondiente a un vértice de G) tiene grado $2d_i$.

(d) La cantidad de vértices de $T(G)$ es $m + n$ por definición. La cantidad de ejes es la mitad de la suma de los grados de sus vértices. Cada vértice v_i de G aporta d_i a sus d_i ejes incidentes (aporta $d_i - 1$ por los ejes distintos, y 1 por el propio vértice). También aporta 2 a sus d_i vértices adyacentes (1 por el eje que los une y 1 por el propio vértice). La suma de los grados de los vértices de $T(G)$ es entonces $\sum_{i=1}^{n} d_i(d_i + 2) = \sum_{i=1}^{n} d_i^2 + 2d_i = 4m + \sum_{i=1}^{n} d_i^2$. La cantidad de ejes de $T(G)$ es la mitad de eso, que coincide con lo indicado en el enunciado. Otra forma parecida de verlo es sumando los grados de los vértices de $T(G)$, dividiendo en los provenientes de ejes y de vértices de G. La forma más sencilla es notar que $T(G)$ está formado por G, $L(G)$, y dos ejes por cada vértice de $L(G)$, uniendolo con sus extremos en G; de tal modo, la cantidad de ejes de $T(G)$ es la suma de la cantidad de ejes en G, más la cantidad en $L(G)$, más dos veces la cantidad de ejes de G, es decir $m - m + \sum_{i=1}^{n} d_i^2/2 + 2m = 2m + \sum_{i=1}^{n} d_i^2/2$.

(e) Surge de la definición. Ver al final del punto anterior.

TOMADO: 2013C2P1 12-OCT-2013.

124. Sea G un grafo. ¿Es cierto que G...

(a) tiene un ciclo si y sólo si tiene un ciclo simple?

(b) tiene un ciclo impar si y sólo si tiene un ciclo simple impar?

(c) tiene un ciclo par si y sólo si tiene un ciclo simple par?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar. (La paridad de un ciclo es la paridad de su cantidad de ejes; dicha cantidad debe ser mayor que 0.)

SOLUCIÓN: La vuelta siempre es trivialmente cierta, de modo que la respuesta depende sólo de la ida.

(a) Falso; por ejemplo para K_2, o en general para cualquier bosque con ejes.

(b) Verdadero; sea C un ciclo impar de G; lo empezamos a recorrer desde cualquier vértice hasta que aparece el primer vértice repetido; consideremos la porción entre los dos vértices repetidos; si es impar, ese es un ciclo simple impar, ya que no repite vértices (por definición) y tampoco puede repetir ejes; si es par, lo podemos remover de C, y lo que queda es un ciclo impar más corto, sobre el cual podemos repetir el procedimiento; como la longitud del ciclo está acotada inferiormente, el procedimiento eventualmente termina con la extracción de un ciclo simple impar.

(c) Ídem primer punto.

TOMADO: 2014C2P1 11-OCT-2014.

Árboles, árboles generadores

125. Sea G un

Variante 1: grafo bipartito.

Variante 2a: grafo sin circuitos simples.

Variante 2b: bosque.

Variante 3: árbol.

Diseñar un algoritmo eficiente que encuentre un subgrafo completo máximo de G. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Como K_3 no es subgrafo de G, el subgrafo completo máximo es K_1 si no tiene ejes, o K_2 si tiene al menos uno. Encontrar un eje, en caso de que exista, tiene costo $O(1)$, $O(n)$ u $O(n^2)$, dependiendo de la representación.

TOMADO: 2015C2R1 11-DIC-2015 (Variante 1).

126. Sea G un

Variante 1: grafo bipartito

Variante 2a: grafo sin circuitos simples

Variante 2b: bosque

Variante 3: árbol

de n vértices. Demostrar que $K_{\lceil n/2 \rceil}$ es subgrafo de G^c.

SOLUCIÓN: Si G tiene un único vértice, el resultado es trivial. Caso contrario, G es bipartito. En tal caso al menos una de las partes del conjunto de vértices tiene al menos $\lceil n/2 \rceil$ elementos, y cualquiera de ellas en G^c induce un subgrafo completo, que tiene a $K_{\lceil n/2 \rceil}$ como subgrafo.

NOTA: Propuesto originalmente por Pablo Grassi (Variante 3).

TOMADO: 2003C1P1 17-MAY-2003 (Variante 3), 2016C1R1 11-JUL-2016 (Variante 2a).

127. (a) Sea T un árbol que tiene h hojas. Demostrar que $h \neq 1$.

(b) Sea B un bosque no trivial. Demostrar que B tiene al menos dos nodos de grado 0, o al menos dos nodos de grado 1.

SOLUCIÓN:

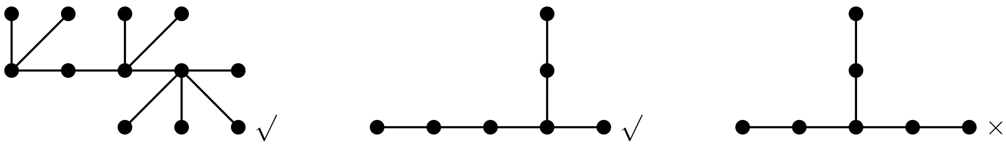
(a) Si T es trivial entonces $h = 0$. Si T no es trivial por el Ejercicio 6.5 de la Práctica se tiene $h \geq 2$.

(b) Si B tiene alguna componente conexa no trivial, esa componente conexa es un árbol no trivial, que por el Ejercicio 6.5 de la Práctica tiene al menos dos hojas; esas hojas tienen grado 1 tanto en el árbol como en el bosque. Si todas las componentes conexas de B son triviales, todos sus nodos tienen grado 0, y como B no es trivial tiene al menos dos nodos.

NOTA: Propuesto originalmente por Federico Lebrón (segundo punto).

TOMADO: 2017C1P1 13-MAY-2017.

128. 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.



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

(a) K_n

(b) $K_{p,q}$

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

SOLUCIÓN:

K_n: Es trivialmente oruga para $n \leq 2$. Para $n \geq 3$ no es oruga porque tiene ciclos, de modo que no es un árbol.

$K_{p,q}$: Para $p = 1$ o $q = 1$ es oruga (basta tomar como camino simple dominante el nodo de la parte de un solo nodo). Para $p \geq 2$ y $q \geq 2$ no es oruga porque tiene ciclos.

ABC h: Es trivialmente oruga para $h \leq 1$.

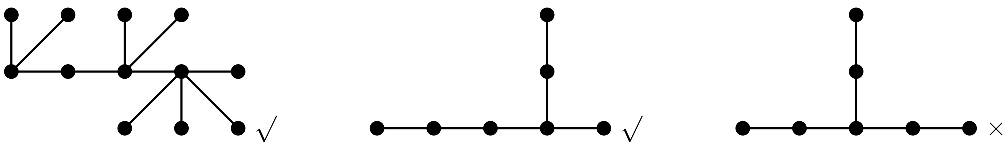
Para $h = 2$ se puede tomar como camino aquel que va de una hoja a otra que no sea hermana.

Es conocido que los grafos oruga son una subclase de los grafos de intervalos. Según el Ejercicio 130d, para $h \geq 3$ el árbol no es grafo de intervalos, de modo que no puede ser oruga. Alternativamente, si fuera oruga el camino debería conectar nodos a distancia al menos 2 de la raíz, y quedarían otros nodos a distancia al menos 2 de la raíz.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. No se incluye C_n (trivial), P_n (trivial), A n (variable), GBC n m (variable), ni W_n (trivial).

TOMADO: 2015C2P1 03-OCT-2015.

129. 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, se define su grafo de intervalos como 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.

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

SOLUCIÓN:

(a) Sean $v_1, v_2, \dots v_k$ los nodos de un camino simple dominante de G. Si tomamos k intervalos de la forma $[i, i+1]$ para $i = 1, 2, \dots k$, su grafo de intervalos es exactamente el camino. Los nodos fuera del camino deben ser adyacentes al mismo para cumplir la cota de distancia. Además, deben ser hojas porque se formarían ciclos si tuvieran más de un adyacente en el camino, o fueran adyacentes a otro nodo fuera del camino. Es así que cada nodo fuera del camino es adyacente a un único nodo del camino. Para cada nodo fuera del camino, si el nodo es adyacente al nodo v_i del camino, agregamos un intervalo included en $(i, i+1)$ que no se superponga con los otros intervalos agregados para v_i.

(b) K_3 es un grafo de intervalos (basta elegir 3 intervalos iguales), pero no es oruga porque no es un árbol (tiene ciclos). $2K_1$ es un grafo de intervalos (basta elegir 2 intervalos disjuntos), pero no es oruga porque no es un árbol (no es conexo).

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2015C2P1 03-OCT-2015.

130. Dada una colección de intervalos sobre la recta real, se define su grafo de intervalos como 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 K_n es un grafo de intervalos para todo $n \in \mathbb{N}$.
    (b) ¿Es cierto que si G es un grafo de intervalos, entonces todo *subgrafo* de G también lo es? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.
    (c) ¿Es cierto que si G es un grafo de intervalos, entonces todo *subgrafo inducido* de G también lo es? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.
    (d) Determinar para qué valores de $h \in \mathbb{N}_0$ un árbol binario completo de altura h es un grafo de intervalos. Justificar.

    SOLUCIÓN:

    (a) Considerar n intervalos exactamente iguales.
    (b) No es cierto. Por ejemplo, K_4 es un grafo de intervalos (por el punto anterior), mientras que C_4 no lo es.
    (c) Es cierto. Basta tomar la subcolección de intervalos correspondiente a los vértices del subgrafo inducido.
    (d) Son grafos de intervalos si y sólo si $h \leq 2$. Por el punto anterior, basta ver que para $h = 2$ es un grafo de intervalos, y que para $h = 3$ no lo es. Para $h = 2$ el árbol es un grafo oruga, y es conocido que los grafos oruga son una subclase de los grafos de intervalos; alternativamente, se puede encontrar a mano la colección de intervalos adecuada. Para $h = 3$, si se intenta construir los intervalos, se tiene que las adyacencias en el árbol imponen condiciones sobre los mismos, hasta que finalmente no se pueden definir intervalos para todas las hojas.

    NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

    TOMADO: 2015C2R1 11-DIC-2015.

131. Un bosque de grado acotado por d es un grafo sin ciclos en el cual todos los vértices tienen grado a lo sumo d.

    Demostrar que un grafo es un bosque de grado acotado por 2 si y sólo si cada una de sus componentes conexas es un camino simple.

    SOLUCIÓN: Similar a la demostración para grafos 2-regulares del Ejercicio 49. También se puede demostrar por inducción sacando una hoja en el paso inductivo.

    TOMADO: 2015C1P1 16-MAY-2015.

132. Sea G un grafo de n vértices.

    (a) Demostrar que G puede obtenerse a partir del grafo trivial repitiendo $n - 1$ veces el siguiente paso: agregar un vértice y cierta cantidad de ejes (eventualmente ninguno) incidentes al vértice agregado.
    (b) Demostrar que si G es un árbol, lo dicho puede lograrse aún exigiendo que al final de cada paso se obtenga un árbol.

    SUGERENCIA: Inducción en n.

    SOLUCIÓN: Inducción sacando un nodo en el paso inductivo. Para el árbol sacar una hoja (todo árbol no trivial tiene al menos dos hojas).

    NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

    TOMADO: 2014C2R1 12-DIC-2014.

133. Demostrar que para cada árbol T se cumple exactamente una de las siguientes posibilidades:

 (a) T es un grafo trivial.

 (b) T es un grafo bipartito completo con exactamente un vértice en al menos una de sus partes.

 (c) T^c es conexo y no trivial.

 SOLUCIÓN: Es claro que las 3 opciones son mutuamente excluyentes. Por el Ejercicio 6.7 de la Práctica el complemento de un árbol es conexo, o es un vértice aislado más un grafo completo. Por lo tanto, si T^c es conexo, ya sea trivial o no, estamos en el primer o en el tercer caso. Si no es conexo, por el ejercicio mencionado es un nodo más un completo, lo que significa que T es lo indicado en el segundo caso.

 NOTA: Propuesto originalmente por Pablo Heiber.

 TOMADO: 2010C2P1 16-OCT-2010.

134. (a) Sea G un grafo de 5 o más vértices. Demostrar que G tiene algún ciclo o G^c tiene algún ciclo.

 (b) Exhibir un grafo H de 4 vértices tal que tanto H como H^c no tengan ciclos. Justificar.

 SOLUCIÓN:

 (a) Por absurdo, supongamos que G y G^c no tienen ciclos. En consecuencia, la cantidad de ejes de cada uno es a lo sumo $n-1$, por lo que la suma de las cantidades de ejes en ambos (que es la cantidad de ejes en un completo) es a lo sumo $2(n-1)$, es decir $n(n-1)/2 \leq 2(n-1)$, que al ser $n \neq 1$ equivale a $n \leq 4$. Otra forma más elegante es como sigue. Si G tiene ciclos no hay nada que probar. Si no los tiene, como tiene al menos 5 nodos es bipartito con una parte de al menos 3 nodos no adyacentes, que en G^c forman un ciclo.

 (b) $H = H^c = P_4$.

 NOTA: Encontrado originalmente en Internet (primer punto).

 TOMADO: 2007C1R1 17-JUL-2007.

135. Sea G un grafo de n vértices, sin circuitos simples y formado por k componentes conexas. Demostrar que G tiene $n-k$ ejes.

 SOLUCIÓN: Cada componente conexa es un árbol, por lo que tiene un eje menos que la cantidad de vértices. Sumando sobre vértices y ejes sobre todas las componentes conexas, se obtiene el resultado

 NOTA: Ahora está en las transparencias de la Teórica.

 TOMADO: 2005C2P1 15-OCT-2005.

136. Sea G un grafo y T un árbol, ambos de n vértices.

 (a) Demostrar que si G es conexo entonces tiene igual o más ejes que T.

 (b) Demostrar que si G no tiene circuitos simples entonces tiene igual o menos ejes que T.

 SOLUCIÓN: Para el primer punto, si G tiene circuitos simples se le pueden sacar ejes hasta que se convierta en un árbol, y en ese momento se da la igualdad. El segundo punto surge de que un bosque de k componentes conexas tiene $n-k \leq n-1$ ejes ya que $k \geq 1$, y $n-1$ es la cantidad de ejes del árbol T.

 NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Esa idea talvez no estéxpresada claramente. Lo que se intenta decir es que los árboles minimizan la cantidad de ejes de los grafos conexos, y maximizan la cantidad de ejes de los grafos sin circuitos simples.

137. (a) Determinar el mayor número de ejes que puede tener un grafo de n vértices sin circuitos. Justificar.

 (b) Determinar el mayor número de ejes que puede tener un digrafo de n vértices sin ciclos dirigidos. Justificar.

 SOLUCIÓN: Para el primer punto, el grafo debe ser conexo, ya que de lo contrario se le podrían agregar ejes sin formar circuitos; por lo tanto, es conexo y sin circuitos, de modo que es un árbol, y entonces tiene $n-1$ ejes. Para el segundo punto, no puede haber más de un eje entre cada par de nodos, porque entonces habría un circuito formado por el par de nodos que los tuviera; por lo tanto, hay una cota superior de $n(n-1)/2$ ejes; esa cota se alcanza armando un grafo con todos los ejes apuntando hacia "adelante", es decir, $(i, j) \in E \Leftrightarrow i < j$.

TOMADO: 2003C1R1 18-JUL-2003.

138. Demostrar que un grafo G es un bosque si y sólo si todo subgrafo inducido de G tiene un vértice de grado 0 o 1.

SOLUCIÓN: Para la ida, cualquier subgrafo inducido es también un bosque (es decir, no tiene ciclos simples), de modo que cada una de sus componentes conexas es un árbol; tomemos cualquier componente conexa de cualquier subgrafo inducido; si tiene un único nodo, su grado es 0; si tiene más de uno, es un árbol no trivial por lo que tiene al menos 2 hojas. Para la vuelta, por absurdo supongamos que no es un bosque, es decir, tiene al menos un ciclo simple; tomemos cualquier ciclo simple, y el subgrafo inducido por sus nodos; en ese subgrafo está el ciclo, de modo que todos los nodos tienen grado al menos 2, absurdo.

TOMADO: 2001C1P1 19-MAY-2001, 2013C2R1 16-DIC-2013.

139. Sea G un grafo no dirigido de n vértices. Diseñar un algoritmo eficiente que decida si G

Variante 1a: tiene circuitos simples.

Variante 1b: es un bosque.

Variante 2: es un árbol.

Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

SOLUCIÓN: Hacer BFS o DFS para determinar las componentes conexas y al mismo tiempo contar cuántos nodos y ejes tienen. Si cada una de las componentes conexas tiene menos ejes que nodos, entonces son todas árboles y el grafo es un bosque; caso contrario tiene circuitos simples (no es un bosque). Para la Variante 2 hay que verificar adicionalmente que hay una sola componente conexa. La complejidad es la de BFS o DFS, es decir, $O(m + n)$ con listas de adyacentes, donde m es la cantidad de ejes de G. Sin embargo, el algoritmo puede parar en cuanto encuentra que G tiene n o más ejes, y entonces la complejidad es $O(n)$. Alternativamente, al comienzo puede verificarse si $m < n$, ya que de lo contrario el grafo tiene circuitos simples; luego se puede hacer BFS o DFS, sabiendo que resulta $O(m + n) = O(n)$.

TOMADO: 2004C1R1 21-JUL-2004 (Variante 1a).

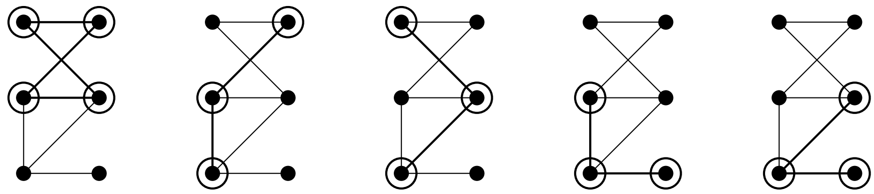
140. Sea T un árbol no trivial, y por lo tanto bipartito. Sea $\{V_1, V_2\}$ una partición de los vértices de T tal que todos los ejes tienen un extremo en V_1 y otro en V_2. Supongamos sin pérdida de generalidad que $|V_1| \geq |V_2|$. Demostrar que V_1 contiene al menos una hoja.

SOLUCIÓN: Por absurdo supongamos que V_1 no contiene hojas, y por lo tanto todos sus vértices tienen grado al menos 2, lo cual implica $n - 1 = m = \sum_{v \in V_1} d(v) \geq 2|V_1| \geq |V_1| + |V_2| = n$, es decir, $n - 1 \geq n$. También se puede demostrar usando que debe ser $|V_1| \geq n/2$, lo cual implica $n - 1 = m = \sum_{v \in V_1} d(v) \geq 2|V_1| \geq 2(n/2) = n$. También se puede demostrar por inducción en la cantidad de vértices como sigue. Casos base P_2 y P_3. En el paso inductivo, elegimos cualquier hoja h de un árbol T_n de $n \geq 4$ nodos (sabemos que todo árbol no trivial tiene al menos dos hojas). Si h está en V_1, listo. Si h está en V_2, sea p el nodo adyacente a ella (en V_1). Sacamos h de T_n para obtener un árbol T_{n-1} de $n - 1$ nodos. Se cumple que el V_1 actual es estrictamente más grande que el V_2 actual. Por hipótesis inductiva vale que T_{n-1} tiene una hoja en su V_1. Si esa hoja no es p, listo. Si es p, sacamos p y obtenemos un árbol T_{n-2} que por hipótesis inductiva tiene una hoja en su V_1, la cual no desaparece al volver a agregar p y h.

NOTA: Propuesto originalmente por Michel Mizrahi.

TOMADO: 2012C2P1 13-OCT-2012.

141. Dado un grafo G, una biclique de G es un subgrafo inducido de G que es bipartito completo maximal. En el siguiente ejemplo, podemos ver cierto grafo con sus 5 bicliques remarcadas.



Sea T un árbol de $n \geq 3$ vértices y ℓ hojas (vértices de grado 1).

(a) Demostrar que cada hoja de T pertenece a una única biclique.

(b) Demostrar que cualquier biclique de T es de la forma $K_{1,r}$ con $r \geq 2$.

(c) Demostrar que T tiene exactamente $n - \ell$ bicliques.
SUGERENCIA: Inducción en n o demostración directa.

SOLUCIÓN:

(a) Sea h una hoja, p su padre, $v_1, v_2, \ldots v_d$ los vértices adyacentes a p. Tanto h como p deben estar en cualquier biclique que contenga a h. Cualquier v_i puede agregarse a la biclique, y agregado este, sus adyacentes distintos de p no pueden agregarse ya que deberían ser adyacentes a h, y no lo son. El hecho de agregar v_i a la biclique no impide agregar los otros v_j, ya que los mismos no son adyacentes entre sí, por lo que deben agregarse todos para que el subgrafo sea maximal.

(b) Por definición, cualquier biclique de cualquier grafo es de la forma $K_{q,r}$. Sin embargo, si tanto q como r son al menos 2, la biclique tiene ciclos, lo cual no puede ocurrir si es subgrafo inducido de T. Por lo tanto, debe ser $q = 1$ o $r = 1$ (o ambos). Sin pérdida de generalidad podemos suponer $q = 1$. En tal caso, no puede ser $r = 1$, porque significaría que $K_{1,1}$ es maximal en T, lo cual sólo puede ocurrir si $n = 2$.

(c) Basta probar que el subgrafo inducido por cada nodo interno junto con sus adyacentes forma una biclique, y que no hay otras en el grafo. Claramente eso es un subgrafo inducido; es bipartito porque los adyacentes al nodo interno no son adyacentes entre sí, dado que los árboles no tienen ciclos; es completo porque el nodo interno es adyacentes a todos sus adyacentes; es maximal porque del lado de los adyacentes agregamos a todos los adyacentes del nodo interno (al menos 2, caso contrario el nodo interno en realidad sería una hoja), y del lado del nodo interno no es posible agregar nada porque la supuesta biclique no sería de la forma $K_{1,r}$ con $r \geq 2$. Para ver que no hay más, tomemos una biclique cualquiera, que sabemos que es de la forma $K_{1,r}$ con $r \geq 2$; del lado del 1 hay un nodo interno (porque tiene grado al menos $r \geq 2$), con algunos de sus adyacentes del otro lado; si no fueran todos, la supuesta biclique no sería maximal. Otra forma de demostrar todo esto es por inducción (usando el primer punto), eliminando en el paso inductivo una hoja del árbol y separando en los casos en que el padre se vuelve o no hoja.

NOTA: Propuesto originalmente por Leandro Montero.

TOMADO: 2008C1P1 17-MAY-2008 (dos primeros puntos), 2008C1R1 18-JUL-2008 (último punto).

142. (a) Demostrar que si $G = (V, E)$ es un grafo no trivial entonces

$$\sum_{v \in V} m(G - v) = m(G) \times [n(G) - 2] \quad ,$$

donde $n(\cdot)$ y $m(\cdot)$ denotan respectivamente cantidad de vértices y cantidad de ejes.

(b) Sea $G = (V, E)$ un grafo de $n \geq 3$ vértices. Demostrar que son equivalentes:

i. G es un ciclo simple.
ii. Para todo $v \in V$ se cumple que $G - v$ es un camino simple.
iii. Para todo $v \in V$ se cumple que $G - v$ es un árbol.
iv. Para todo $v \in V$ se cumple que $G - v$ es conexo y tiene exactamente dos vértices de grado 1.

SUGERENCIA: Demostrar (142(b)iii) $\Rightarrow$ (142(b)i) usando el primer punto, y (142(b)iv) $\Rightarrow$ (142(b)i).

SOLUCIÓN:

(a) $\sum m(G-v) = \sum m(G) - d(v) = \sum m(G) - \sum d(v) = n(G)m(G) - 2m(G) = m(G)\,[n(G)-2]$.

(b)

(142(b)i) $\Rightarrow$ (142(b)ii): Trivial.

(142(b)i) $\Rightarrow$ (142(b)iii): Trivial.

(142(b)i) $\Rightarrow$ (142(b)iv): Trivial.

(142(b)ii) $\Rightarrow$ (142(b)i): La misma demostración que en (142(b)iii) $\Rightarrow$ (142(b)i) sirve en este caso.
Alternativamente, sea v cualquier nodo de G, y sea $v_1, v_2, \dots v_{n-1}$ el camino $G-v$. No puede ser que v no sea adyacente en G a ningún nodo del camino, porque en tal caso v sería un nodo aislado en $G-v_1$, por lo que $G-v_1$ no sería un camino. No puede ser que v sea adyacente en G a un único nodo v_i del camino, porque en tal caso v sería un nodo aislado en $G-v_i$, por lo que $G-v_i$ no sería un camino. No puede ser que v sea adyacente en G al menos a tres nodos v_i, v_j y v_k (supongamos $i<j<k$), porque en tal caso $G-v_i$ tendría un ciclo, por lo que no sería un camino. Por lo tanto, v es adyacente a exactamente dos nodos del camino, los cuales deben ser los extremos, porque si alguno de ellos no lo fuera, sacando el otro quedaría un ciclo.

(142(b)ii) $\Rightarrow$ (142(b)iii): Trivial.

(142(b)ii) $\Rightarrow$ (142(b)iv): Trivial.

(142(b)iii) $\Rightarrow$ (142(b)i): Si G tuviera 3 o más componentes conexas, entonces $G-v$ tendría al menos 2 componentes conexas, y no se cumpliría la hipótesis. Si G tuviera 2 componentes conexas, al menos una de ellas tendría al menos 2 nodos, y al sacar alguno de ellos $G-v$ tendría al menos 2 componentes conexas, de modo que tampoco se cumpliría la hipótesis. Por lo tanto, G es conexo.
Por hipótesis $m(G-v) = n(G-v)-1$ y como $n(G-v) = n(G)-1$, resulta $m(G-v) = n(G)-2$. De acuerdo al primer punto tenemos entonces $n(G)\,[n(G)-2] = \sum n(G)-2 = \sum m(G-v) = m(G)\,[n(G)-2]$. Por lo tanto, $n(G) = m(G)$. De acuerdo al Ejercicio 6.2.b de la Práctica, G tiene exactamente un circuito. No puede tener nodos fuera del mismo porque entonces al sacar cualquiera de esos nodos el circuito permanecería. No puede tener cuerdas en el circuito porque entonces no tendría exactamente un circuito.
Alternativamente, una vez que tenemos $n(G) = m(G)$, como $d(v) = m(G) - m(G-v) = n(G) - [n(G)-2] = 2$, G resulta un grafo conexo 2-regular, de modo que es un ciclo simple según el Ejercicio 49.

(142(b)iii) $\Rightarrow$ (142(b)ii): Pendiente, aunque innecesario.

(142(b)iii) $\Rightarrow$ (142(b)iv): Pendiente, aunque innecesario.

(142(b)iv) $\Rightarrow$ (142(b)i): Puede demostrarse que G es conexo de la misma manera que en (142(b)iii) $\Rightarrow$ (142(b)i).
Veamos que G es también 2-regular. No hay hojas en G porque si las hubiera, al remover el padre de alguna de ellas quedaría un grafo no conexo (por $n \geq 3$), lo que no ocurre por hipótesis. Por lo tanto, cada nodo v es adyacente en G a las dos únicas hojas de $G-v$, las cuales tienen grado 2 en G. Sea v cualquier nodo, y sea h una hoja de $G-v$, la cual como vimos tiene grado 2 en G. Como $G-h$ tiene exactamente dos hojas que no existen en G, v debe ser una de ellas, y por lo tanto v (que es cualquier nodo) tiene grado 2 en G.
Como G es conexo y 2-regular, es un ciclo simple según el Ejercicio 49.

(142(b)iv) $\Rightarrow$ (142(b)ii): Pendiente, aunque innecesario.

(142(b)iv) $\Rightarrow$ (142(b)iii): Pendiente, aunque innecesario.

NOTA: Propuesto originalmente por Guido Tagliavini Ponce.

TOMADO: 2016C1P1 07-MAY-2016 (segundo punto, (142(b)i) $\Leftrightarrow$ (142(b)ii)), 2016C1R1 11-JUL-2016 (excepto (142(b)iv)).

143. (a) Demostrar que ningún grafo tiene exactamente 2 árboles generadores.
(b) Para cada $n \in \mathbb{N}$ exhibir un grafo de n vértices que tenga un único árbol generador. Justificar.

(c) Para cada par de valores $n \in \mathbb{N}$ y $k \in \mathbb{N}$ con $n \geq k \geq 3$, exhibir un grafo de n vértices que tenga exactamente k árboles generadores. Justificar.

SOLUCIÓN:

(a) Por absurdo, supongamos que determinado grafo tiene exactamente 2 árboles generadores (distintos) T_1 y T_2. Sea e_2 un eje que está en T_2 y no en T_1, y consideremos $T_1 + e_2$ que tiene un único ciclo, el cual está formado por al menos tres ejes. Si removemos cualquiera de los ejes del ciclo volvemos a obtener un árbol generador. Cada árbol es dintinto de los demás porque no tiene un eje que sí está en los otros árboles.

(b) P_n.

(c) P_n con un eje agregado entre uno de sus extremos y otro nodo, de manera tal que se forme un ciclo de k nodos (una especie de globo con hilo).

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez (primer punto).

TOMADO: 2017C1P1 13-MAY-2017.

144. 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.

(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.

¿Sigue valiendo la propiedad si G no es conexo? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Para la ida, sea T un árbol generador de G en el cual v es hoja. Como $T - v$ es árbol, resulta un árbol generador de $G - v$. Por lo tanto, $G - v$ es conexo, ya que $T - v$ lo es, de modo que v no es punto de corte de G.

Para la vuelta, por hipótesis $H = G - v$ es conexo. Sea T_H un árbol generador de H. Como G es conexo, en G existe un eje e que va de v a algún vértice de T_H. Sea T_G el resultado de agregar v y e a T_H. Es facil comprobar que T_G es árbol generador de G, con v como hoja.

(b) Sea T un árbol generador de G. Por el Ejercicio 6.5 de la Práctica, T tiene al menos 2 hojas, las cuales no son puntos de corte por la ida del primer punto.

La propiedad sigue valiendo si G no es conexo. Notar que un vértice es punto de corte si y sólo si lo es en su componente conexa. Si todas las componentes conexas de G son triviales, el resultado es inmediato tomando dos vértices cualesquiera de G. Caso contrario, elegir una componente conexa de G que no sea trivial, y utilizar el punto anterior.

NOTA: El segundo punto también se puede demostrar tomando los extremos de un camino simple maximal (ver Ejercicio 95c) o por inducción.

TOMADO: 2003C1RX 11-AGO-2003 (parecido), 2009C1R1 19-AGO-2009 (parecido), 2015C2P1 03-OCT-2015.

145. 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) Demostrar que cualquier árbol con 3 o más vértices tiene al menos un punto de corte.

(b) Exhibir una familia infinita de grafos tal que el n-ésimo grafo tenga n vértices y $n - 2$ puntos de corte ($n \geq 2$). Justificar.

(c) Exhibir una familia infinita de árboles tal que el n-ésimo árbol tenga n vértices y exactamente un punto de corte ($n \geq 3$). Justificar.

(d) Exhibir una familia infinita de grafos conexos tal que el n-ésimo grafo tenga n vértices, ningún punto de corte, y la mínima cantidad posible de ejes. Justificar.

SOLUCIÓN:

(a) Al ser $n \geq 3$, es $n \geq 2$, de modo que el árbol tiene al menos 2 hojas, que no pueden ser adyacentes. Cualquier vértice en el único camino que las une es un punto de corte.
(b) P_n.
(c) $K_{1,n-1}$.
(d) K_1, K_2 y C_n ($n \geq 3$). Para $n \geq 3$ no puede ser un árbol por el primer punto, y al ser conexo debe tener ciclos.

TOMADO: 2009C1R1 19-AGO-2009 (con un punto adicional).

146. 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 y sea v un vértice de G. Sea T el árbol que se obtiene al hacer un recorrido DFS de G comenzando en el vértice v. Demostrar que v es punto de corte de G si y sólo si tiene grado al menos 2 en T.
(b) ¿Sigue valiendo la propiedad del punto anterior si se usa BFS en vez de DFS? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Para la ida, si v es punto de corte, hay un par de adyacentes a v tal que todos los caminos entre ellos pasan por v; por lo tanto, al explorar el subárbol de uno de los adyacentes no es posible llegar al otro, y v resulta con grado al menos 2 en T. Para la vuelta, si v tiene grado al menos 2, significa que no es posible llegar de un adyacente en T a otro sin pasar por v, porque el funcionamiento de DFS haría que el otro adyacente se explorara antes de volver a la raíz v.
(b) No vale. Por ejemplo para $G = C_3$, con v cualquiera, queda con grado 2 en T y no es punto de corte.

NOTA: Propuesto originalmente por Pablo Heiber.

TOMADO: 2010C2R1 15-DIC-2010.

147. Sea T un árbol

Variante 1: no trivial con $t \geq 0$

Variante 2: con $t > 0$

vértices de grado al menos 3. Demostrar que T tiene al menos $t + 2$ hojas.

SOLUCIÓN: Sea h la cantidad de hojas. En la Variante 2 el árbol tampoco es trivial porque tiene al menos un vértice de grado al menos 3. Dado que T no es trivial, no tiene vértices de grado 0, por lo que tienen grado 2 los $n - t - h$ vértices que no tienen grado al menos 3 ni son hojas. En consecuencia, $2(n-1) = 2m = \sum d(v) \geq 3t+h+2(n-t-h)$, es decir, $2(n-1) \geq 3t+h+2(n-t-h)$, lo que equivale a $h \geq t + 2$.

Otra solución es por inducción en t. Para $t = 0$ sabemos por el Ejercicio 6.5 de la Práctica que todo árbol no trivial tiene al menos 2 hojas. Para $t = 1$ el árbol es una estrella de al menos 3 rayos de longitudes arbitrarias, que tiene al menos 3 hojas. Para el paso inductivo, se toma un árbol con $t > 1$ nodos de grado al menos 3, y aparentemente se puede demostrar que existe un nodo de grado al menos 3 que tiene colgados todos rayos salvo un camino que va a parar a otro nodo de grado al menos 3. Si se eliminan todos los rayos, el nodo se vuelve hoja, y ya no tiene grado al menos 3. Resulta un árbol con $t - 1$ nodos de grado al menos 3, que por hipótesis inductiva tiene al menos $t + 1$ hojas. El árbol original tenía una hoja menos por el nodo de grado al menos 3, pero a la vez al menos 2 hojas más por los rayos eliminados.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2009C1P1 16-MAY-2009.

148. Una correspondencia de un grafo es un subconjunto de ejes que no comparten extremos. Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia.

Demostrar que un

Variante 1: árbol

Variante 2: bosque

tiene a lo sumo una correspondencia perfecta.

SOLUCIÓN: Se demuestra por inducción sacándole al árbol (o bosque) una de sus hojas, y el vértice al cual la hoja es adyacente. El eje que los une debe pertenecer a cualquier correspondencia perfecta. Al sacar ese eje de la correspondencia, los ejes que quedan forman una correspondencia perfecta (única) en el grafo sin los dos nodos. Es decir, cualquier correspondencia perfecta de G tiene un eje obligatorio, y para el resto tampoco hay alternativas, de modo que es única.

La propiedad también vale para grafos sin ciclos en general (es decir, bosques), ya que vale para cada uno de los árboles que lo componen. En este caso la inducción parece más sencilla, porque al sacar el par de vértices, el árbol puede volverse un bosque, mientras que el bosque lo sigue siendo.

TOMADO: 2001C2P1 20-OCT-2001 (Variante 1), 2001C2RX 29-DIC-2001 (Variante 2).

149. Demostrar que si T es un árbol no trivial de n vértices entonces la cantidad de hojas que tiene es $1 + \sum_{i=1}^{n} |d_i - 2|/2$, donde d_i es el grado del i-ésimo vértice de T.

SOLUCIÓN: Se demuestra por inducción sacándole a T una de sus hojas v, y separando en el caso en que el vértice que era adyacente a v se vuelve una hoja al sacar v, o no se vuelve una hoja al sacar v. Otra forma más elegante de demostrarlo es separando la sumatoria en vértices que son hojas ($d_i = 1$), y vértices que no lo son ($d_i \geq 2$, porque el árbol es no trivial).

TOMADO: 2001C2R1 19-DIC-2001, 2001C2RX 29-DIC-2001, 2005C2R1 15-DIC-2005, 2018C2P1 03-OCT-2018.

150. Sean $d_1, d_2, \dots d_n$ enteros positivos tales que $\sum_{i=1}^{n} d_i = 2(n-1)$, con $n \geq 2$. Demostrar que existe un árbol con n vértices tales que sus grados son $d_1, d_2, \dots d_n$.

SUGERENCIA: Inducción en n. No confundir con el Ejercicio 6.8 de la Práctica.

SOLUCIÓN: El caso base ($n = 2$) es trivial. Para el paso inductivo, puede demostrarse que existe un $d_i = 1$ y otro $d_j > 1$; eliminar el $d_i = 1$, y reemplazar d_j por $d_j - 1$; se obtienen $n - 1$ enteros positivos, cuya suma es $2(n-2)$; por hipótesis inductiva con eso se puede armar un árbol, al que luego se le agrega una hoja adyacente al vértice que tiene grado $d_j - 1$.

NOTA: Basado en 2.26 de Manber. El hecho de que $n \geq 2$ en realidad se deduce de que los d_i son enteros positivos cuya suma es $2(n-1)$.

TOMADO: 2005C1P1 21-MAY-2005.

151. Sea U un conjunto de nodos. Para $i = 1, 2, \dots k$, sea $G_i = (V_i, E_i)$ un grafo tal que $V_i \subseteq U$. Decimos que $G = (V, E)$ es la intersección de los grafos G_i si y sólo si

$$\begin{aligned} V &= \cap_{i=1}^{k} V_i \ , \\ E &= \cap_{i=1}^{k} E_i \ . \end{aligned}$$

Cuando $V = \emptyset$ decimos que G es nulo.

Sea T un árbol. Sean T_1, T_2 y T_3 subárboles (subgrafos que son árboles) de T.

(a) Demostrar que si $T_1 \cap T_2$ no es nulo, entonces es un subárbol de T.

(b) Demostrar que si $T_i \cap T_j$ no es nulo ($i, j \in \{1, 2, 3\}$; $i \neq j$), entonces $T_1 \cap T_2 \cap T_3$ no es nulo.

SOLUCIÓN:

(a) Alcanza con ver que $T_1 \cap T_2$ es conexo. Si $T_1 \cap T_2$ tiene un solo vértice, no hay nada que demostrar. Si tiene más de uno, tomemos dos cualesquiera de ellos. Consideremos el camino entre ambos vértices en T_1, y análogamente en T_2. Si ambos caminos son el mismo camino de T, como el camino está tanto en T_1 como en T_2, entonces está en $T_1 \cap T_2$. Si ambos caminos fueran distintos, habría en T dos caminos simples distintos entre el mismo par de vértices, lo que implica que T tendría un ciclo por el Ejercicio 5.8.a de la Práctica.

(b) Puede demostrarse con el Ejercicio 152, o como sigue.
Sea v_{12} un vértice en $T_1 \cap T_2$, v_{13} un vértice en $T_1 \cap T_3$, y v_{23} un vértice en $T_2 \cap T_3$. Sea c_1 el único camino en T_1 entre v_{12} y v_{13}, c_2 el único camino en T_2 entre v_{12} y v_{23}, y c_3 el único camino en T_3 entre v_{23} y v_{13}. Todos estos caminos son caminos simples. Consideremos el camino (no necesariamente simple) que surge de concatenar c_2 con c_3. Este camino va de v_{12} a v_{13} (al igual que c_1). Dicho camino contiene un camino simple entre v_{12} y v_{13}, el cual tiene un nodo intermedio que está tanto en T_2 como en T_3. Si este camino coincide con c_1, ese nodo intermedio está en $T_1 \cap T_2 \cap T_3$. Caso contrario, habría dos caminos simples distintos entre v_{12} y v_{13}, lo que implica que T tendría un ciclo por el Ejercicio 5.8.a de la Práctica.

TOMADO: 2002C1R1 17-JUL-2002.

152. Sea T un árbol. Sea F una familia de subárboles (subgrafos que son árboles) de T, tal que cualquier par de elementos de F tienen al menos un nodo en común. Demostrar que existe al menos un nodo que está en todos los elementos de F.

SOLUCIÓN: Por inducción en la cantidad de elementos de F. Para menos de 3 subárboles, es trivial. Para 3 o más, es como sigue. Sean $T_1, T_2, \ldots T_k$ los subárboles. Por el Ejercicio 151b sabemos que $T_i \cap T_{k-1} \cap T_k$ no es nulo para cualquier $i \leq k-2$. Por el Ejercicio 151a sabemos que $T_{k-1} \cap T_k$ es un árbol. Podemos entonces aplicar la hipótesis inductiva a $F' = (F - \{T_{k-1}, T_k\}) \cup \{T_{k-1} \cap T_k\}$.

NOTA: Propuesto originalmente por Flavia Bonomo. La solución de ella considera una familia minimal que no cumple la propiedad, y luego hace algo parecido al Ejercicio 151b.

TOMADO: 2002C1R1 17-JUL-2002.

153. Dado un grafo $G = (V, E)$, la matriz de alcance $R(G)$ del grafo es una matriz binaria tal que $R_{ij}(G) = 1$ si y sólo si existe un camino que sale del i-ésimo vértice de G y llega al j-ésimo vértice de G. Diseñar un algoritmo eficiente que dado un grafo no dirigido, calcule la matriz de alcance del grafo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n^2)$, donde $n = |V|$.

SOLUCIÓN: Componentes conexas.

TOMADO: 2001C2R1 19-DIC-2001.

154. Un puente de un grafo es un eje del mismo tal que al removerlo se obtiene un grafo con más componentes conexas.

Sea G un grafo conexo y e uno de sus ejes. Demostrar que e es un puente de G si y sólo si e pertenece a todo árbol generador de G.

SOLUCIÓN: Para la ida, la única forma de que los extremos de e pertenezcan al mismo árbol generador es que e pertenezca al árbol. Para la vuelta, si e no es un puente, podrímos removerlo y obtener un grafo que seguiría siendo conexo; si calculamos un árbol generador de ese grafo, no contendría a e y también sería árbol generador de G, lo cual es absurdo.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2012C1R1 18-JUL-2012.

155. Un puente de un grafo es un eje del mismo tal que al removerlo se obtiene un grafo con más componentes conexas.

Sea G un grafo de n vértices.

(a) Demostrar que G no tiene ciclos simples formados únicamente por puentes.

(b) Demostrar que G tiene a lo sumo $n-1$ puentes.

(c) Demostrar que G tiene $n-1$ puentes si y sólo si G es un árbol.
SUGERENCIA: Para la ida demostrar primero que los puentes forman un árbol.
SUGERENCIA: Para la ida demostrar primero que G es conexo.

(d) Demostrar que todo eje de G es un puente si y sólo si G es un bosque.

(e) Exhibir una familia infinita de grafos tal que el n-ésimo grafo tenga n vértices y $n-1$ puentes. Justificar.

SOLUCIÓN:

(a) Si los tuviera, al remover cualquier eje de cualquiera de esos ciclos, los caminos que usaban a ese eje podrían reconstruirse con el resto del ciclo, lo que contradice el hecho de que ese eje era un puente.

(b) Si hubiera n, formarían al menos un ciclo, lo cual no puede ocurrir por el punto anterior.

(c) Para la vuelta, un árbol es conexo y al remover cualquiera de sus ejes queda un grafo no conexo, de modo que todos los ejes son puentes. Como la cantidad de ejes de un grafo de n vértices es $n-1$, el árbol tiene esa cantidad de puentes. Para la ida, los puentes son $n-1$ y no forman ciclos por el primer punto, de modo que forman un árbol. Si G tuviera otros ejes además de los $n-1$ puentes, cada eje adicional formaría ciclo con algunos de los puentes, que por lo tanto no serían puentes. Otra forma es notar que G es conexo porque caso contrario en cada componente conexa de k nodos habría a lo sumo $k-1$ puentes por el punto anterior, y en total no serían $n-1$. Luego se puede argumentar de la misma manaera que antes que G no tiene otros ejes además de los puentes.

(d) Es el Ejercicio 6.6 de la Práctica. Para la ida, si G tuviera ciclos los ejes de los mismos no serían puentes, ya que al remover cualquiera de ellos los caminos que lo usaban podrían reconstruirse usando el resto del ciclo. Para la vuelta, al remover cualquier eje no es posible que queden menos componentes conexas, y efectivamente quedan más porque los extremos del eje quedan en componentes conexas distintas, ya que si hubiera un camino alternativo fomaría un ciclo con el eje removido.

(e) Por el tercer punto basta elegir cualquier árbol, por ejemplo P_n.

NOTA: Encontrado originalmente en Internet. Es la versión no dirigida y más general del Ejercicio 156.

TOMADO: 2014C1P1 17-MAY-2014 (excepto el primer y último puntos).

156. Sea G un digrafo de n vértices y m ejes.

(a) Decimos que un vértice v de G es especial si y sólo si para todo vértice w de G hay un camino dirigido de v a w y hay un camino dirigido de w a v.

Demostrar que son equivalentes:

i. G es fuertemente conexo.

ii. *Todo* vértice v de G es especial.

iii. *Algún* vértice v de G es especial.

(b) Diseñar un algoritmo eficiente basado en el punto anterior que decida si G es fuertemente conexo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(m+n)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

(c) Un puente fuerte de un digrafo fuertemente conexo es un eje del mismo tal que al removerlo se obtiene un digrafo que no es fuertemente conexo.

Demostrar que si G es fuertemente conexo entonces tiene a lo sumo $2(n-1)$ puentes fuertes.

SUGERENCIA: Considerar un vértice v de G y dos árboles generadores de G que tengan a v como raíz: uno con caminos desde v y el otro con caminos hacia v.

(d) Exhibir una familia infinita de digrafos fuertemente conexos tal que el n-ésimo digrafo tenga n vértices y $2(n-1)$ puentes fuertes. Justificar.

SOLUCIÓN:

(a)

(156(a)i) $\Rightarrow$ (156(a)ii): Ver implicación siguiente.

(156(a)i) $\Rightarrow$ (156(a)iii): Sea v cualquier vértice. Por definición de fuertemente conexo, para cualquier par de vértices a y b hay un camino dirigido de a a b. Si primero elegimos $a = v$, y luego $b = v$, el vértice v resulta especial.

(156(a)ii) $\Rightarrow$ (156(a)i): Ver implicación siguiente.

(156(a)iii) $\Rightarrow$ (156(a)i): Sea v un vértice especial, el cual existe por hipótesis. Sean a y b vértices cualesquiera. Como v es especial, existe un camino de a a v y uno de v a b, cuya concatenación es un camino de a a b. En consecuencia el digrafo es fuertemente conexo según la definición.

(156(a)ii) $\Rightarrow$ (156(a)iii): Si todos son especiales, alguno cualquiera lo es.

(156(a)iii) $\Rightarrow$ (156(a)ii): Sea v un vértice especial, el cual existe por hipótesis. Sean v' y w vértices cualesquiera. Como v es especial existe un camino de v' a v y uno de v a w, cuya concatenación es un camino de v' a w. Análogamente, como v es especial existe un camino de w a v y uno de v a v', cuya concatenación es un camino de w a v'. En consecuencia, v' es especial según la definición.

(b) Por el punto anterior basta elegir un vértice v cualquiera y decidir si es especial. Para ello podemos correr BFS o DFS desde v a fin de determinar si todos los vértices son alcanzables desde v, y luego correr BFS o DFS con los ejes invertidos a fin de deterimnar si v es alcanzable desde todos los vértices. Si el digrafo está representado con listas de sucesores, correr BFS o DFS es $O(m+n)$ y la misma complejidad tiene invertir todos los ejes.

(c) La sugerencia se refiere a los dos árboles del punto anterior (considerando los ejes con las orientaciones originales). La unión de ambos árboles es un subgrafo generador fuertemente conexo porque v es especial. Si sacamos del digrafo cualquier eje fuera de la unión, seguimos teniendo un digrafo fuertemente conexo porque ese subgrafo sigue existiendo. Por lo tanto, cualquier puente fuerte debe estar en la unión, la cual tiene a lo sumo $2(n-1)$ ejes porque cada árbol tiene $n-1$ ejes.

(d) Basta elegir un árbol no dirigido de n nodos y remplazar cada eje por dos ejes con direcciones opuestas.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez. Probablemente pueda generalizarse a digrafos con cualquier cantidad de componentes fuertemente conexas, como ocurre en el Ejercicio 155 para grafos con cualquier cantidad de componentes conexas.

157. Sea G un grafo conexo de n vértices y m ejes.

Variante 1: Demostrar que G tiene al menos $m-(n-1)$ ciclos simples distintos.
SUGERENCIA: Considerar un árbol generador de G.

Variante 2: Sea T un árbol generador de G. El único ciclo que se produce al agregar a T un eje de G que no está en T se llama ciclo fundamental de G relativo a T. Demostrar que existen $m-(n-1)$ ciclos fundamentales de G relativos a T.

SOLUCIÓN: Este ejercicio es muy fácil, ya que hay $m-(n-1)$ ejes de G que no están en T.

TOMADO: 2004C1P1 22-MAY-2004 (Variante 1), 2015C2R1 11-DIC-2015 (Variante 1).

158. Sea G un grafo conexo y H un subgrafo de G. Demostrar que H es subgrafo de algún árbol generador de G si y sólo si H es un bosque.

SOLUCIÓN: Para la ida, H no puede tener ciclos porque los tendría el árbol generador del cual es subgrafo.

Para la vuelta, podemos asignar peso 0 a todos los ejes de H y peso 1 a los otros ejes de G (o simplemente ordenarlos de manera tal que aparezcan primero los ejes de H). Si ahora corremos Kruskal van a ser elegidos primero todos los ejes de H (ya que no forman ciclos). Al terminar el algoritmo tendremos un árbol generador de G conteniendo a esos ejes (y eventualmente a otros). Alternativamente, como G es conexo tiene al menos un árbol generador. Sea T un árbol generador de G que tiene la mayor cantidad posible de ejes en común con H. Digo que H es subgrafo de T. Por absurdo, supongamos que no es así. Sea e un eje que está en H pero no en T. Consideremos $T+e$ que tiene un único ciclo C que contiene a e. En C hay un eje f que no está en H, ya que caso contrario H no sería un bosque. Debe ser $e \neq f$ porque e sí está en H. Definamos $T' = T+e-f$ que también es un árbol generador de G, lo cual es absurdo porque tiene más ejes en común con H que los que tiene T.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2007C1P1 19-MAY-2007.

159. Sea G un grafo conexo, C un ciclo simple de G, e y f dos ejes distintos en C. ¿Es cierto que siempre existe un árbol generador T de G tal que...

(a) e y f no están en T?
(b) e está en T pero f no está en T?
(c) e y f están en T?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Falso. Si $G = C_3$, en cualquier árbol generador está al menos un eje de cualquier par de ejes.
(b) Verdadero. Sea T cualquier árbol generador de G. Si e y f no están en T, consideremos $T + e$ que tiene un único ciclo que contiene a e; si ahora sacamos cualquier eje de ese ciclo que no sea e, volvemos a obtener un árbol generador, en el cual sólo está e. Si e y f están en T, consideremos $T - f$, que es un grafo con dos componentes conexas, cada una conteniendo uno de los extremos de f; si ahora recorremos en G el camino $C - f$ desde un extremo de f al otro, en algún momento encontramos un eje que pasa de una componente conexa a la otra de $T - f$, el cual al ser agregado a $T - f$ produce un árbol generador en el que sólo está e.
(c) Verdadero. Sea T_1 un árbol generador que tiene exactamente uno de los dos ejes (digamos e), el cual se puede conseguir como indica el punto anterior. Consideremos $T_1 + f$, el cual tiene un único ciclo que contiene a f. Sea g cualquier eje de ese ciclo que no sea ni e ni f, el cual existe porque cualquier ciclo (simple) tiene al menos tres ejes. En tal caso $T_1 + f - g$ resulta un árbol generador, y tanto e como f están en T.

    Alternativamente, se puede demostrar de manera directa usando el Ejercicio 158, ya que dos ejes distintos no forman un ciclo.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2016C1R1 11-JUL-2016.

160. Un grafo (simple) se dice 1-árbol si es un árbol con un eje agregado.

Sea G un grafo de n vértices. Demostrar que son equivalentes:

(a) G es un 1-árbol.
(b) G es conexo y tiene n ejes.
(c) G es conexo y tiene un único ciclo.
(d) G tiene n ejes y un único ciclo.

SOLUCIÓN:

(160a) $\Rightarrow$ (160b): Como un árbol es conexo y tiene $n - 1$ ejes, al agregarle un eje sigue siendo conexo y tiene n ejes.

(160b) $\Rightarrow$ (160a): G tiene ciclos porque si no los tuviera sería conexo y sin ciclos, es decir un árbol, y entonces tendría $n - 1$ ejes en vez de n. Si elegimos cualquier ciclo de G y sacamos un eje del ciclo, obtenemos un grafo conexo y con $n - 1$ ejes, es decir un árbol, por lo que G era un árbol con un eje agregado.

(160a) $\Rightarrow$ (160c): G es conexo porque un árbol lo es. Respecto de que tiene un único ciclo, vimos en clase que si se agrega un eje a un árbol se obtiene un grafo con un único ciclo conteniendo al eje agregado.

(160c) $\Rightarrow$ (160a): Al sacar un eje del ciclo obtenemos un grafo conexo y sin ciclos, es decir un árbol, por lo que G era un árbol con un eje agregado.

(160c) $\Rightarrow$ (160d): Por lo anterior (160b) $\Leftrightarrow$ (160c), y ambas implican (160d).

(160d) $\Rightarrow$ (160a): Al sacarle un eje del ciclo obtenemos un grafo con $n - 1$ ejes y sin ciclos, es decir un árbol, por lo que G era un árbol con un eje agregado.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2017C2R1 11-DIC-2017.

161. (a) Exhibir todos los árboles no isomorfos que son autocomplementarios. Justificar.

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

SOLUCIÓN:

(a) Debe ser $m = n(n-1)/4 = n-1$, es decir, $n = 1$ o $n = 4$. Los grafos resultan ser K_1 y P_4.

(b) Debe ser $m = n(n-1)/4 = n$, es decir, $m = n = 5$. Los grafos resultan ser C_5 y un triángulo con dos patas.

NOTA: Algo que puede ayudar a resolver el ejercicio es tener en cuenta que al complementar los grados, deben obtenerse los mismos valores (aunque posiblemente en otro orden).

TOMADO: 2013C2P1 12-OCT-2013.

162. Un grafo (simple) se dice 1-árbol si es un árbol con un eje agregado.

Sea G un grafo conexo con pesos asociados a sus ejes, y que tiene al menos un ciclo. Un 1-árbol generador de G es un 1-árbol que es subgrafo generador de G. Un 1-árbol generador mínimo de G es 1-árbol generador de G que tiene peso mínimo en el conjunto de los 1-árboles generadores de G.

Variante 1: Sea T un árbol generador mínimo de G, y sea e un eje de peso mínimo en el conjunto de ejes que están en G pero no en T. Demostrar que $T + e$ es un 1-árbol generador mínimo de G.

SUGERENCIA: Demostrar que el peso de $T + e$ es menor o igual que el peso de cualquier 1-árbol generador de G.

Variante 2: Sea H un 1-árbol generador mínimo de G, y sea f un eje de peso máximo en el único ciclo de H. Demostrar que $H - f$ es un árbol generador mínimo de G.

SUGERENCIA: Demostrar que el peso de $H - f$ es menor o igual que el peso de cualquier árbol generador de G.

SOLUCIÓN:

Variante 1: $T+e$ es 1-árbol por definición, y es subgrafo generador de G porque T lo es y e está en G. Por lo tanto es un 1-árbol generador de G, y para demostrar lo pedido basta seguir la sugerencia. Sea H cualquier 1-árbol generador de G, y sea C el único ciclo de H. Sea f un eje de C que no está en T, el cual existe porque T es un árbol. Notemos que $p(e) \leq p(f)$ por definición de e. Por propiedades vistas en clase sabemos que $H - f$ es un árbol generador de G, de modo que $p(T) \leq p(H - f)$ ya que T es mínimo. Por lo tanto $p(T + e) = p(T) + p(e) \leq p(H - f) + p(f) = p(H)$ lo cual implica que $p(T + e) \leq p(H)$.

Variante 2: Por propiedades vistas en clase sabemos que $H - f$ es un árbol generador de G, de modo que para demostrar lo pedido basta seguir la sugerencia. Sea T cualquier árbol generador de G, y sea C el único ciclo de H. Sea e un eje de C que no está en T, el cual existe porque T es un árbol. Notemos que $p(e) \leq p(f)$ por definición de f. Como en la Variante 1, es fácil comprobar que $T + e$ es un 1-árbol generador de G, de modo que $p(H) \leq p(T + e)$ ya que H es mínimo. Por lo tanto $p(H - f) = p(H) - p(f) \leq p(T + e) - p(e) = p(T)$ lo cual implica que $p(H - f) \leq p(T)$.

NOTA: La Variante 1 provee un algoritmo para determinar un 1-árbol generador mínimo de un grafo. Talvez también funciona un algoritmo parecido al de Kruskal, que en cada iteración elija el eje de menor peso admitiendo a lo sumo un ciclo. Alternativamente, talvez también funciona un algoritmo que comienza con todo el grafo, e iterativamente remueve el eje de mayor peso que deje al menos un ciclo.

TOMADO: 2019C1P1 08-MAY-2019 (Variante 1), 2019C1R1 12-JUL-2019 (Variante 2).

163. Un grafo (simple) se dice 2-árbol si es un árbol con dos ejes agregados.

(a) Demostrar que cualquier 2-árbol tiene 2 o 3 ciclos simples.
SUGERENCIA: Contar por separado los ciclos simples que contienen a exactamente uno de los ejes agregados, y los que contienen a los dos ejes agregados.

(b) Exhibir un 2-árbol que tenga exactamente 2 ciclos simples. Justificar.

(c) Exhibir un 2-árbol que tenga exactamente 3 ciclos simples. Justificar.

SOLUCIÓN:

(a) Sean $e_1 = (v_1, w_1)$ y $e_2 = (v_2, w_2)$ los ejes agregados. Sabemos que hay un único ciclo simple que contiene sólo a e_1, y lo mismo para e_2. Para demostrar lo pedido, alcanza con probar que hay a lo sumo un ciclo simple que contiene tanto a e_1 como a e_2. Por absurdo supongamos que hay al menos dos ciclos simples C y D que contienen a ambos ejes. Notemos que en cada ciclo simple, las porciones que no son los ejes agregados son los únicos caminos simples en el árbol original conectando a determinado extremo de e_1 con determinado extremo de e_2. Hay dos maneras en que los ejes agregados pueden aparecer en un ciclo simple dado. Una es que si empezamos a recorrer el ciclo simple desde v_1 hacia w_1, luego aparezca v_2 antes que w_2, y la otra es que aparezca w_2 antes que v_2. Supongamos sin pérdida de generalidad que C es de la primera manera, es decir, C está formado por e_1, e_2, el único camino simple en el árbol entre v_1 y w_2, y el único camino simple en el árbol entre w_1 y v_2. Si D fuera de la misma manera, entonces sería exactamente igual a C, lo cual estamos suponiendo que no ocurre. Por lo tanto, D es de la segunda manera, es decir, D está formado por e_1, e_2, el único camino simple en el árbol entre v_1 y v_2, y el único camino simple en el árbol entre w_1 y w_2. Los dos caminos simples que tienen por extremo a v_1 no pueden compartir nodos (excepto v_1) porque habría ciclos simples en el árbol. Por lo tanto, su concatenación es un camino simple, el cual conecta v_2 con w_2. Análogamente respecto de los dos caminos simples que tienen por extremo a w_1, cuya concatenación resulta también un camino simple entre v_2 y w_2. Si esos dos caminos simples entre v_2 y w_2 fueran iguales, entonces C o D no serían ciclos simples. Si fueran distintos, como sus extremos son iguales, por el Ejercicio 5.8.a de la Práctica habría un ciclo simple en el árbol.

(b) Dos triángulos unidos por un eje.

(c) C_n junto con una de sus cuerdas ($n \geq 4$ para que tenga cuerdas).

NOTA: Propuesto originalmente por Guido Tagliavini Ponce.

TOMADO: 2016C2R1 12-DIC-2016.

164. Sea G un grafo

Variante 1: conexo. Sea $T(G)$ el grafo que tiene a los árboles generadores de G como vértices, y dos árboles generadores

Variante 2: conexo con pesos asociados a sus ejes. Sea $T(G)$ un grafo que tiene a los árboles generadores mínimos de G como vértices, y dos árboles generadores mínimos

son adyacentes si y sólo si cada uno de ellos tiene exactamente un eje que no tiene el otro. Demostrar que $T(G)$ es conexo.

SOLUCIÓN: Dados dos árboles hay que demostrar que existe un camino que los une. Básicamente es inducción en la cantidad de ejes no coincidentes, como en el Ejercicio 179. Si son el mismo árbol, hay camino. Caso contrario, se toma un eje que esté en uno y no en el otro, se lo agrega al otro, y se saca un eje (que no esté en el primer árbol) del único ciclo que se forma . Se obtiene un par de árboles con menos ejes diferentes que el original. Para la Variante 2 el eje que se agrega debe ser el de peso mínimo del conjunto de los que están en un árbol y no en el otro. Eso permite argumentar que el eje que se saca tiene peso mayor o igual, y finalmente igual, ya que el árbol modificado era mínimo. De modo que el nuevo árbol también es mínimo y por lo tanto está en $T(G)$.

NOTA: Propuesto originalmente por Federico Lebrón (Variante 1).

TOMADO: 2011C2P1 15-OCT-2011 (Variante 1), 2011C2R1 14-DIC-2011 (Variante 2).

165. (a) Sea G un grafo conexo. Sean $T_1 \neq T_2$ dos árboles generadores de G. Demostrar que existen dos ejes e_1 y e_2 tales que e_1 está en T_1 y no en T_2, e_2 está en T_2 y no en T_1, y tanto $T_1 + e_2 - e_1$ como $T_2 + e_1 - e_2$ son árboles generadores de G.

(b) Sea G un grafo conexo con pesos asociados a sus ejes.

Variante 1: Sea $T(G)$ el grafo que tiene a los árboles generadores de G como vértices, y dos árboles generadores son adyacentes si y sólo si cada uno de ellos tiene exactamente un eje que no tiene el otro. A cada vértice de $T(G)$ le asociamos el peso del árbol generador correspondiente de G. Sea T_1 cualquier vértice de $T(G)$, es decir, cualquier árbol generador de

G. Demostrar que T_1 es mínimo local si y sólo si es mínimo global. Es decir, demostrar que T_1 tiene peso menor o igual que todos sus vecinos en $T(G)$ si y sólo si es un árbol generador mínimo de G.

Variante 2: Sea T_1 un árbol generador de G. Demostrar que T_1 no tiene peso mínimo si y sólo si existe un árbol generador de G que tiene peso estrictamente menor que T_1 y que difiere de T_1 en exactamente un eje.

SUGERENCIA: Para la ida tomar un árbol generador mínimo de G que tenga la mayor cantidad posible de ejes en común con T_1, y usar o no el primer punto.

SOLUCIÓN:

(a) Como $T_1 \neq T_2$ existe un eje e_1 que está en T_1 y no en T_2. Sea C_2 el único camino en T_2 entre los extremos de e_1. El subgrafo $T_1 - e_1$ tiene dos componentes conexas, y el camino C_2 va de una a otra en G, de modo que existe un eje e_2 del mismo que va de una componente conexa a la otra, y por lo tanto $T_1 - e_1 + e_2$ es un árbol, que es generador porque tiene todos los vértices de G. El eje e_2 no está en T_1 porque entonces tendría un ciclo. Finalmente, $T_2 + e_1 - e_2$ es un árbol como de costumbre, que es generador al igual que T_1.

(b) La Variante 2 pide las contrarrecíprocas de la Variante 1, así que vamos a ver sólo la Variante 1. La vuelta es trivial. La ida es como sigue. Sea T_2 como dice la sugerencia. Si $T_1 = T_2$, entonces T_1 es árbol generador mínimo, y no hay nada más que decir. Caso contrario, es como sigue. Usando el primer punto, sean e_1 y e_2 como dice ese punto; como T_1 es mínimo local su peso es menor o igual que el de $T_1 + e_2 - e_1$, lo que equivale a decir que el peso de e_2 es mayor o igual que el peso de e_1; como T_2 es mínimo global su peso es menor o igual que el de $T_2 + e_1 - e_2$, lo que equivale a decir que el peso de e_1 es mayor o igual que el peso de e_2; en consecuencia, e_1 y e_2 tienen el mismo peso, por lo que $T_2 + e_1 - e_2$ es un árbol generador mínimo, lo cual es absurdo porque tiene más ejes en común con T_1 que los que tiene T_2. Sin usar el primer punto, sea D el conjunto de ejes de peso mínimo entre aquellos que están en T_1 y no en T_2, o viceversa; si D contiene algún eje e_1 que está en T_1, podemos agregarlo a T_2 y sacar del único ciclo que se forma un eje e_2 que está en T_2 pero no en T_1, obteniendo un árbol generador $T_2 + e_1 - e_2$; por definición de D el peso de e_1 es menor o igual que el peso de e_2; como T_2 es árbol generador mínimo el peso de e_2 es menor o igual que el peso de e_1; en consecuencia, e_1 y e_2 tienen el mismo peso, lo cual es absurdo como en la primera demostración; por lo tanto, D no puede contener ejes de T_1; sea entonces e_2 un eje de D, el cual debe estar en T_2; podemos agregar e_2 a T_1 y sacar del único ciclo que se forma un eje e_1 que está en T_1 pero no en T_2, obteniendo un árbol generador $T_1 + e_2 - e_1$; por definición de D el peso de e_2 es menor o igual que el peso de e_1, pero no pueden ser iguales porque entonces e_1 estaría en D; en consecuencia el peso de e_2 es menor que el de e_1, lo cual es absurdo porque T_1 es mínimo local.

NOTA: Propuesto originalmente por Federico Lebrón (segundo punto, Variante 1).

TOMADO: 2013C1P1 18-MAY-2013 (Variante 1), 2017C1R1 14-JUL-2017 (Variante 2).

166. Sea G un grafo conexo con pesos asociados a sus ejes. Demostrar que G tiene dos árboles generadores de distinto peso total si y sólo si tiene un ciclo simple en el cual hay dos ejes de distinto peso.

SUGERENCIA: Para la ida, considerar dos árboles generadores de G de distinto peso total y que tengan la mayor cantidad posible de ejes en común.

SOLUCIÓN: Para la ida, sean T_1 y T_2 dos árboles generadores de distinto peso con la mayor cantidad posible de ejes en común (no todos, porque en tal caso no tendrían distinto peso). Sea e_1 un eje que está en T_1 y no en T_2. Consideremos el grafo $T_2 + e_1$ el cual tiene un único ciclo. Sea e_2 cualquier eje que está en ese ciclo pero no en T_1 (existe porque caso contrario T_1 tendría un ciclo). Si e_1 y e_2 tienen distinto peso, listo, encontramos el ciclo. Si tienen el mismo peso, sea $T_3 = T_2 + e_1 - e_2$, que resulta un árbol generador de G de igual peso que T_2, y por lo tanto de distinto peso que T_1. Además, tiene un eje más en común con T_1 que lo que tenía T_2, lo cual es absurdo.

Para la vuelta, sea C un ciclo simple con dos ejes e_1 y e_2 de distinto peso. Se pueden ir sacando ejes de los ciclos de G que no sean C, hasta que C sea el úniclo ciclo. En ese momento, si se saca e_1 queda un árbol de cierto peso, y en cambio si se sace e_2 queda un árbol de otro peso.

Alternativamente, según el Ejercicio 158 existe un árbol generador T de G que tiene a todos los ejes de $H = C - e_2$ (y por lo tanto a e_1 pero no a e_2). Consideremos $T + e_2$ que tiene un único ciclo. Ese ciclo no puede ser otro que C, ya que T tiene a todos los ejes de $C - e_2$. Si ahora eliminamos e_1 obtenemos $T + e_2 - e_1$, que vuelve a ser un árbol generador, y tiene distinto peso que T ya que e_1 y e_2 tienen distinto peso.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

TOMADO: 2016C1P1 07-MAY-2016.

167. Sea $G = (V, E)$ un grafo conexo con pesos asociados a sus ejes. Dada cualquier partición de V en (V_1, V_2), se dice que un eje es liviano respecto de esa partición si y sólo si tiene peso mínimo dentro del conjunto de ejes que tienen un extremo en V_1 y otro en V_2. Este concepto está bien definido ya que al ser G conexo, existe al menos un eje entre V_1 y V_2.

    (a) Sea e cualquier eje de G. Demostrar que e está en *algún* árbol generador mínimo de G si y sólo si existe una partición de V tal que e es *liviano* respecto de ella.

    (b) Sea e cualquier eje de G. Demostrar que e está en *todo* árbol generador mínimo de G si y sólo si existe una partición de V tal que e es *el único eje liviano* respecto de ella.

    (c) Demostrar que si para toda partición de V existe un único eje liviano respecto de ella, entonces G tiene un único árbol generador mínimo. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

    SOLUCIÓN:

    (a) Sean v_1 y v_2 los extremos de e.

    Para la ida, sea T un árbol generador mínimo de G que contiene a e. Sean T_1 y T_2 los dos árboles que forman $T - e$, con v_i en T_i, y sea V_i el conjunto de nodos de T_i ($i \in \{1, 2\}$). El eje e debe ser liviano respecto de esa partición, ya que si no lo fuera, habría otro eje f entre V_1 y V_2 que sería liviano, lo que implica que $T - e + f$ tendía menor peso que T.

    Para la vuelta, sea T cualquier árbol generador mínimo de G. Si e está en T, listo. Caso contrario, sea (V_1, V_2) una partición de V tal que e es liviano respecto de ella, con $v_i \in V_i$ ($i \in \{1, 2\}$). Sea C el único camino en T que va de v_1 a v_2, el cual junto con e forma un ciclo en G. Dicho camino debe contener al menos un eje f que va de V_1 a V_2. Si tomamos $T + e - f$ resulta un árbol generador de G de peso menor o igual que T, ya que e es liviano. Como este nuevo árbol no pude tener peso menor que T, su peso resulta igual, y por lo tanto también es árbol generador mínimo de G. Además, contiene a e.

    (b) Sean v_1 y v_2 los extremos de e.

    Para la ida, la partición es la de la ida del punto anterior. Por el punto anterior ya sabemos que e es liviano respecto de esa partición. Si hubiera otro eje liviano f, entonces $T - e + f$ sería un árbol generador mínimo que no contendría a e, lo cual es absurdo por hipótesis.

    Para la vuelta, supongamos por absurdo que existe T un árbol generador mínimo de G que no contiene a e. Consideremos la misma partición que en la vuelta del punto anterior. Nuevamente $T + e - f$ resulta un árbol generador mínimo de G, lo cual es absurdo porque en tal caso f también sería liviano.

    (c) Sea T cualquier árbol generador mínimo de G, y sea e cualquiera de sus ejes. Por la ida del primer punto existe una partición de V tal que e es liviano respecto de ella. Pero ahora por hipótesis, e es el único eje liviano respecto de la partición. En consecuencia, por la vuelta del segundo punto, e está en todo árbol generador mínimo de G. Como eso vale para todos los ejes de T, todos los árboles generadores mínimos de G deben ser iguales a T, que resulta único.

    La recṕroca es falsa. Basta tomar C_3 con dos ejes de un peso y el tercero con peso mayor.

    NOTA: Propuesto originalmente por Michel Mizrahi (tercer punto).

    TOMADO: 2012C2P1 13-OCT-2012 (dos primeros puntos), 2012C2R1 12-DIC-2012.

168. Sea G un grafo conexo con pesos asociados a sus ejes. Sean v y w dos vértices distintos de G. Decimos que un camino entre v y w en G es min-max si minimiza el peso del eje más pesado del camino (sobre el conjunto de todos los caminos entre v y w que hay en G). Sea T un árbol

generador mínimo de G, y sea p el único camino entre v y w que hay en T. Demostrar que p es un camino min-max entre v y w en G. Es decir, demostrar que para todo camino p' entre v y w que haya en G, el peso del eje más pesado de p es menor o igual que el peso del eje más pesado de p'.

SUGERENCIA: Por absurdo.

SOLUCIÓN: Sea e el eje más pesado de p. Si se elimina e de T, resulta un grafo con dos componentes conexas, una conteniendo a v y otra a w. Si suponemos que p no es min-max, existe otro camino p' entre v y w en G cuyo eje más pesado pesa menos que e, lo que implica que todos sus ejes pesan menos que e. Existe un eje e' dentro de C' que pasa de una componente de $T - e$ a la otra. Si se agrega e' a $T - e$ se obtiene un árbol menos pesado que T, lo cual es absurdo.

NOTA: Propuesto originalmente por Flavia Bonomo.

TOMADO: 2005C1R1 22-JUL-2005.

169. (a) Sea G un grafo conexo con pesos no negativos asociados a sus ejes. Sea T un árbol generador mínimo de G, y sea $e = (u, v)$ un eje de T. Demostrar que e es un

Variante 1: camino mínimo

Variante 2: camino simple mínimo

en G entre u y v.

(b) ¿Sigue valiendo la propiedad del punto anterior si los pesos pueden ser negativos? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Es trivial comprobar que e es un camino (simple). Sea P cualquier camino (para la Variante 2, simple) en G entre u y v. Queremos demostrar que el peso de e es menor o igual que el peso de P.

Sea e' un eje de peso máximo en P. Por el Ejercicio 168 el peso de e es menor o igual que el peso de e', el cual a su vez es menor o igual que el peso de P porque los pesos son no negativos.

Alternativamente, se puede demostrar con argumentos similares a los del Ejercicio 168. Concretamente, se puede eliminar e de T y agregar un eje e' de P que va de una componente conexa de $T - e$ a la otra, lo cual forma el árbol generador $T' = T - e + e'$. Como T es mínimo, su peso es menor o igual que el peso de T'. Eso implica que el peso de e es menor o igual que el peso de e', el cual a su vez es menor o igual que el peso de P porque los pesos son no negativos.

Alternativamente, sea G' el grafo que se obtiene al eliminar e de T y luego agregar los ejes de P que no están en $T - e$. Este grafo G' es subgrafo de G ya que T y P lo son, es generador porque T lo es, y es conexo porque al menos un eje de P va de una componente conexa de $T - e$ a la otra. Sea T' un árbol generador de G', el cual también es árbol generador de G porque G' es subgrafo generador de G. Como T es mínimo, su peso es menor o igual que el peso de T'. El peso de T' es menor o igual que el peso de G' porque los pesos son negativos. Por el mismo motivo el peso de G' es menor o igual que el peso de $T - e + P$, lo cual implica que el peso de e es menor o igual que el peso de P.

(b) No vale.

Para la Variante 1 basta que exista un eje de peso negativo para que no exista camino mínimo entre u y v. Esto es así porque tal eje sería alcanzable desde u y desde v en G conexo, y podría rocorrerse un número arbitrariamente grande de veces en un camino entre u y v que no tiene la restricción de ser simple.

Para la Variante 2 lo anterior no sirve y es necesario buscar un contraejemplo con más cuidado. Una posibilidad es $G = C_4$ con dos ejes no adyacentes de peso 1, y los otros dos de peso -1. Cualquier árbol generador mínimo contiene un eje de peso 1, pero el camino simple mínimo entre sus extremos es el resto del ciclo, el cual tiene peso -1.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2018C1P1 09-MAY-2018 (Variante 1).

170. Sea G un grafo conexo con pesos asociados a sus ejes. Sea $T_{\min}$ un árbol generador mínimo de G. Sea T un árbol generador de G. Demostrar que el peso máximo de los ejes de $T_{\min}$, es menor o igual que el peso máximo de los ejes de T.

 SUGERENCIA: Eliminar un eje de peso máximo de $T_{\min}$.

 SOLUCIÓN: Al eliminar un eje de peso máximo de $T_{\min}$, resulta un grafo con dos componentes conexas. Elijamos un nodo de cada una de ellas. En T existe un camino entre ambos, que en algún momento pasa de una componente a la otra. Es decir, existe en T un eje que conecta ambas componentes conexas entre sí. Si agregamos uno de esos ejes a las componentes conexas, volvemos a obtener un árbol. Si el peso máximo en $T_{\min}$ fuera mayor que el peso máximo en T, el árbol obtenido pesaría menos que $T_{\min}$, lo cual es absurdo.

 TOMADO: 2005C2P1 15-OCT-2005, 2015C1R1 13-JUL-2015.

171. Sea G un grafo conexo con pesos asociados a sus ejes. Sea C cualquier circuito simple de G, y sea e cualquier eje de peso máximo de C. Sea $G_{-e} = G - e$ el grafo que se obtiene al eliminar el eje e de G. Notar que G_{-e} es siempre conexo. Sea T cualquier árbol generador mínimo de G, y sea T_{-e} cualquier árbol generador mínimo de G_{-e}.

 (a) ¿Es cierto que T_{-e} es árbol generador mínimo de G?
 SUGERENCIA: Si e está en T, eliminarlo y luego obtener un nuevo árbol generador cuyo peso está entre el de T_{-e} y el de T.

 (b) ¿Es cierto que T es árbol generador mínimo de G_{-e}?
 SUGERENCIA: No es cierto.

 En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

 SOLUCIÓN:

 (a) Basta ver que T_{-e} es árbol generador de G y que es mínimo. Lo primero es inmediato, porque G y G_{-e} tienen los mismos nodos. Para lo segundo, basta ver que el peso de T_{-e} es menor o igual que el peso de un árbol generador mínimo de G, por ejemplo T. Si T no contiene a e, es un árbol generador de G_{-e}, y al ser T_{-e} un árbol generador mínimo de G_{-e}, su peso es menor o igual al de T. Si T contiene a e, $T - e$ es un bosque con dos componentes conexas; el camino $C - e$ une en G los extremos de e, por lo que en algún momento pasa de una a la otra; si agregamos a $T - e$ cualquier eje e' que lo hace, obtenemos un nuevo árbol $T' = T - e + e'$ que es árbol generador tanto de G_{-e} como de G; al ser T_{-e} un árbol generador mínimo de G_{-e}, su peso es menor o igual al de T', y este es a su vez menor o igual al de T ya que el peso de e' es menor o igual al peso de e, por tener e peso máximo en C.

 (b) No es cierto cuando e pertenece a T, lo cual puede ocurrir si en C todos los ejes tienen el mismo peso.

 NOTA: Propuesto originalmente por Martiniano Eguía.

 TOMADO: 2010C2R1 15-DIC-2010.

172. Sea G un grafo conexo con pesos asociados a sus ejes, y sea T un árbol generador mínimo de G.

 (a) Sea e un eje de G que no está en T, y sea p el único camino entre los extremos de e que hay en T. Demostrar que el peso de e es mayor o igual que el peso de cualquier eje de p.

 (b) Demostrar que si los pesos de los ejes de G son todos distintos, ningún eje de T que pertenezca a un ciclo en G, puede ser el eje de mayor peso entre los ejes del ciclo.

 SOLUCIÓN:

 (a) Si el peso de e fuera menor que el de algún eje e_p de p, podríamos agregar e a T, sacarle e_p, y obtendríamos un árbol generador de menor peso que T.

 (b) Demostración incorrecta: Sea e' un eje de T que pertenece a un ciclo en G. Sea e un eje del ciclo que no está en T (no pueden estar en T todos los ejes del ciclo). Si e' fuera el eje de mayor peso del ciclo, podríamos agregar e a T, sacarle e', y obtendríamos un árbol generador de menor peso que T, debido a que al ser todos los pesos distintos, el peso de e es menor que el de e'. La demostración es incorrecta debido a que $T + e - e'$ puede no ser conexo, si al agregar e se forma un ciclo que no contiene a e', lo cual puede ocurrir si e y e' están en más de un ciclo.

TOMADO: 2001C2RX 29-DIC-2001, 2003C1R1 18-JUL-2003.

173. Sea G un grafo conexo con pesos asociados a sus ejes. Sea T un árbol generador mínimo de G. Sea e un eje de T. Sea C un camino simple en G entre los extremos de e.

(a) Demostrar que si los pesos son no negativos, entonces el peso total de C es mayor o igual que el peso de e.

(b) ¿Sigue valiendo la propiedad del punto anterior si los pesos pueden ser negativos? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Por absurdo, supongamos que el peso de C es menor que el peso de e. Si eliminamos e de T obtenemos un bosque con exactamente dos componentes conexas. Si luego recorremos C desde un extremo de e al otro, en algún momento pasamos de una componente conexa a la otra con algún eje f cuyo peso es menor o igual que el de C ya que los (otros) ejes de C tiene peso no negativo. Por lo tanto, el peso de f es menor que el peso de e, lo cual es absurdo porque $T - e + f$ sería un árbol generador con peso menor que T.

(b) No vale. Se puede buscar un ejemplo a mano. Alternativamente, se puede usar que los AGMs no varían si se suma (o resta) un mismo valor a todos los pesos. En consecuencia, el peso de cualquier camino de al menos dos ejes se puede hacer arbitrariamente menor que el peso de cualquier eje de un AGM, si se resta a todos los pesos un valor suficientemente grande (ya que el eje disminuye una vez y el camino tantas veces como ejes tiene). Por ejemplo, si G es un triángulo con pesos 1, 1, y 2, el único AGM está formado por los dos ejes de peso 1. Si ahora restamos 10 (por decir algo) a todos los ejes, los pesos se vuelven -9, -9 y -8. El único AGM está formado por los dos ejes de peso -9, pero si tomamos cualquiera de ellos, el camino alternativo tiene peso total -17.

Se pide camino simple porque caso contrario es muy fácil encontrar un contraejemplo si los pesos pueden ser negativos: basta elegir cualquier eje negativo e y que C lo recorrar un número impar de veces (al menos 3).

TOMADO: 2013C1R1 17-JUL-2013.

174. Sea G un grafo conexo con pesos asociados a sus ejes. Demostrar que si G es un árbol o todos sus ejes tienen el mismo peso, entonces todos los árboles generadores de G tienen el mismo peso total. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN: Si G es un árbol, existe un único árbol generador, que a la vez es mínimo y máximo; si todos los ejes son del mismo peso, todos los árboles generadores tienen el mismo peso. La recíproca es falsa; un contraejemplo es un grafo formado por dos ciclos con todos los ejes del mismo peso, unidos entre sí por un eje con otro peso.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2005C1P1 21-MAY-2005.

175. Dado un grafo conexo G con pesos asociados a sus ejes, un árbol generador máximo de G es un árbol generador de G que tiene peso máximo entre todos los árboles generadores de G.

Sea G un grafo conexo en el cual cada eje e tiene asociado un peso no necesariamente positivo $p(e) \in \mathbb{R}$. Dada una función $f : \mathbb{R} \to \mathbb{R}$, definimos G_f como el grafo que tiene los mismos vértices y ejes que G, pero en el cual cada eje e tiene asociado el peso $f(p(e))$ en vez de $p(e)$. Sea T un árbol generador de G, el cual por definición también lo es de G_f, y cuyo peso total en cada grafo es la suma de los pesos asociados a sus ejes en ese grafo. ¿Es cierto que...

(a) para $f(x) = x + b$, T es un árbol generador *mínimo* de G si y sólo si T es un árbol generador *mínimo* de G_f?

(b) para $f(x) = ax$ con $a < 0$, T es un árbol generador *mínimo* de G si y sólo si T es un árbol generador *máximo* de G_f?

(c) para $f(x) = ax$ con $a > 0$, T es un árbol generador *mínimo* de G si y sólo si T es un árbol generador *mínimo* de G_f?

(d) para $f(x) = ax$ con $a > 0$, T es un árbol generador *máximo* de G si y sólo si T es un árbol generador *máximo* de G_f?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN: Todos los puntos son verdaderos. Para demostrar cada uno se puede tomar cualquier árbol generador y probar que T es óptimo con determinados pesos, usando que lo es con los otros pesos. Los dos últimos puntos también se pueden demostrar usando que el segundo punto es cierto y que $ax = (-1)(-a)x$ con -1 y $-a$ menores que 0.

NOTA: Es similar al Ejercicio 211 pero de árboles generadores.

TOMADO: 2001C1R1 17-JUL-2001 (primer punto), 2015C1P1 16-MAY-2015.

176. (a) Dado un grafo conexo G con pesos asociados a sus ejes, un árbol generador máximo de G es un árbol generador de G que tiene peso máximo entre todos los árboles generadores de G.

Sea G un grafo conexo con pesos asociados a sus ejes. Sea T un árbol generador de G. Sea G' el grafo que se obtiene multiplicando por -1 el peso de cada eje de G. Sea T' el grafo que se obtiene multiplicando por -1 el peso de cada eje de T. Demostrar que T es un árbol generador mínimo de G si y sólo si T' es un árbol generador máximo de G'.

(b) La ciudad es muy compleja, y en muchos casos existe más de una secuencia de cuadras que permite ir de una esquina a otra dada. Como mantener transitables las cuadras requiere un gasto reiterado, el intendente ha decidido destruir las cuadras superfluas. Destruir cada cuadra tiene un costo asociado, que depende de su longitud, indemnizaciones a pagar, y otros factores. El intendente quiere destruir tantas cuadras como sea posible hasta que para cada par de esquinas distintas exista una única manera de ir de una a otra sin repetir esquinas, gastando en la destrucción lo mínimo necesario.

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 en ambos sentidos. Se puede ir de cualquier esquina a cualquier otra recorriendo las cuadras, pasando eventualmente por esquinas intermedias.

Diseñar un algoritmo eficiente que determine cuáles cuadras destruir para cumplir los deseos del intendente. La entrada del algoritmo es la cantidad n de esquinas, la cantidad m de cuadras, y para cada cuadra las esquinas que conecta y el costo de destruirla. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SUGERENCIA: Aplicar a un grafo G un algoritmo visto en clase.

SOLUCIÓN:

(a) Es un caso particular del Ejercicio 175b.

(b) Armar un grafo con un nodo por cada esquina y un eje por cada cuadra con el costo de destruirla. De acuerdo al enunciado, el grafo resulta conexo y lo que se quiere es eliminar ejes hasta que quede un árbol generador. Como minimizar el costo de los ejes eliminados es lo mismo que maximizar el costo de los ejes remanentes, lo que se quiere es obtener un árbol generador máximo. De acuerdo al primer punto puede hacerse multiplicando los costos por -1 y calculando un AGM. Los ejes incorporados al árbol se pueden marcar, y los no marcados indican las cuadras a destruir. La complejidad es la de un algoritmo que obtenga un AGM, ya que la complejidad de armar el grafo y obtener los ejes no marcados es despreciable en comparación al resto.

NOTA: Propuesto originalmente por Guido Tagliavini Ponce.

177. Sea G un grafo conexo con pesos asociados a sus ejes. Sea $e_{\min}(G)$ cualquier eje de G que tenga peso mínimo y $e_{\max}(G)$ cualquier eje de G que tenga peso máximo. ¿Es cierto que...

(a) $e_{\min}(G)$ pertenece a algún árbol generador mínimo de G?

(b) $e_{\min}(G)$ pertenece a todo árbol generador mínimo de G?

(c) $e_{\max}(G)$ no pertenece a algún árbol generador mínimo de G?

(d) $e_{\max}(G)$ no pertenece a ningún árbol generador mínimo de G?

Demostrar o dar un (contra) ejemplo y justificar según corresponda.

Repetir suponiendo que existe un único eje de peso mínimo y un único eje de peso máximo.

SOLUCIÓN:

(a) Verdadero; por ejemplo el árbol que se obtiene usando el algoritmo de Prim a partir de un extremo de $e_{\min}(G)$.

(b) Falso si puede haber más de un eje de peso mínimo; por ejemplo para K_3 con todos ejes de igual peso. Verdadero si hay un único eje de peso mínimo, ya que de lo contrario se podría tomar un supuesto árbol de peso mínimo, agregarle $e_{\min}(G)$, sacarle cualquier eje del ciclo que se forma, y obtener un árbol de peso menor.

(c) Falso; por ejemplo para K_2, o en general si $e_{\max}(G)$ es un puente.

(d) Ídem punto anterior.

TOMADO: 2001C2P1 20-OCT-2001, 2018C1R1 13-JUL-2018.

178. Sea $G = (V, E)$ un grafo conexo con pesos asociados a sus ejes, los cuales son todos distintos entre sí. Sea e_v el eje de peso mínimo entre los incidentes a $v \in V$. Sea $E_{\min} = \{e_v / v \in V\}$, y sea $G_{\min} = (V, E_{\min})$. ¿Es cierto que…

(a) para todo v, el eje e_v está en *algún* árbol generador mínimo de G?

(b) para todo v, el eje e_v está en *todo* árbol generador mínimo de G?

(c) $G_{\min}$ no tiene ciclos simples?

(d) $G_{\min}$ es conexo?

(e) $G_{\min}$ es un árbol generador de G?

(f) $G_{\min}$ es un árbol generador mínimo de G?

(g) $G_{\min}$ es subgrafo de *algún* árbol generador mínimo de G?

(h) $G_{\min}$ es subgrafo de *todo* árbol generador mínimo de G?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN: Es claro que el segundo punto implica al primero, y el último implica al anteúltimo. Por el Ejercicio 6.22 de la Práctica, al ser todos los pesos distintos el árbol generador mínimo es único, de modo que las implicaciones también van en sentido inverso. En lo que sigue no se usan esas implicaciones, sino que se dan demostraciones directas probablemente más complicadas.

(a) Verdadero; por ejemplo el árbol que se obtiene usando el algoritmo de Prim a partir de v.

(b) Verdadero; supongamos que existe un vértice v y un árbol generador mínimo T de G tal que e_v no está contenido en T; si agregamos e_v a T se forma un ciclo que contiene a v y a e_v; si ahora eliminamos de ese ciclo el eje incidente a v que no es e_v, obtenemos un árbol generador de menor peso que T, lo cual es absurdo porque T era mínimo.

(c) Verdadero; por el punto anterior. Alternativamente, por absurdo supongamos que hay un ciclo; sean $v_1, v_2, \dots v_k$ los nodos del ciclo, y $e_1, e_2, \dots e_k$ los ejes (es decir, $e_i = (v_i, v_{(i \bmod k)+1})$; supongamos sin pérdida de generalidad que v_k eligió a e_k, lo que implica que el peso de e_{k-1} (distinto al de e_k) es mayor; pero como e_{k-1} está en el ciclo, debe haber sido elegido por v_{k-1}, lo que implica que el peso de e_{k-2} es mayor; en definitiva, e_1 fue elegido por v_1, lo que implica que el peso de e_k es mayor, que termina teniendo peso mayor a su propio peso. Alternativamente, por absurdo supongamos que hay un ciclo; sea $e = (v, w)$ el eje de peso estrictamente mayor a los otros del ciclo; v no puede haber elegido a e, ni w tampoco, lo cual es absurdo porque sólo ellos podrían haberlo elegido.

(d) Falso; por ejemplo si G es un camino de tres ejes, de pesos 1, 3 y 2.

(e) Falso; debido al punto anterior.

(f) Falso; debido al punto anterior.

(g) Verdadero; para demostrarlo, tomar el árbol generador mínimo que contiene más ejes de $E_{\min}$; si contiene todos los ejes de $E_{\min}$, listo; caso contrario, tomar cualquier eje e_v de $E_{\min}$ que no esté en el árbol, y agregarlo al árbol, lo cual forma un ciclo; sacar el eje del ciclo que es incidente a v y que no es e_v; se obtiene un árbol generador de menor peso que el original, lo cual es absurdo.

(h) Verdadero; debido al segundo punto.

TOMADO: 2002C1P1 18-MAY-2002, 2014C2P1 11-OCT-2014.

179. Un multi-conjunto es una colección de elementos eventualmente repetidos. Dos multi-conjuntos son iguales si y sólo si tienen los mismos elementos con la misma multiplicidad. Por ejemplo, $\{1,5,1\} = \{5,1,1\} \neq \{5,5,1\}$.

Sea G un grafo de n vértices, conexo y con pesos asociados a sus ejes; estos pesos *no* necesariamente son positivos ni distintos entre sí. Para $i \in \{1,2\}$, sea T_i un árbol generador mínimo de G, y sea P_i el multi-conjunto de $n-1$ elementos formado por los pesos de cada uno de los ejes de T_i. ¿Es cierto que $P_1 = P_2$? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SUGERENCIA: Inducción en la cantidad de ejes de la diferencia simétrica entre $E(T_1)$ y $E(T_2)$ (es decir, en la cantidad de ejes que están en uno de los árboles y no en el otro).

SOLUCIÓN: Es cierto, es decir, todos los árboles generadores mínimos de un grafo tienen ejes de la misma longitud. Si $T_1 = T_2$ (están formados por los mismos ejes), no hay nada que demostrar. Caso contrario sea e_1 un eje de menor longitud entre los que están en uno de los árboles y no en el otro; sin pérdida de generalidad, supongamos que e_1 está en T_1. Consideremos el grafo $T_2 + e_1$ el cual tiene un único ciclo. Sea e_2 cualquier eje que está en ese ciclo pero no en T_1 (existe porque caso contrario T_1 tendría un ciclo). Sea $T_3 = T_2 + e_1 - e_2$, que resulta un árbol generador de G. Analicemos los tres casos posibles: 1) si $p(e_1) < p(e_2)$, resulta $p(T_3) < p(T_2)$, lo cual es absurdo porque T_2 era mínimo. 2) si $p(e_1) > p(e_2)$, no hubiéramos elegido e_1. 3) si $p(e_1) = p(e_2)$, resulta $p(T_3) = p(T_2)$, por lo que T_3 es un árbol generador mínimo de G; además, como sacamos y pusimos ejes de igual peso, T_3 tiene ejes de igual longitud que T_2 y tiene un eje en común más con T_1. Repetimos todo para T_1 y T_3 en vez de T_1 y T_2. En cada paso obtenemos dos árboles generadores mínimos que tienen ejes de igual longitud que los anteriores, y un eje en común más que los dos anteriores. Finalmente, obtenemos dos árboles generadores exactamente iguales, por lo que las longitudes de los ejes de los árboles originales eran coincidentes.

TOMADO: 2003C1P1 17-MAY-2003, 2014C1R1 16-JUL-2014.

180. Dado un grafo H con pesos asociados a sus ejes, un árbol generador quasi-mínimo de H es un árbol generador de H que tiene peso mínimo entre los árboles generadores de H que no son mínimos.

Sea G un grafo conexo de n vértices y m ejes, con $m \geq n$. Cada eje tiene un peso asociado, y los pesos son todos distintos entre sí.

(a) ¿Es cierto que G tiene *al menos* un árbol generador quasi-mínimo? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

(b) ¿Es cierto que G tiene *a lo sumo* un árbol generador quasi-mínimo? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

(c) Demostrar que si T_1 es un árbol generador mínimo de G, y T_2 es un árbol generador quasi-mínimo de G, entonces T_1 tiene exactamente un eje que no está en T_2, y viceversa.
SUGERENCIA: Considerar el eje de menor peso en la diferencia simétrica entre $E(T_1)$ y $E(T_2)$.

(d) Repetir los dos primeros puntos sin la restricción $m \geq n$.

(e) Repetir los dos primeros puntos sin la restricción de que los pesos son todos distintos entre sí.

SOLUCIÓN:

(a) Por el Ejercicio 6.22 de la Práctica, al ser todos los pesos distintos el árbol generador mínimo es único. Al ser $m \geq n$, hay más de un árbol generador. Por lo tanto, el conjunto de los árboles generadores que no son mínimos es no vacío, y entonces tiene un elemento de menor peso.

(b) No. Por ejemplo en el grafo formado por un triángulo con pesos 5, 6 y 7, unido por un eje de peso 1 a otro triángulo con pesos 2, 3 y 4. Los ejes de pesos 1, 2, 3, 5 y 7 forman un árbol generador quasi-mínimo, lo mismo que los ejes de pesos 1, 2, 4, 5 y 6.

(c) Sea e el eje de menor peso entre los que están en uno de los árboles y no en el otro. Si e estuviera en T_2, consideremos el grafo $T_1 + e$ el cual tiene un único ciclo; sea e' cualquier eje que está en ese ciclo pero no en T_2 (existe porque caso contrario T_2 tendría un ciclo); sea

$T' = T_1 + e - e'$, que resulta un árbol generador de G; como e' está en T_1 pero no en T_2, el peso de e es menor o igual que el de e', y como todos los pesos son distintos, el peso de e en realidad es menor que el de e', y por lo tanto el peso de T' es menor que el de T_1, lo cual es absurdo porque T_1 era mínimo. Por lo tanto e está en T_1, y repitiendo lo mismo de antes con T_1 y T_2 intercambiados, obtenemos $T' = T_2 + e - e'$ de peso menor que T_2, y que difiere con T_2 en exactamente un eje. Como T_2 es quasi-mínimo, T' debe ser mínimo, y por la unicidad del árbol generador mínimo cuando los pesos son todos distintos, resulta $T' = T_1$.

(d) Si se permite $m < n$, como G es conexo debe ser $m = n - 1$, de modo que es un árbol, y por lo tanto tiene un único árbol generador, el cual es necesariamente mínimo; en ese caso G no tiene árboles generadores quasi-mínimos, de modo que el primer punto es falso. El segundo punto sigue siendo falso, ya que al eliminar ciertas restricciones, el conjunto de grafos posibles todavía incluye a aquellos grafos que hacen que la propiedad no sea cierta en general.

(e) Si los pesos pueden repetirse, en particular pueden ser todos iguales, en cuyo caso todo árbol generador tiene el mismo peso, el cual es mínimo; en ese caso G no tiene árboles generadores quasi-mínimos, de modo que el primer punto es falso. El segundo punto sigue siendo falso, por lo dicho.

NOTA: Propuesto originalmente por Flavia Bonomo. Del tercer punto surge un algoritmo para calcular un árbol generador quasi-mínimo, que consiste un calcular un árbol generador mínimo, y luego para cada eje del árbol sacarlo y encontrar el menor eje distinto del que sacamos que vuelve a conectar las dos partes en que queda dividido el árbol. El costo de este algoritmo es $O(x)+O(mn)$, donde $O(x)$ es el costo de calcular un árbol generador mínimo. Yo esperaría que se pudiera calcular un árbol generador quasi-mínimo en $O(x)$.

TOMADO: 2003C1RX 11-AGO-2003, 2014C2R1 12-DIC-2014.

181. Un grafo (simple) se dice 1-árbol si es un árbol con un eje agregado.

Diseñar un algoritmo eficiente que encuentre un 1-árbol generador mínimo. La entrada del algoritmo es un grafo conexo no dirigido con pesos asociados a sus ejes, y que tiene al menos un ciclo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Un algoritmo que parecería funcionar es uno como el de Kruskal, que siempre elige el eje de menor peso, pero con la condición de que el eje no forme un segundo ciclo. La demostración de que funciona sería similar a la demostración de que el algoritmo de Kruskal funciona. Otro algoritmo que podría funcionar (aunque no lo analicé) es encontrar un árbol generador mínimo del grafo y agregarle el eje de menor peso entre los que no están en el árbol.

NOTA: Este ejercicio es difícil.

182. Un grafo (simple) se dice 1-árbol si es un árbol con un eje agregado.

Sea G un 1-árbol de n vértices con pesos asociados a sus ejes. Diseñar un algoritmo de complejidad

Variante 1: $O(n)$

Variante 2: estrictamente mejor que $\Theta(n^2)$

que determine el peso de un árbol generador mínimo de G. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: El árbol tiene $n - 1$ ejes y es conexo de modo que G tiene $m = n$ ejes y es conexo. Suponemos que G está representado con listas de adyacentes.

Variante 1: El algoritmo consiste en determinar todos los árboles generadores de G y elegir entre ellos el de peso mínimo. Por propiedades vistas sabemos que G tiene un único ciclo. También sabemos que si elimimanos cualquier eje de ese ciclo, se obtiene un árbol generador de G. Veamos que no hay otros árboles generadores. Sea T cualquier árbol generador de G; hay un eje del ciclo que no está en T, porque caso contrario T tendría un ciclo; todos los otros ejes de G tienen que estar en T, porque caso contrario T tendría menos de $n - 1$ ejes. De tal modo, para determinar todos los árboles generadores de G es suficiente con encontrar el ciclo y sacar alternativamente cada eje del ciclo. El peso del árbol es el peso de G menos el peso del eje removido. Para calcular el peso de G basta recorrer las listas de adyacentes y sumar, lo cual tiene complejidad $O(m+n) = O(n)$. Para encontrar el ciclo se puede hacer BFS desde cualquier vértice, y cuando se detecta que desde dos vértices $v_1 \neq v_2$ se llega a un mismo vértice w, se busca el ancestro común más cercano en el árbol de BFS que tengan v_1 y v_2. El ancestro puede encontrarse en $O(n)$ si se marcan en un vector los ancestros de v_1 y luego se recorren en orden los ancentros de v_2 hasta encontrar el primero marcado. Alternativamente se puede hacer un segundo BFS desde w, v_1 o v_2, y cuando se detecta que desde dos vértices $v'_1 \neq v'_2$ se llega a un mismo vértice w', no sólo se encontró el ciclo sino que ya se sabe que la raíz del árbol del segundo BFS es el ancestro común más cercano en ese árbol que tienen v'_1 y v'_2. Con el ancestro se pueden determinar los ejes del ciclo y sus pesos si se guardaron en el árbol de BFS. Como la complejidad de BFS es $O(m+n) = O(n)$, esa es la complejidad total.

Variante 2: Basta usar Prim (con heap) o Kruskal, ya que en las condiciones dadas sus complejidades son $O(m \log n) = O(n \log n)$. Alternativamente se puede hacer lo mismo que en la Variante 1.

NOTA: Propuesto originalmente por Brian Bokser.

TOMADO: 2017C2P1 04-OCT-2017 (Variante 1).

183.

Variante 1: Sea G un grafo conexo con pesos asociados a sus ejes, y sea v un vértice de G. Diseñar un algoritmo eficiente que encuentre un árbol generador mínimo de G en el cual v tenga grado mínimo entre todos los árboles generadores mínimos de G. Demostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SUGERENCIA: Modificar la forma de elegir los ejes en algún algoritmo que encuentre un árbol generador mínimo standard.

Variante 2: Lo mismo, pero pidiendo que se cumpla cualquier propiedad sobre los ejes, que permita decir que un eje es preferible sobre otro del mismo peso.

SOLUCIÓN: Hacer como en Kruskal, pero en caso de empate de dos ejes, elegir el eje que se prefiere sobre el que no se prefiere (para la Variante 1, se trataría de elegir un eje no incidente a v). La complejidad es la de Kruskal.

Respecto de la correctitud, se demuestra de manera similar a Kruskal, tomando del conjunto de todos los árboles generadores mínimos en los cuales v tiene grado mínimo, uno que tenga más ejes en común con el generado por el algoritmo. Sea K el resultado del algoritmo, y sea T el árbol elegido. Si $K = T$ listo. Caso contrario, tomemos el primer eje (en el orden en que los elige el algoritmo) de los ejes que están en K pero no en T. Sea e_k ese eje. Agreguemos e_k a T, con lo cual se forma un único ciclo que lo contiene, ciclo que a su vez tiene al menos un eje e_t que no está en K. Saquemos e_t de $T + e_k$, lo cual produce un nuevo árbol generador T' cuyo peso es $p(T) + p(e_k) - p(e_t)$. No puede ocurrir $p(e_k) < p(e_t)$ porque T' tendría menos peso que el árbol generador mínimo T (igual que en Kruskal). No puede ocurrir $p(e_k) > p(e_t)$ porque el algoritmo hubiera elegido e_t en vez de e_k (igual que en Kruskal). Por lo tanto $p(e_k) = p(e_t)$ y T' es un árbol generador mínimo que tiene un eje más en común con K que los que tiene T. En tal caso, no puede ocurrir que tanto e_k como e_t sean incidentes a v porque T' tendría la menor cantidad posible de ejes incidentes a v, y sería más parecido a K que T. No puede ocurrir que ambos ejes no sean incidentes a v por el mismo motivo. No puede ocurrir que e_k no sea incidente a v y e_t lo sea, porque T' tendría menos ejes incidentes a v que T. Finalmente, no puede ocurrir que e_k sea incidente a v y e_t no lo sea, porque el algoritmo hubiera elegido e_t en vez de e_k.

Otra forma sería modificar los pesos de los ejes agregándole a cada uno una segunda coordenada que vale 1 si es incidente a v, y 0 en caso contrario, y luego correr cualquier algoritmo para AGM. Los pesos se comparan primero por la primera coordenada, y si son iguales se comparan por la segunda. Aparentemente un AGM del grafo modificado es AGM del grafo original, y un AGM del grafo original o bien es AGM del grafo modificado, o existe un AGM del grafo modificado que sólo es mejor en la segunda coordenada.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2008C1P1 17-MAY-2008 (Variante 1).

184. (a) Sea $I \subseteq \mathbb{R}$ un intervalo, y sea $f : I \to \mathbb{R}$ una función. Se dice que f es (estrictamente) creciente en I si y sólo si para todo par $a, b \in I$ se cumple que $a < b \Rightarrow f(a) < f(b)$.

Sea G un grafo conexo con pesos asociados a sus ejes, y sea T un subgrafo de G. Sea $I \subseteq \mathbb{R}$ un intervalo que contiene a todos los pesos, y sea $f : I \to \mathbb{R}$ una función creciente en I. Sea

G_f el grafo que se obtiene remplazando en G cada peso p por $f(p)$. Análogamente se define T_f a partir de T. Demostrar que T es un árbol generador mínimo de G si y sólo si T_f es un árbol generador mínimo de G_f.

SUGERENCIA: Usar que el algoritmo de Kruskal es correcto, y que si f es creciente entonces preserva el orden y tiene inversa, la cual también es creciente.

(b) Sea G un grafo conexo con pesos positivos asociados a sus ejes. Demostrar que T es un árbol generador mínimo de G si y sólo si la productoria de los pesos de sus ejes es mínima dentro del conjunto de los árboles generadores de G.

SOLUCIÓN:

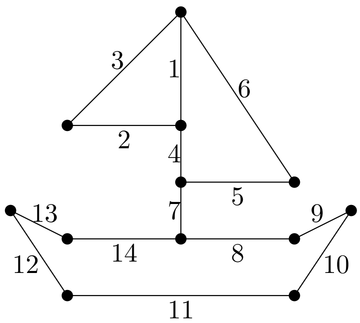
(a) La vuelta es similar a la ida, usando f^{-1} en vez de f, intercambiando T con T_f, y G con G_f. La ida es como sigue. Sin pérdida de generalidad podemos suponer que T es generado por el algoritmo de Kruskal, ya que el mismo es capaz de generar cualquier AGM si los ejes se ordenan apropiadamente. Como f preserva el orden, el mismo orden usado para generar T puede usarse para generar un AGM en G_f. Dado que las adyacencias no cambian por f, cuando un eje es elegido o rechazado por el algoritmo de Kruskal en G para generar T, ese eje va a ser también elegido o rechazado en G_f, dando por resultado exactamente T_f, que por la correctitud del algoritmo resulta un AGM de G_f.

(b) Sea $f(x) = \log x$, que es estrictamente creciente en $\mathbb{R}_{>0}$. Sea $p_H(e)$ el peso del eje e en el grafo H. Por el primer punto T es AGM de G si y sólo si T_f es AGM de G_f. Por definición de AGM eso equivale a decir que para todo T' que sea AG de G_f se cumple $\sum_{e\in E(T_f)} p_{G_f}(e) \leq \sum_{e\in E(T')} p_{G_f}(e)$. Por definición de G_f eso equivale a decir que para todo T' que sea AG de G_f se cumple $\sum_{e\in E(T_f)} \log p_G(e) \leq \sum_{e\in E(T')} \log p_G(e)$. Por propiedad de la función logaritmo eso equivale a decir que para todo T' que sea AG de G_f se cumple $\log \Pi_{e\in E(T_f)} p_G(e) \leq \log \Pi_{e\in E(T')} p_G(e)$. Por ser el logaritmo una función creciente eso equivale a decir que para todo T' que sea AG de G_f se cumple $\Pi_{e\in E(T_f)} p_G(e) \leq \Pi_{e\in E(T')} p_G(e)$. Y como los ejes son los mismos en G que en G_f, salvo sus pesos, eso equivale a decir que para todo T' que sea AG de G se cumple $\Pi_{e\in E(T)} p_G(e) \leq \Pi_{e\in E(T')} p_G(e)$.

NOTA: Propuesto originalmente por Federico Lebrón.

TOMADO: 2012C1P1 19-MAY-2012.

185. Dado el grafo de la figura, exhibir todos sus árboles generadores mínimos. Justificar.



SOLUCIÓN: Todos los pesos son distintos, por lo que existe un único árbol generador mínimo. Se lo puede hallar mediante cualquier algoritmo, por ejemplo Prim o Kruskal.

TOMADO: 2009C1P1 16-MAY-2009.

Camino mínimo

186. En este ejercicio consideramos digrafos con longitudes asociadas a sus ejes, al menos una de las cuales es negativa, pero sin que haya circuitos de longitud negativa. Además, cada digrafo tiene un vértice distinguido llamado origen.

(a) ¿Existe algún digrafo de ese tipo para el cual el algoritmo de Dijkstra determina *correctamente* los caminos mínimos desde el origen?

(b) ¿Existe algún digrafo de ese tipo para el cual el algoritmo de Dijkstra determina *incorrectamente* los caminos mínimos desde el origen?

En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2019C1R1 12-JUL-2019.

187. Sea G un grafo conexo no dirigido con longitudes no negativas asociadas a sus ejes. Sean $s \neq t$ dos vértices de G. Decidir para cada acción propuesta si la longitud de un camino mínimo de s a t puede aumentar, disminuir, o quedar igual. En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar.

(a) Eliminar de G un eje e tal que $G - e$ es conexo.

(b) Agregar a G un eje de cierta longitud.

(c) Dados tres vértices v_1, v_2 y w, tales que v_1 y v_2 son adyacentes pero ninguno de ellos es adyacente a w en G, remplazar el eje (v_1, v_2) de longitud x por los ejes (v_1, w) y (v_2, w), cada uno con longitud $x/2$.

SOLUCIÓN:

(a) Aumenta si e está en todos los caminos mínimos de s a t; por ejemplo si G es K_3 con todos los ejes de longitud 1. No puede disminuir, ya que los caminos en $G - e$ existen en G. Queda igual si e no está en todos los caminos mínimos de s a t; por ejemplo si G es K_3 con el eje entre s y t de longitud 2, y los otros ejes de longitud 1.

(b) Sea f el eje agregado. Los argumentos son similares a los del punto anterior, definiendo $H = G + f$, y contrastando H contra $H - f = G$. No puede aumentar, ya que los caminos en $H - f = G$ existen en $H = G + f$. Disminuye si f está en todos los caminos mínimos de s a t en H; por ejemplo si H es K_3 con todos los ejes de longitud 1. Queda igual si f no está en todos los caminos mínimos de s a t en H; por ejemplo si H es K_3 con el eje entre s y t de longitud 2, y los otros ejes de longitud 1.

(c) No puede aumentar, ya que para cada camino en G existe uno con la misma longitud en el grafo modificado (en particular para el mínimo en G). Disminuye si en el grafo modificado conviene pasar por w para ir de s a t; no es difícil encontrar un ejemplo donde eso ocurra. Queda igual si en el grafo modificado no conviene pasar por w para ir de s a t; no es difícil encontrar un ejemplo donde eso ocurra.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

TOMADO: 2016C1P1 07-MAY-2016.

188. Sea G un grafo o digrafo de n vértices con longitudes no negativas asociadas a sus ejes. Cada combinación que surge de tomar una de las 2 alternativas en cada uno de los 3 ítems que aparecen abajo, define un problema particular de camino mínimo. Indicar el mejor procedimiento posible para resolver cada uno de esos $2^3 = 8$ problemas. Cada procedimiento debe consistir de un preprocesamiento opcional de G, seguido de una o más aplicaciones de algún algoritmo visto en la materia. Para cada procedimiento propuesto indicar la complejidad de cada parte del mismo y la complejidad total del procedimiento.

(a) Desde un único vértice hacia todos, o entre todo par de vértices.

(b) Para $\delta(G) \geq n - 4$, o para $\Delta(G) \leq 4$, donde $\delta(G)$ y $\Delta(G)$ son los grados mínimo y máximo de los vértices de G, respectivamente.

(c) Con G representado mediante matriz de adyacencia, o con G representado mediante listas de adyacentes/sucesores.

SOLUCIÓN: En la tabla siguiente aparecen sobre la izquierda los costos en general de Dijkstra implementado con heap y con vector, Floyd y Dantzig (m es la cantidad de ejes). Para el caso de utilizar Floyd o Dantzig para un único origen (1 a n en la tabla), se calculan todos y se toma el origen deseado. Para el caso de utilizar Dijkstra para todos los orígenes (n a n en la tabla), se lo corre una vez para cada origen diferente. Ford no aparece en la tabla porque es siempre peor o igual que Dijkstra, tanto con heap como con vector, tanto para matriz como para listas. En la parte central de la tabla aparecen las complejidades de la izquierda considerando $m \approx n^2$, ya que ese es el caso cuando $\delta(G) \geq n - 4$. En la parte derecha de la tabla aparecen las complejidades de la izquierda considerando $m = O(n)$, ya que ese es el caso cuando $\Delta(G) \leq 4$. Las mejores alternativas aparecen en negrita. En general no es necesario el preprocesamiento, excepto para el caso n a n, $\Delta(G) \leq 4$ y matriz, en el cual lo mejor es convertir la matriz a listas en $O(n^2)$ y luego correr n veces Dijkstra con heap utilizando las listas.

	general		$\delta(G) \geq n-4$		$\Delta(G) \leq 4$	
	matriz	listas	matriz	listas	matriz	listas
1 a n						
Dijkstra heap	$n^2 + (m+n)\log n$	$(m+n)\log n$	$n^2 \log n$	$n^2 \log n$	$\mathbf{n^2}$	$\mathbf{n \log n}$
Dijkstra vector	n^2	n^2	$\mathbf{n^2}$	$\mathbf{n^2}$	$\mathbf{n^2}$	n^2
Floyd / Dantzig	n^3	n^3	n^3	n^3	n^3	n^3
n a n						
Dijkstra heap	$n^3 + n(m+n)\log n$	$n(m+n)\log n$	$n^3 \log n$	$n^3 \log n$	$\mathbf{n^3} \rightarrow$	$\mathbf{n^2 \log n}$
Dijkstra vector	n^3	n^3	$\mathbf{n^3}$	$\mathbf{n^3}$	n^3	n^3
Floyd / Dantzig	n^3	n^3	$\mathbf{n^3}$	$\mathbf{n^3}$	n^3	n^3

TOMADO: 2015C1R1 13-JUL-2015.

189. Dado un grafo o digrafo con longitudes positivas asociadas a sus ejes, el algoritmo de Dijkstra permite calcular los caminos mínimos desde un vértice hasta cada uno de los vértices del grafo.

(a) Modificar el algoritmo de Dijkstra para que manteniendo su complejidad también calcule para cada vértice

Variante 1: la cantidad de ejes del camino mínimo que termina en ese vértice.

Variante 2: si es o no único el camino mínimo que termina en ese vértice.

Variante 3: la cantidad de caminos mínimos que terminan en ese vértice.

Presentar pseudocódigo del algoritmo modificado, indicando claramente las partes que han cambiado respecto del algoritmo original. Justificar.

(b) El algoritmo de Dijkstra también funciona correctamente cuando hay ejes de longitud cero. ¿Ocurre lo mismo con el algoritmo modificado? Justificar.

SOLUCIÓN:

Variante 1: Mantener un acumulador por cada nodo con la cantidad de ejes por los que pasa el camino que termina en ese nodo. Cada vez que se actualiza la etiqueta de un nodo, actualizar el acumulador sumando 1 al acumulador del predecesor del nodo en el camino. También puede calcularse luego de Dijkstra, usando el vector de predecesores.

No hay problema en que hayas ejes de longitud cero.

Variante 2: Mantener una marca para cada nodo indicando si el camino es único. Cuando una etiqueta es igual a su valor de actualización marcar que el camino mínimo no es único. Cuando una etiqueta se actualiza por menor, tomar la marca del predecesor, sea cual sea.

No funciona con ejes de longitud cero.

Variante 3: Mantener un acumulador para cada nodo indicando la cantidad de caminos mínimos. Cuando una etiqueta es igual a su valor de actualización sumar el acumulador del predecesor. Cuando una etiqueta se actualiza por menor, remplazar por el acumulador del predecesor.

No funciona con ejes de longitud cero.

Puede utilizarse para resolver la Variante 2, viendo al final si el acumulador vale 1.

TOMADO: 2004C1R1 21-JUL-2004 (Variante 1), 2010C2P1 16-OCT-2010 (Variante 2, primer punto), 2011C2R1 14-DIC-2011 (Variante 3).

190. Sea $G = (V, E)$ un grafo o digrafo con longitudes no negativas asociadas a sus ejes.

Dados $V_1 \subseteq V$ y $V_2 \subseteq V$ dos subconjuntos cualesquiera de vértices, definimos la distancia entre V_1 y V_2 como $D(V_1, V_2) = \min_{v_1 \in V_1, v_2 \in V_2} d(v_1, v_2)$, donde $d(v_1, v_2)$ es la longitud de un camino mínimo de v_1 a v_2.

Sean $s \in S \subseteq V$ y $t \in T \subseteq V$. Para cada uno de los problemas indicados diseñar un algoritmo que tenga complejidad mejor o igual que la del algoritmo de Dijkstra. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

(a) Calcular $D(\{s\}, T)$.
(b) Calcular $D(S, \{t\})$.
(c) Calcular $D(S, T)$.
(d) Calcular $D(\{v\}, T)$ para todo $v \in V$.
(e) Calcular $D(S, \{v\})$ para todo $v \in V$.
(f) Calcular $D(\{v\}, \{t\}) = d(v, t)$ para todo $v \in V$.

SOLUCIÓN:

(a) Calcular con el algoritmo standard desde s, y luego tomar el mínimo sobre los vértices de T. Otra posibilidad es agregar un vértice destino, sucesor de los vértices de T con ejes de longitud 0, y calcular con el algoritmo standard desde s a ese vértice.
(b) Invertir todos los ejes y utilizar el punto anterior. Alternativamente, agregar un origen, antecesor de los vértices de S, y calcular con el algoritmo standard desde ese vértice a t.
(c) Combinar las dos soluciones anteriores. Por ejemplo, agregar un origen y un destino, y calcular entre esos dos, o agregar sólo un origen, calcular, y tomar el mínimo sobre los destinos, o agregar sólo un destino, calcular, y tomar el mínimo sobre los orígenes.
(d) Agregar un destino, invertir, y calcular, o invertir, agregar un origen, y calcular.
(e) Agregar un origen y calcular.
(f) Invertir y calcular.

NOTA: Propuesto originalmente por Alejandro Martín Deymonnaz (algunos puntos). Las variantes $D(\{s\}, \{v\})$ y $D(\{s\}, \{t\})$ fueron suprimidas por ser problemas standard.

TOMADO: 2002C1R1 17-JUL-2002 (anteúltimo punto), 2003C1P1 17-MAY-2003 (último punto), 2008C1R1 18-JUL-2008 (tres primeros puntos), 2012C1P1 19-MAY-2012 (cuarto punto).

191. La Federación Espacial ha construido las bases Alien (A), Barbarella (B), Challenger (C), Duran-Duran (D) y Enterprise (E) en el sector AG-3 del Sistema Solar. Hay además dos dispositivos robóticos en ese sector. El Dispositivo Robótico 2 (R2) ha sufrido un desperfecto, y el Dispositivo Robótico 1 (R1) debe ir a su encuentro a fin de poder repararlo. Las distancias entre bases y dispositivos robóticos son las siguientes:

	B	C	D	E	R1	R2
A	1	1	2	4	1	7
B		1	2	5	4	6
C			2	3	5	5
D				2	3	6
E					5	3
R1						14

El R1 se autopropulsa a velocidad constante, y consume 2 unidades de energía por unidad de distancia recorrida. A fin de alcanzar al R2, el R1 también puede ingresar y egresar de las bases construidas. El R1 no consume energía mientras está dentro de una base, pero para ingresar a la base consume 1 unidad de energía, mientras que para egresar consume 1 unidad adicional. Se desea determinar el recorrido que debe seguir el R1 a fin de minimizar el consumo de energía. Modelar la situación como un problema de camino mínimo en un grafo, y resolverlo por el método de Dijkstra. Justificar.

SOLUCIÓN: Armar un grafo con un nodo por cada base o robot. Poner un eje entre cada par de nodos, con un peso igual a dos veces la distancia, más 1 si uno de sus extremos es una base (lo que representa el gasto de ingresar o egresar de ese extremo), o más 2 si ambos extremos son bases (lo que representa el gasto de salir de un extremo e ingresar en el otro).

NOTA: Propuesto originalmente por Flavia Bonomo. Puede pedirse que la energía para ingresar y egresar de las bases sea diferente para cada base.

TOMADO: 2002C1P1 18-MAY-2002.

192. Luego de tanto esperar, Luis finalmente consiguió un auto que lo lleve por las rutas argentinas hasta El Fin. Sin embargo, El Fin queda lejos y es necesario mucho combustible para llegar allá. Luis conoce varias estaciones de servicio en las cuales puede cargar combustible. Además, conoce varias rutas, cada una de las cuales conecta un par de posiciones del conjunto formado por su ubicación actual, El Fin, y las estaciones de servicio. Para cada ruta r Luis sabe cuántas unidades de combustible $c_r \in \mathbb{N}$ necesita el auto para transitarla. Cada vez que Luis llega a una estación de servicio puede cargar combustible, pero debido al racionamiento impuesto por el gobierno el auto sólo puede cargar 1 unidad de combustible en cada estación de servicio. Luis sabe que en este momento el auto tiene $c \in \mathbb{N}$ unidades de combustible, y debido al racionamiento es imposible que durante el viaje hasta El Fin el tanque de combustible se llene, de modo que en la práctica es como si tuviera capacidad ilimitada. Luis está confundido. Antes de partir, quiere que le digamos cuál es el máximo de combustible que puede tener en el tanque del auto cuando llegue a El Fin, o enterarse de que no puede llegar sin quedarse sin combustible durante el viaje. Diseñar un algoritmo eficiente basado en grafos que permita responderle a Luis. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

    SOLUCIÓN: Poner un nodo por la ubicación inicial de Luis, otro por El Fin, y otro por cada estación de servicio. Agregar cada ruta r como un eje dirigido de longitud $c_r - 1$ (si sale de una estación de servicio) o c_r (si no sale), lo cual indica el gasto en combustible de transitarla. Notar que los ejes son no negativos. El máximo de combustible en El Fin equivale al mínimo gasto de combustible. Alcanza con calcular un camino mínimo desde la posición inicial hasta El Fin, por ejemplo con Dijkstra. Si bien un camino mínimo podría repetir nodos (y por lo tanto estaciones de servicio), el que devuelve Dijkstra no lo hace. Sea ℓ la longitud del camino mínimo encontrado. Se puede realizar el viaje si y sólo si $x = c - \ell \geq 0$, y en tal caso x es el máximo combustible con el cual se puede llegar. Notar que no puede ocurrir que en algún momento el tanque se vacíe y luego aparezca combustible, porque ninguna ruta agrega combustible al tanque (las longitudes del modelo son no negativas).

    NOTA: Propuesto originalmente por Pablo Heiber.

    TOMADO: 2010C2R1 15-DIC-2010.

193. En la ciudad hay k espías, y cada uno se oculta en una esquina distinta. Es necesario elegir una esquina para que todos los espías se reunan y puedan intercambiar información secreta. Una vez elegida la esquina, la misma será comunicada simultáneamente a todos los espías, quienes se dirigirán hacia ella de inmediato. La reunión comenzará en cuanto todos los espías hayan llegado a la esquina elegida.

    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 en ambos sentidos. Se puede ir de cualquier esquina a cualquier otra recorriendo las cuadras, pasando eventualmente por esquinas intermedias. Cada espía tarda exactamente 42 segundos en recorrer cada cuadra.

    Diseñar un algoritmo eficiente que determine cuáles esquinas permiten realizar la reunión lo antes posible. La entrada del algoritmo es la cantidad n de esquinas, la cantidad m de cuadras, la cantidad k de espías, para cada cuadra las esquinas que conecta, y para cada espía la esquina en la que está oculto. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(km)$.

    SOLUCIÓN: Armar un grafo con un nodo por cada esquina y un eje por cada cuadra; este grafo resulta conexo de acuerdo al enunciado. Hacer BFS desde cada esquina que tenga un espía, lo cual permite calcular la (mínima) cantidad de cuadras que necesita recorrer ese espía para llegar a cada una de las esquinas de la ciudad. Para cada esquina de la ciudad calcular el máximo sobre todos los espías, de la cantidad de cuadras necesarias para llegar a la esquina, lo cual indica la cantidad de cuadras que debe recorrer el espía que tarda más en llegar (debe recorrer más cuadras). Cualquier esquina con máximo lo menor posible es apropiada para realizar la reunión. Representar el grafo como listas de adyacentes puede hacerse en $O(m + n)$; ejecutar k veces BFS tiene un costo de $O(km + kn)$; encontrar los n máximos cuesta $O(kn)$; calcular el mínimo de los máximos y elegir las esquinas cuesta $O(n)$; el costo total es $O(km)$ ya que $n = O(m)$ en un grafo conexo.

    NOTA: Propuesto originalmente por Michel Mizrahi.

    TOMADO: 2013C1P1 18-MAY-2013.

194. El gobierno de la ciudad decidió construir una nueva cuadra entre algún par de esquinas a fin de agilizar el tránsito vehicular en la hora pico. El gobierno tiene una lista de cuadras 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 cuadra candidata cuya construcción reduciría lo máximo posible la cantidad de cuadras que hay que recorrer en ese viaje.

   La ciudad está formada por n esquinas y m cuadras. 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 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 elija la cuadra candidata cuya construcción reduciría lo máximo posible la cantidad de cuadras 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 cuadras actuales, la cantidad c de cuadras candidatas, y para cada cuadra (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 $O(m + c)$.

   SOLUCIÓN: Una posibilidad es armar un grafo con un nodo por cada esquina y un eje por cada cuadra actual, calcular con BFS la distancia $D_1(v)$ desde la esquina 1 hasta cada esquina v, invertir todos los ejes y calcular con BFS la distancia $D_n(w)$ desde la esquina n hasta cada esquina w (que equivale a la distancia desde la esquina w hasta la esquina n en el grafo original), y elegir la cuadra candidata (v, w) que minimiza $D_1(v)+1+D_n(w)$; si ese valor es mayor o igual que $D_1(n) = D_n(1)$ significa que ninguna cuadra candidata mejora el recorrido desde la esquina 1 hasta la esquina n, y es lo mismo agregar cualquier cuadra candidata (en particular la que minimiza esa expresión). Otra posibilidad es armar dos copias del grafo descripto, por cada cuadra candidata (v, w) agregar un eje desde el nodo v en la primera copia hacia el nodo w en la segunda copia, y calcular con BFS la distancia desde la esquina 1 de la primera copia hasta la esquina n de la segunda copia; como sólo se puede pasar de la primera copia a la segunda usando las cuadras candidatas, y no es posible pasar de la segunda copia a la primera, cualquier camino que vaya desde la esquina 1 de la primera copia hasta la esquina n de la segunda copia usa exactamente una cuadra candidata; hay que elegir la cuadra candidata que realiza la distancia; nuevamente, si esa distancia es mayor o igual que la distancia original, significa que ninguna cuadra candidata mejora el recorrido.

   NOTA: Propuesto originalmente por Michel Mizrahi.

   TOMADO: 2013C2P1 12-OCT-2013.

195. Sea $G = (V, E)$ un grafo o digrafo con longitudes no negativas asociadas a sus ejes. Sean $v_1, v_2 \in V$.

Variante 1: Sea también $w \in V$. Diseñar un algoritmo que encuentre un camino (no necesariamente simple) de v_1 a v_2 que pase por w

Variante 2: Sea $f \in E$. Diseñar un algoritmo que encuentre un camino (no necesariamente simple) de v_1 a v_2 que pase por f

Variante 3: Sea $W \subseteq V$. Diseñar un algoritmo que encuentre un camino (no necesariamente simple) de v_1 a v_2 que pase por al menos un vértice en W

Variante 4: Sea $F \subseteq E$. Diseñar un algoritmo que encuentre un camino (no necesariamente simple) de v_1 a v_2 que pase por al menos un eje en F

   y que tenga longitud total mínima entre todos los caminos de ese tipo; si no existe ningún camino de ese tipo, el algoritmo debe informarlo. El algoritmo debe tener complejidad mejor o igual que la del algoritmo de Dijkstra. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

   SOLUCIÓN: En lo que sigue, siempre usar Dijkstra para calcular caminos mínimos. La Variante 3 y la Variante 4 admiten soluciones muy parecidas a la del Ejercicio 194.

Variante 1: Calcular un camino mínimo de v_1 a w y otro de w a v_2; concatenar ambos tramos. Si el grafo es no dirigido se pueden obtener los tramos aplicando una sola vez el algoritmo para calcular los caminos mínimos desde w hacia el resto de los nodos. También puede resolverse como un caso particular de la Variante 3.

Variante 2: Si el grafo es dirigido calcular un camino mínimo de v_1 al origen de f y otro del destino de f a v_2; intercalar f entre ambos tramos. Si el grafo es no dirigido hay que contemplar las dos orientaciones posibles de f; calcular caminos mínimos desde v_1 a cada extremo de f, y lo mismo para v_2; eso puede hacerse desde v_1 y v_2, o desde los extremos de f; tomar el mínimo entre ir desde v_1 a un extremo de f más ir desde el otro extremo a v_2, y lo mismo con los extremos intercambiados; de acuerdo a la orientación ganadora, intercalar f entre los tramos apropiados. También puede resolverse como un caso particular de la Variante 4.

Variante 3: Calcular la distancia $D_{v_1}(w)$ desde v_1 hasta cada nodo w. Calcular la distancia $D_{v_2}(w)$ desde cada nodo w hasta v_2; esto puede hacerse tomando como origen v_2 e invirtiendo previamente los ejes si el grafo es dirigido. Elegir el nodo de W que minimiza $D_{v_1}(w)+D_{v_2}(w)$ y concatenar los tramos correspondientes.

Otra posibilidad es armar dos copias del grafo, y por cada nodo de W agregar un eje dirigido (independientemente de si el resto del grafo lo es o no) desde el nodo el la primera copia hacia el mismo nodo en la segunda copia. Calcular un camino mínimo desde v_1 en la primera copia hasta v_2 en la segunda copia, el cual forzosamente va a usar al menos un nodo de W.

Variante 4: Calcular la distancia $D_{v_1}(w)$ y $D_{v_2}(w)$ como en la variante anterior. Elegir el eje de F que minimiza la suma de su longitud más la suma de $D_{v_1}()$ y $D_{v_2}()$ de sus extremos. Si el grafo es no dirigido hay que contemplar las dos orientaciones posibles de cada eje de F.

Otra posibilidad es armar dos copias del grafo, y por cada eje de F agregar un eje dirigido (independientemente de si el resto del grafo lo es o no) desde el origen del eje el la primera copia hacia el destino del eje en la segunda copia. Si el grafo es no dirigido hay que agregar dos ejes por cada eje de F a fin de contemplar las dos orientaciones posibles. Calcular un camino mínimo desde v_1 en la primera copia hasta v_2 en la segunda copia, el cual forzosamente va a usar al menos un eje de F.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2012C2P1 13-OCT-2012 (Variante 1), 2013C2R1 16-DIC-2013 (Variante 2), 2014C1P1 17-MAY-2014 (Variante 3), 2014C1R1 16-JUL-2014 (Variante 4).

196. Bob el Destructor, también conocido como BeD, se acaba de despertar en una habitación de una casa que no es la suya. Como de costumbre, tiene a mano sus herramientas. BeD está preocupado porque no sabe dónde se encuentra, de modo que quiere llegar al exterior de la casa lo antes posible. BeD puede pasar de una habitación a otra o al exterior atravesando las puertas que hay en la casa. Además, puede utilizar sus herramientas tantas veces como sea necesario para hacer un agujero en cualquier pared. Una vez que BeD ha agujereado una pared, puede atravesar la misma tantas veces como desee.

La casa está formada por $n-1$ habitaciones, y cada una se identifica por un entero distinto entre 1 y $n-1$. La habitación donde está BeD se identifica por el número 1, y el exterior por el número n. La casa tiene m_1 puertas y m_2 paredes. Cada puerta y cada pared está ubicada entre determinado par de habitaciones, o entre una habitación y el exterior. Cada puerta y cada pared ya agujereada se puede atravesar en un sentido o en el otro. Existe al menos una forma de ir de la habitación inicial de BeD al exterior atravesando las puertas o las paredes ya agujereadas, pasando eventualmente por habitaciones intermedias. BeD tarda 3 minutos en agujerear una pared, y tarda 1 minuto en atravesar una puerta o una pared ya agujereada. Una pared no agujereada no puede ser atravesada. Inicialmente ninguna pared está agujereada.

Diseñar un algoritmo eficiente que determine el mínimo tiempo que BeD necesita para llegar al exterior de la casa. La entrada del algoritmo es el valor n (habitaciones y exterior), la cantidad m_1 de puertas, la cantidad m_2 de paredes, y para cada puerta y cada pared el par de elementos que están a ambos lados de la misma. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(m_1 + m_2 + n)$.

SOLUCIÓN: Armar un grafo con un nodo por cada elemento (habitaciones y exterior), y un eje por cada puerta y cada pared. Asignar peso 1 a los ejes correspondientes a puertas, lo cual representa el tiempo que tarda BeD en atravesarlas. Asignar peso $3 + 1 = 4$ a los ejes correspondientes a paredes, lo cual representa el tiempo que tarda BeD en agujerear la pared y atravesar la pared ya agujereada. Lo que se busca es el peso de un camino mínimo desde el nodo 1 al nodo n. Podría argumentarse que el peso de los ejes correspondientes a paredes no es correcto ya que el costo de pasar por una pared ya agujereada no es 4 sino 1. Sin embargo, en su camino óptimo al exterior BeD nunca va a pasar más de una vez por el mismo elemento, de modo que no es necesario considerar el costo de atravesar una pared que no se acaba de agujerear. Para encontrar el costo del camino mínimo se podría usar el algoritmo de Dijkstra, pero habría que justificar que también funciona en multigrafos, y además no sería eficiente. Una mejor alternativa es remplazar cada eje de peso 4 por 4 ejes de peso 1 y utilizar BFS ya que en el grafo modificado (que además ya no puede ser un multigrafo) todos los pesos son iguales. Si representamos el grafo con listas de adyacentes, tanto el costo de armar el grafo como el de aplicar BFS es $O(m_1 + m_2 + n)$, por lo que ese es el costo total.

NOTA: Propuesto originalmente por Michel Mizrahi.

TOMADO: 2013C1R1 17-JUL-2013.

197. El lobo se come a la cabra y la cabra se come al repollo, pero no lo hacen cuando hay un hombre presente. Un hombre, un lobo, una cabra y un repollo deben cruzar un río en un bote que sólo puede contener a dos de los cuatro. El hombre es el único que sabe conducir el bote, y quiere cruzar el río haciendo la menor cantidad posible de viajes de ida y vuelta. Modelar la situación como un problema de camino mínimo en un grafo, y resolverlo por algún método eficiente. Justificar.

    SOLUCIÓN: Armar un grafo con un nodo por cada estado posible de la situación, teniendo en cuenta que el hombre y el bote se deben unificar. El método apropiado es BFS, ya que todos los ejes tienen la misma longitud.

    TOMADO: 2004C1P1 22-MAY-2004.

198. Sea G un grafo o digrafo con longitudes asociadas a sus ejes, cada una de las cuales vale 1 o 2. Diseñar un algoritmo lo más eficiente posible que calcule los caminos mínimos desde un vértice dado hacia cada uno de los vértices del grafo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

    SOLUCIÓN: Remplazar cada eje de longitud 2 por dos ejes de longitud 1, y luego hacer BFS.

    TOMADO: 2011C2P1 15-OCT-2011.

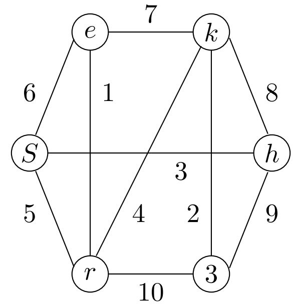
199. Alicia, un burro y un conejo quieren ir desde El País de las Maravillas hasta El Palacio de las Zanahorias. Para hacerlo deben cruzar un puente colgante que resiste el peso de a lo sumo dos individuos. Además, como es de noche, cualquiera que cruce el puente debe llevar una linterna. Alicia y sus amigos tienen una única linterna. Si Alicia cruza sola el puente, tarda 5 minutos, mientras que el burro tarda 2 minutos, y el conejo 1 minuto. Si dos individuos cruzan el puente juntos, tardan lo que tardaría el más lento de los dos si fuera solo, debido a que deben usar la linterna para iluminar el recorrido de ambos individuos. Por ejemplo, si Alicia y el conejo cruzan juntos el puente, tardan 5 minutos en hacerlo. Alicia, el burro y el conejo quieren llegar lo antes posible a destino. Modelar la situación como un problema de camino mínimo en un grafo, y resolverlo por el método de Dijkstra. Justificar.

    SOLUCIÓN: Armar un grafo con un vértice por cada estado posible de la situación, y ejes de peso adecuado entre los vértices.

    NOTA: Basado en un acertijo comentado por Darío Guzik durante 2004C1P1.

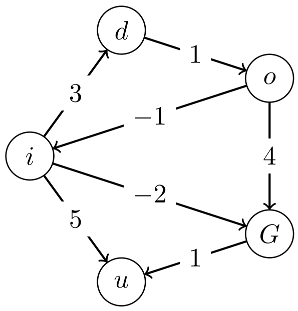
    TOMADO: 2005C1R1 22-JUL-2005, 2018C1R1 13-JUL-2018.

200. Utilizar el algoritmo de Dijkstra para calcular los caminos mínimos desde el vértice S hacia todos los vértices del grafo que aparece en la figura. Presentar pseudocódigo del algoritmo y realizar un seguimiento del mismo. Hecho esto, si se ordenan los vértices de acuerdo a su distancia al origen, podrá leerse el título de una película que se estrenó ayer en EEUU, y anteayer en Rusia.



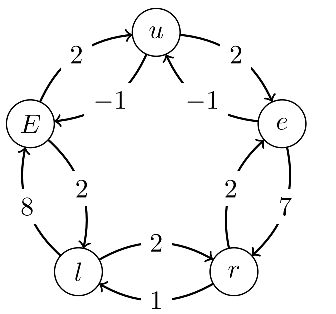
TOMADO: 2007C1P1 19-MAY-2007 (Shrek3), 2016C1R1 11-JUL-2016 (Matrix).

201. Utilizar el algoritmo de Ford para calcular los caminos mínimos desde el vértice i hacia todos los vértices del digrafo que aparece en la figura. Presentar pseudocódigo del algoritmo y realizar un seguimiento del mismo. Hecho esto, si se ordenan los vértices de acuerdo a su distancia al vértice inicial, podrá leerse el nombre de un ex-docente de la materia.



TOMADO: 2016C2R1 12-DIC-2016 (Guido), 2019C1P1 08-MAY-2019 (Knuth).

202. Utilizar un algoritmo eficiente visto en clase para calcular los caminos mínimos desde el vértice u hacia todos los vértices del digrafo que aparece en la figura. Presentar pseudocódigo del algoritmo y realizar un seguimiento del mismo. Hecho esto, si se ordenan los vértices de acuerdo a su distancia al vértice inicial, podrá leerse el apellido del matemático suizo que descubrió la relación entre la exponenciación para números complejos y las funciones trigonométricas, uno de cuyos casos particulares es la igualdad $e^{\pi i} + 1 = 0$.



SOLUCIÓN: Utilizar Ford.

TOMADO: 2017C2P1 04-OCT-2017 (Euler).

203. En cada iteración del algoritmo de Ford se analizan todos los ejes de un grafo intentando mejorar los caminos desde cierto vértice origen hacia cada uno de los vértices del grafo. Considerar la versión del algoritmo que termina cuando no hay ninguna mejora durante una iteración completa. Exhibir una familia infinita de instancias tal que la n-ésima instancia cumple a la vez las siguientes tres condiciones.

- El grafo tiene n vértices.
- La complejidad del algoritmo es $O(n)$ si los ejes son analizados en cierto orden.
- La complejidad del algoritmo es $\Omega(n^2)$ si los ejes son analizados en cierto orden.

Justificar.

SOLUCIÓN: El grafo puede ser P_n con pesos positivos y el vértice origen cualquiera. Si se analizan primero los ejes más cercanos al origen, luego de $O(1)$ iteraciones se obtienen los caminos mínimos. Si se analizan primero los ejes más lejanos, se necesitan tantas iteraciones como la porción más larga a cada lado del origen, la cual tiene $\Omega(n)$ ejes. Cada iteración tiene complejidad $\Theta(n)$.

TOMADO: 2015C1P1 16-MAY-2015.

204. Sea G un digrafo con longitudes asociadas a sus ejes. El algoritmo de Bellman-Ford visto en clase permite decidir si G tiene ciclos de longitud negativa alcanzables desde un vértice que se elige como origen.

    (a) Supongamos que efectivamente G tiene ciclos de longitud negativa alcanzables desde el origen. Queremos encontrar uno de esos ciclos además de saber que existe. Modificar el algoritmo de Bellman-Ford, sin alterar su complejidad asintótica, para que luego de ejecutarlo sea posible encontrar un ciclo C de longitud negativa con costo proporcional a la cantidad de ejes de C. Presentar pseudocódigo del algoritmo modificado, indicando claramente las partes que han cambiado respecto del algoritmo original. Justificar.

    (b) Queremos decidir si hay ciclos de longitud negativa en G, pero no nos interesa si son alcanzables o no desde un vértice dado. Es fácil resolver este problema ejecutando el algoritmo de Bellman-Ford tantas veces como vértices tiene G. Diseñar un mecanismo que resuelva el problema con una única aplicación del algoritmo mencionado y sin alterar su complejidad asintótica. Justificar.

    SOLUCIÓN:

    (a) En cada iteración recordar cualquier nodo que haya cambiado su etiqueta. Si en la n-ésima iteración un nodo v cambió su etiqueta, ese nodo pertenece a un ciclo negativo. Podemos encontrar el ciclo si comenzamos desde v y nos movemos a su antecesor de Ford, luego al antecesor del antecesor, y así sucesivamente hasta que v aparezca nuevamente.

    (b) Agregar un nodo ficticio que sea antecesor de todos los otros nodos del grafo, y ejecutar el algoritmo tomando ese nodo como origen.

    NOTA: Propuesto originalmente por Min Chih Lin.

    TOMADO: 2018C2P1 03-OCT-2018.

205. Un coleccionista de automóviles quiere promocionar su actividad para lo cual está planeando hacer un recorrido por la ciudad mientras maneja 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.

    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.

    SOLUCIÓN: Para cada cuadra q sea $p(q) = c(q) - k \times b(q)$. Se cumple $c(R)/b(R) < k \Leftrightarrow c(R) < k \times b(R) \Leftrightarrow c(R) - k \times b(R) < 0 \Leftrightarrow \sum_{q \in R} [c(q) - k \times b(q)] < 0 \Leftrightarrow \sum_{q \in R} p(q) < 0$. Por lo tanto, si armamos un grafo con un nodo por cada esquina y un eje dirigido de peso $p(q)$ por cada cuadra q, decidir si existe un recorrido aceptable equivale a decidir si el grafo tiene un ciclo negativo. Para decidir eso puede utilizarse el algoritmo de Ford comenzando desde cualquier nodo, ya que ese algoritmo detecta ciclos negativos alcanzables desde el origen, y el enunciado indica que el grafo resultante es fuertemente conexo. La complejidad es la de armar el grafo más la de correr el algoritmo de Ford. Esta última complejidad es la que domina, de modo que el algoritmo completo con listas de adyacentes es $O((m + n)n) = O(mn)$ ya que el grafo es conexo.

    NOTA: Propuesto originalmente por Michel Mizrahi.

    TOMADO: 2015C2P1 03-OCT-2015.

206. Ruth está tan especializada en su trabajo que gana fortunas con cada cosa que hace. Como contrapartida, sólo algunas empresas requieren sus servicios. Ruth conoce las empresas que podrían convocarla, así como las ciudades donde están ubicadas. Un conjunto de rutas comunican las ciudades entre sí, y cada ruta cuenta con una estación de peaje. Usualmente cada estación de peaje cobra cierto importe a cualquiera que transite por la ruta correspondiente, pero algunas rutas están tan subsidiadas que entregan dinero a los viajeros en vez de cobrarles.

Variante 1: Ruth no cree en la redistribución de la riqueza, por lo que quiere vivir en una ciudad tal que para cada empresa exista al menos una manera de ir a la misma con el importe neto de los peajes a su favor.

Variante 2: Ruth cree firmemente en la redistribución de la riqueza, por lo que quiere vivir en una ciudad tal que para cada empresa todas la maneras de ir a la misma tengan el importe neto de los peajes en su contra.

Hay n ciudades, p ciudades que tienen empresas, y m rutas. Cada ciudad se identifica por un entero distinto entre 1 y n. Cada ruta conecta determinado par de ciudades y se puede recorrer desde la primera hacia la segunda. Se puede ir de cualquier ciudad a cualquier otra recorriendo las rutas, pasando eventualmente por ciudades intermedias.

Diseñar un algoritmo de complejidad $O(n^3)$ que decida si existe alguna ciudad en la que Ruth quiera vivir. La entrada del algoritmo es la cantidad n de ciudades, la cantidad p de ciudades que tienen empresas, la cantidad m de rutas, la lista de ciudades que tienen empresas, y para cada ruta su ciudad inicial, su ciudad final y el importe de su peaje (negativo cuando la estación de peaje entrega dinero en vez de cobrarlo). Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

Variante 2: El mejor algoritmo que conocemos tiene complejidad $O(mn)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

SOLUCIÓN: Armar un grafo con un nodo por cada ciudad, y un arco por cada ruta, etiquetado con el importe de su peaje. De acuerdo al enunciado, el grafo resulta fuertemente conexo, de modo que $n = O(m)$.

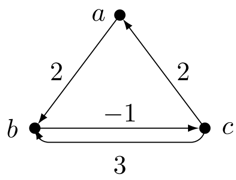
Variante 1: Para cada ciudad y empresa existe al menos una manera de ir de la ciudad a la empresa con el importe neto de los peajes a favor si y sólo si el camino mínimo de la ciudad a la empresa es negativo (o al menos no positivo). Calcular los caminos mínimos entre todo par de vértices con Floyd o Dantzig, y determinar si existe alguna ciudad para la cual los caminos mínimos a todas las empresas son negativos. Si los caminos mínimos no se pueden calcular porque hay ciclos negativos, como el grafo es fuertemente conexo entonces toda ciudad sirve para vivir. Si se representa el grafo con matriz de adyacencia, el costo de armar el grafo es $O(n^2)$, el costo de ejecutar Floyd o Dantzig es $O(n^3)$, y el costo de verificar si cada ciudad es apropiada es $O(n^2)$, de modo que el costo total es $O(n^3)$.

Variante 2: Para cada ciudad y empresa todas la maneras de ir de la ciudad a la empresa tienen el importe neto de los peajes en contra si y sólo si el camino mínimo de la ciudad a la empresa es positivo (o al menos no negativo). Una solución similar a la de la Variante 1 es posible. Para cada ciudad el camino mínimo de la ciudad a todas las empresas es positivo si y sólo si el mínimo (sobre todas las empresas) de los caminos mínimos de la ciudad a cada empresa es positivo. En definitiva se trata de determinar si existe alguna ciudad v para la cual la distancia $D(\{v\}, T) > 0$, donde T es el conjunto de empresas y la distancia es la definida en el Ejercicio 190. En ese ejercicio se propone utilizar el algoritmo de Dijkstra, lo cual no es posible aquí porque hay longitudes negativas. No obstante, puede utilizarse el algoritmo de Ford. Si los caminos mínimos no se pueden calcular porque hay ciclos negativos, como el grafo original es fuertemente conexo entonces ninguna ciudad sirve para vivir. Si se representa el grafo con listas de sucesores, el costo de armar el grafo es $O(m + n)$, el costo de agregar un nodo es $O(n)$, el costo de ejecutar Ford es $O((m + n)n)$, y el costo de verificar si cada ciudad es apropiada es $O(n)$, de modo que el costo total es $O((m + n)n) = O(mn)$.

NOTA: Propuesto originalmente por Brian Bokser (Variante 2).

TOMADO: 2017C1P1 13-MAY-2017 (Variante 1), 2017C1R1 14-JUL-2017 (Variante 2).

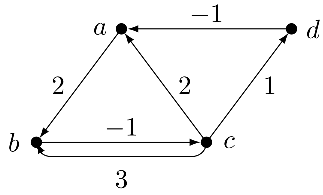
207. (a) Dado el grafo de la figura, calcular los caminos mínimos entre todos los pares de vértices utilizando el algoritmo de Floyd. Justificar.



(b) ¡Uy! Perdón, nos equivocamos.

Variante 1: La longitud del arco (c, a) es en realidad 0.

Variante 2: Nos faltó dibujar el vértice d. El grafo es en realidad el siguiente.



Diseñar un algoritmo que recalcule las distancias en un grafo original de n vértices, y aplicarlo a la instancia del enunciado. El algoritmo debe tener complejidad temporal y espacial estrictamente mejor que $\Omega(n^3)$. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar.

SOLUCIÓN: En cualquiera de las dos variantes agregar un vértice y usar Dantzig. Resulta $O(n^2)$.

TOMADO: 2005C2P1 15-OCT-2005 (Variante 1), 2005C2R1 15-DIC-2005 (Variante 2), 2018C2R1 12-DIC-2018 (Variante 2).

208. Sea $G = (V, E)$ un grafo o digrafo donde cada eje y cada vértice tiene asociado un peso. Dado un camino entre dos vértices del grafo, se define su costo como la suma de los pesos de los ejes del camino, más el máximo peso de los vértices del camino (incluyendo los extremos). Diseñar un algoritmo eficiente que calcule los costos mínimos de los caminos entre cada par de vértices del grafo. El algoritmo debe tener complejidad $O(n^3)$, donde $n = |V|$. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal y espacial coincidentes con las del algoritmo de Dantzig.

SUGERENCIA: Ordenar los vértices.

SOLUCIÓN: Correr Dantzig y considerar las matrices de los sucesivos pasos. Sumar en todas las posiciones de cada matriz el peso del vértice agregado en ese paso. Para cada posición quedarse con el mínimo de todas las matrices que contienen esa posición, que resulta el costo mínimo entre los vértices correspondientes a esa posición.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

TOMADO: 2014C2P1 11-OCT-2014.

209. Dado un digrafo, un orden $v_1, v_2, \ldots$ de sus vértices se dice topológico si y sólo si para todo eje (v_i, v_j) del digrafo se cumple que $i < j$ (es decir, si hay un eje dirigido de v_i a v_j, entonces v_i aparece antes que v_j en el orden).

(a) Sea $G = (V, E)$ un digrafo sin ciclos dirigidos y con longitudes asociadas a sus ejes, algunas de las cuales podrían ser negativas. Demostrar que es correcto el siguiente algoritmo para encontrar caminos *máximos* desde un vértice v hacia todos los vértices del digrafo.

PASO 1: Sea $n = |V|$. Sea $v_1, v_2, \ldots v_n$ un orden topológico de los vértices de G. Sea $M = \max_{(v_i,v_j)\in E} \text{longitud}(v_i, v_j)/(j - i)$.

PASO 2: Ajustar las longitudes de todos los ejes haciendo para cada $(v_i, v_j) \in E$

$$\text{longitud}'(v_i, v_j) = M \times (j - i) - \text{longitud}(v_i, v_j) \ .$$

PASO 3: Aplicar el algoritmo para caminos mínimos de Dijkstra a partir del vértice v utilizando las longitudes modificadas en el paso anterior en vez de las longitudes originales.

SUGERENCIA: Demostrar primero que las longitudes modificadas son no negativas.

(b) ¿Sigue siendo correcto el algoritmo del punto anterior si se utiliza un orden arbitrario de los nodos, en lugar de un orden topológico? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

NOTA: Propuesto originalmente por Flavia Bonomo.

TOMADO: 2001C2P1 20-OCT-2001, 2001C2RX 29-DIC-2001.

210. Sea $G = (V, E)$ un digrafo con longitudes asociadas a sus ejes, algunas de las cuales podrían ser negativas. No obstante, G no tiene circuitos de longitud negativa. Decidir si es correcto el siguiente algoritmo para encontrar caminos mínimos desde un vértice v hacia todos los vértices de G. En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

PASO 1: Calcular $a = \min_{e \in E} \text{longitud}(e)$.

PASO 2: Ajustar las longitudes de todos los ejes haciendo

$$\text{longitud}'(e) = \text{longitud}(e) - a \ ,$$

para cada $e \in E$.

PASO 3: Aplicar el algoritmo de Dijkstra utilizando las longitudes modificadas en el paso anterior (que resultan no negativas) en vez de las longitudes originales.

SOLUCIÓN: No anda. Tomar por ejemplo K_3, con ejes -2, -1 y -2.

TOMADO: 2001C1R1 17-JUL-2001, 2009C1R1 19-AGO-2009.

211. Sea G un grafo o digrafo en el cual cada eje e tiene asociada una longitud no necesariamente positiva $\ell(e) \in \mathbb{R}$. Dada una función $f : \mathbb{R} \to \mathbb{R}$, definimos G_f como el grafo que tiene los mismos vértices y ejes que G, pero en el cual cada eje e tiene asociada la longitud $f(\ell(e))$ en vez de $\ell(e)$. Sea C un camino desde el vértice v hasta el vértice w en G, el cual por definición también existe en G_f, y cuya longitud total en cada grafo es la suma de las longitudes asociadas a sus ejes en ese grafo. ¿Es cierto que...

(a) para $f(x) = x + b$, C es un camino *mínimo* en G si y sólo si C es un camino *mínimo* en G_f?

(b) para $f(x) = ax$ con $a < 0$, C es un camino *mínimo* en G si y sólo si C es un camino *máximo* en G_f?

(c) para $f(x) = ax$ con $a > 0$, C es un camino *mínimo* en G si y sólo si C es un camino *mínimo* en G_f?

(d) para $f(x) = ax$ con $a > 0$, C es un camino *máximo* en G si y sólo si C es un camino *máximo* en G_f?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Falso; el inconveniente es que cuando se suma b a cada eje, los caminos de muchos ejes alteran más su longitud que los de pocos ejes, de modo que su orden relativo puede cambiar; es sencillo encontrar un contraejemplo en un grafo con dos caminos de v a w, uno de un eje y otro de dos.

(b) Verdadero; para demostrarlo hay que tomar cualquier camino de v a w y probar que C es óptimo con determinadas longitudes, usando que lo es con las otras longitudes.

(c) Verdadero; se puede demostrar de la misma manera que el segundo punto; alternativamente se puede usar que el segundo punto es cierto y que $ax = (-1)(-a)x$ con -1 y $-a$ menores que 0.

(d) Ídem punto anterior.

NOTA: Es similar al Ejercicio 175 pero de caminos.

TOMADO: 2014C2R1 12-DIC-2014.

212. Dado un grafo conexo no dirigido con pesos no negativos asociados a sus ejes, un árbol de caminos mínimos desde un vértice v es un árbol generador del grafo tal que los caminos simples desde v en el árbol son caminos mínimos desde v en el grafo. El algoritmo de Dijkstra permite construir un árbol de caminos mínimos desde el vértice elegido como origen.

 Sea G un grafo conexo no dirigido con todos sus ejes del mismo peso $p \geq 0$. ¿Es cierto que...

 (a) algún árbol de caminos mínimos de G es un árbol generador mínimo de G?
 (b) todo árbol de caminos mínimos de G es un árbol generador mínimo de G?
 (c) algún árbol generador mínimo de G es un árbol de caminos mínimos de G?
 (d) todo árbol generador mínimo de G es un árbol de caminos mínimos de G?

 En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

 SOLUCIÓN:

 (a) Verdadero; por el punto siguiente.
 (b) Verdadero; cualquier árbol generador es AGM, ya que tiene el peso de un AGM.
 (c) Verdadero; si T es un árbol de caminos mínimos, por el punto anterior es un AGM.
 (d) Falso; por ejemplo si $G = K_n$ con $n \geq 4$ y el AGM es P_n, ya que todo árbol de caminos mínimos tiene la forma de $K_{1,n-1}$.

 NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

 TOMADO: 2013C2R1 16-DIC-2013.

213. (a) Exhibir un grafo conexo no dirigido sin ejes negativos para el cual *todo* árbol de caminos mínimos sea un árbol generador mínimo. Justificar.
 (b) Exhibir un grafo conexo no dirigido sin ejes negativos para el cual *ningún* árbol de caminos mínimos sea un árbol generador mínimo. Justificar.

 SOLUCIÓN:

 (a) Es suficiente con que el grafo sea un árbol, o que tenga todos los ejes de igual longitud.
 (b) La idea es tomar un grafo completo en el cual todos los caminos mínimos sean directos, pero que de cada vértice salga un eje más largo que los demás, de modo tal que todo árbol de caminos mínimos tenga un eje largo, pero ningún árbol generador mínimo tenga un eje largo; se puede elegir K_4 con dos ejes no adyacentes de longitud 1.5, y los otros ejes de longitud 1.

 TOMADO: 2001C2R1 19-DIC-2001, 2007C1R1 17-JUL-2007, 2018C2R1 12-DIC-2018.

214.

Variante 1: Sea T un árbol de n vértices. Diseñar un algoritmo de complejidad $O(n^2)$ que calcule la distancia (medida en cantidad de ejes) entre todos los pares de vértices de T.

Variante 2: Sea T un árbol de n vértices con longitudes positivas asociadas a sus ejes. Diseñar un algoritmo de complejidad $O(n^2)$ que calcule la distancia entre todos los pares de vértices de T.

Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: La Variante 1 es un caso particular de la Variante 2.

Una posible solución para la Variante 1 es hacer BFS usando como raíz cada vértice de T, e ir etiquetando los otros vértices de manera adecuada. Esto funciona dado que cuando las longitudes son todas iguales BFS se comporta igual que el algoritmo de Dijkstra. La complejidad es la de BFS, que en un árbol es $O(n)$. Sin embargo, como en un árbol existe un único camino entre cada par de vértices, en realidad puede recorrerse T de cualquier manera (por ejemplo, con DFS) usando como raíz cada vértice, y calculando la distancia de la raíz a cada vértice nuevo como 1 más la distancia al padre.

El algoritmo propuesto también funciona para la Variante 2, sumando la longitud del eje en vez de 1.

En vez de resolver para cada nodo, otra posibilidad es sacarle una hoja h a T, resolver recursivamente, y luego calcular la distancia entre h y los otros vértices sumando la longitud de eje (h, v)

con las distancias de v a los otros vértices de T, donde v es el vértice en T que es adyacente a h. La complejidad en este caso es n más la complejidad de la solución recursiva. Se prueba fácilmente que la complejidad total es $O(n^2)$.

Probablemente se puede hacer también una variante de Dantzig que corra con esa complejidad, usando el mismo argumento (cada iteración de Dantzig sería $O(n)$).

NOTA: Basado en 5.14 de Manber. La Variante 1 creo que puede resolverse en $O(n)$ usando LCA.

TOMADO: 2008C1P1 17-MAY-2008 (Variante 1).

215. Dado un grafo $G = (V, E)$, se define la distancia entre dos de sus vértices como la mínima cantidad de ejes que hay en cualquier camino que una a esos vértices, o infinito si no existe ningún camino entre esos vértices. Además, se define

$$\begin{aligned} \text{diametro}(G) &= \max_{v,w \in V} \text{distancia}(v, w) \ , \\ \text{radio}(G) &= \min_{v \in V} \max_{w \in V} \text{distancia}(v, w) \ , \\ \text{centro}(G) &= \arg\min_{v \in V} \max_{w \in V} \text{distancia}(v, w) \ . \end{aligned}$$

(a) Demostrar que si G es conexo entonces $\text{radio}(G) \leq \text{diametro}(G) \leq 2 \times \text{radio}(G)$.

(b) Exhibir un grafo conexo no trivial para el cual $\text{radio}(G) = \text{diametro}(G)$. Justificar.

(c) Exhibir un grafo conexo no trivial para el cual $\text{diametro}(G) = 2 \times \text{radio}(G)$. Justificar.

(d) Exhibir un grafo conexo para el cual $\text{radio}(G) = \text{diametro}(G) = 2 \times \text{radio}(G)$. Justificar.

(e) Demostrar que si G es un árbol entonces tiene un único centro, o tiene dos centros adyacentes.

216. Dado un grafo G, se define la distancia entre dos de sus vértices como el mínimo, sobre todos los caminos que van del primer al segundo vértice, de la cantidad de ejes que hay en el camino, o infinito si tales caminos no existen; además, se define $D(G)$ como el máximo, sobre todos los pares de vértices de G, de la distancia entre esos vértices (diámetro).

Sea G un grafo no dirigido.

(a) Demostrar que si $D(G)$ es infinito (es decir, G no es conexo) entonces $1 \leq D(G^c) \leq 2$ (en particular, G^c es conexo).

(b) Demostrar que si $D(G) \geq 3$ entonces $D(G^c) \leq 3$.
SUGERENCIA: Si G es conexo, organizar los vértices en $D(G) + 1$ niveles.
SUGERENCIA: Si G es conexo, considerar dos vértices a distancia $D(G)$.

(c) Demostrar que si G es autocomplementario entonces $D(G) \in \{0, 2, 3\}$.

(d) Para cada $k \in \{0, 2, 3\}$ exhibir un grafo autocomplementario cuyo diámetro sea k. Justificar.

SOLUCIÓN:

(a) Según el Ejercicio 98, G^c contiene un subgrafo generador bipartito completo, cuyo diámetro es a lo sumo 2.

(b) Si G no es conexo, según el Ejercicio 98, G^c contiene un subgrafo generador bipartito completo, cuyo diámetro es a lo sumo 2. Si G es conexo la demostración es como sigue. Sean u y v dos nodos que realizan $D(G)$.

Dividamos los nodos de G por su distancia a u. Hay $D(G) + 1$ niveles. En el primero está u y en el último v. Ningún nivel está vacío. Hay ejes de cada nivel al siguiente, y puede haberlos dentro de cada nivel, pero no hay ejes entre niveles no consecutivos (caso contrario, las distancias no serían las dadas). En G^c cada nodo de nivel i se relaciona con todos los nodos que no están en los niveles $i - 1$ e $i + 1$. Si se toma cualquier par de nodos, puede verse que las distancias están acotadas como se quiere.

Alternativamente, notemos que u y v no son adyacentes, y cualquier otro nodo no es adyacente a ambos. Eso implica que en el complemento u y v son adyacentes, y cualquier otro nodo es adyacente a al menos uno de ellos. Si tomamos dos nodos cualesquiera de G^c, su distancia es 1 sin son u y v, su distancia es a lo sumo 2 si uno es u o v, y el otro cualquiera distinto, y su distantica es a lo sumo 3 si no son ni u ni v.

(c) Si $D(G) \geq 3$, por el punto anterior debe ser $D(G) \leq 3$, lo que implica $D(G) = 3$. Por otro lado, no puede ser $D(G) = 1$, ya que G debería ser completo no trivial, que no es autocomplementario.

(d) $D(K_1) = 0$, $D(C_5) = 2$, y $D(P_4) = 3$.

NOTA: Propuesto originalmente por Federico Lebrón (aunque partes ya eran conocidas).

TOMADO: 2012C1R1 18-JUL-2012 (segundo punto y su contrarrecíproca sobre G^c), 2015C2R1 11-DIC-2015.

217. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Dado un grafo G, se define la distancia entre dos de sus vértices como el mínimo, sobre todos los caminos que van del primer al segundo vértice, de la cantidad de ejes que hay en el camino, o infinito si tales caminos no existen; además, se define $D(G)$ como el máximo, sobre todos los pares de vértices de G, de la distancia entre esos vértices (diámetro).

Dado un grafo no dirigido G, se define $g(G)$ como el mínimo, sobre todos los ciclos simples de G, de la cantidad de ejes que hay en el ciclo, o infinito si G no tiene ciclos simples (cintura o *girth*).

Por ejemplo, $D(K_4) = 1$ y $g(K_4) = 3$.

Sea G un grafo no dirigido de n vértices d-regular con $g(G) = 4$.

(a) Demostrar que $d \geq 2$ y $n \geq 2d$.

(b) Demostrar que si $n = 2d$ entonces $D(G) = 2$. (Observación: la recíproca es falsa.)

(c) Para cada $d \geq 2$ exhibir todos los G no isomorfos con $n = 2d$. Justificar.

SUGERENCIA: Para justificar $D(G) = \ell$ basta con encontrar una distancia que sea ℓ (lo que prueba $D(G) \geq \ell$), y demostrar que todas las distancias son a lo sumo ℓ (lo que prueba $D(G) \leq \ell$).

SOLUCIÓN:

(a) Para la primera desigualdad, si fuera $d \leq 1$ no habría ciclos y entonces sería $g(G) = \infty$.

Para la segunda, tomemos cualquier vértice r de G, y consideremos sus d adyacentes, los cuales no pueden ser adyacentes entre sí porque sería $g(G) = 3$. Por lo tanto, uno de los adyacentes, además de ser adyacente a r, debe ser adyacente a otros $d-1$ vértices nuevos. Como eventualmente podría haber otros vértices además de los mencionados tenemos $n \geq 1 + d + (d-1) = 2d$.

Alternativamente, consideremos un ciclo simple de 4 nodos, los cuales tienen 2 adyacentes en el ciclo, y no más que eso porque en tal caso no sería $g(G) = 4$. Por lo tanto, cada uno de esos nodos tiene $d-2$ adyacentes fuera del ciclo. Si tomamos dos nodos adyacentes en el ciclo, no pueden tener adyacentes en común, porque no sería $g(G) = 4$. Como eventualmente podría haber otros nodos además de los mencionados tenemos $n \geq 4 + 2(d-2) = 2d$.

(b) Si $n = 2d$ entonces en el punto anterior no hay más vértices que los mencionados. Por lo tanto, todos los vértices adyacentes a r son adyacentes a los mismos vértices. El grafo resultante es $K_{d,d}$ con $d \geq 2$. Es fácil comprobar que $D(G) = 2$. Un ejemplo donde no vale la recíproca es un grafo con $d = 3$, donde los adyacentes a r y sus adyacentes distintos de r forman una especie de `/|/|/|` donde cada vértice de las puntas de abajo también es adyacente a los dos del extremo opuesto. Para este grafo resulta $g(G) = 4$, $D(G) = 2$, $d = 3$, pero $n = 8 \neq 6 = 2d$.

(c) Por el punto anterior, es $K_{d,d}$ para cada $d \geq 2$.

NOTA: Propuesto originalmente por Leandro Montero. Es parecido al Ejercicio 218.

TOMADO: 2009C1P1 16-MAY-2009.

218. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Dado un grafo G, se define la distancia entre dos de sus vértices como el mínimo, sobre todos los caminos que van del primer al segundo vértice, de la cantidad de ejes que hay en el camino, o infinito si tales caminos no existen; además, se define $D(G)$ como el máximo, sobre todos los pares de vértices de G, de la distancia entre esos vértices (diámetro).

Dado un grafo no dirigido G, se define $g(G)$ como el mínimo, sobre todos los ciclos simples de G, de la cantidad de ejes que hay en el ciclo, o infinito si G no tiene ciclos simples (cintura o *girth*).

Por ejemplo, $D(K_4) = 1$ y $g(K_4) = 3$.

Sea G un grafo no dirigido de n vértices d-regular con $g(G) = 5$.

(a) Demostrar que $d \geq 2$ y $n \geq d^2 + 1$.

(b) Demostrar que si $D(G) = 2$ entonces $n = d^2 + 1$. (Observación: la recíproca también es cierta.)

(c) Para $d = 2$ y $d = 3$ exhibir todos los G no isomorfos con $D(G) = 2$. Justificar.

SUGERENCIA: Para justificar $D(G) = \ell$ basta con encontrar una distancia que sea ℓ (lo que prueba $D(G) \geq \ell$), y demostrar que todas las distancias son a lo sumo ℓ (lo que prueba $D(G) \leq \ell$).

SOLUCIÓN:

(a) Para la primera desigualdad, si fuera $d \leq 1$ no habría ciclos y entonces sería $g(G) = \infty$. Para la segunda, tomemos cualquier vértice r de G, y consideremos sus d adyacentes, los cuales no pueden ser adyacentes entre sí porque sería $g(G) = 3$. Cada uno de los adyacentes, además de ser adyacente a r, debe ser adyacente a otros $d - 1$ vértices, que no pueden coincidir con los de otro adyacente porque sería $g(G) = 4$. Como eventualmente podría haber otros vértices además de los mencionados tenemos $n \geq 1 + d + d(d - 1) = d^2 + 1$.

(b) Si $D(G) = 2$ entonces en el punto anterior no hay más vértices que los mencionados, porque habría vértices adyacentes a los nietos de r, que estarían a distancia 3 de r, de modo que sería $D(G) \geq 3$. La demostración de que la recíproca también es cierta es algo complicada, pero una parte sirve para argumentar el punto siguiente. No hay más nodos que los mencionados, y queremos probar $D(G) = 2$. Ya sabemos $D(G) \geq 2$ porque r está a distancia 2 de sus nietos. Por lo tanto, para probar $D(G) = 2$ basta ver que todas las distancias son a lo sumo 2. El vértice r está a distancia 1 de sus hijos, y a distancia 2 de sus nietos. Los hijos están a distancia 2 entre sí vía r. Cada hijo está a distancia 1 de sus propios hijos, y de sus sobrinos está a cierta distancia que vamos a analizar luego. Los nietos que son hermanos están a distancia 2 entre sí vía su padre (no son adyacentes entre sí porque sería $g(G) = 3$). Los primos (nietos de r de distinto padre) están a cierta distancia que vamos a analizar luego. Como r y sus hijos ya tienen grado d, cada nieto es adyacente (además de a su padre) a $d - 1$ vértices que son nietos de r. Pero no puede ser adyacente a sus hermanos porque sería $g(G) = 3$. Tampoco puede ser que un nieto sea adyacente a otros dos que sean hermanos, porque sería $g(G) = 4$. Por lo tanto, como cada nieto debe tener $d - 1$ adyacentes además de su padre, para cada hijo de r distinto de su padre, ese nieto es adyacente a uno sólo de sus hijos. Veamos ahora las distancias que nos faltaban. Tomemos cualquier hijo h de r y cualquiera de sus sobrinos s. Si bien h y s no son adyacentes, s es adyacente a algún hijo de h, por lo que h y s están a distancia 2. Tomemos ahora dos nietos de r que sean primos, digamos p_1 y p_d. Sean h_1 y h_d sus padres (hijos de r), y sean $h_2, h_3, \ldots h_{d-1}$ los otros hijos de r. Sea v_1 el hijo de h_1 que es adyacente a p_d. Sean $w_2, w_3, \ldots w_{d-1}$ los hijos respectivamente de $h_2, h_3, \ldots h_{d-1}$ que son adyacentes a p_d. Sean $v_2, v_3, \ldots v_{d-1}$ los hijos de h_1 que son respectivamente adyacentes a $w_2, w_3, \ldots w_{d-1}$. Los vértices $v_1, v_3, \ldots v_{d-1}$, todos hijos de h_1 deben ser distintos entre sí porque caso contrario sería $g(G) = 3$. Como h_1 tiene $d - 1$ hijos, para algún i es $v_i = p_1$, por lo que la distancia entre p_1 y p_d es a lo sumo 2.

(c) Por el punto anterior, cada nieto de r es adyacente $d - 1$ primos que no sean hermanos entre sí. En base a eso se pueden armar los dos grafos. Para $d = 2$ resulta C_5, y otro grafo más complicado para $d = 3$ (que resulta ser el grafo de Petersen).

NOTA: Propuesto originalmente por Leandro Montero. Es parecido al Ejercicio 217.

TOMADO: 2009C1R1 19-AGO-2009.

219. Sea $r_1, r_2, \ldots r_n$ una secuencia de n números reales ($n \in \mathbb{N}$). Diseñar un algoritmo eficiente que elimine la mínima cantidad posible de números de la secuencia de manera tal que los números que permanezcan queden ordenados de manera creciente. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n^2)$.

SOLUCIÓN: Armar un grafo con un vértice por cada número. Poner un arco dirigido de un vértice a otro si y sólo si el primero puede preceder al segundo en la secuencia definitiva. Agregar un vértice inicial y un arco desde el mismo a cada vértice con grado de entrada 0. Agregar un vértice final y un arco hacia el mismo desde cada vértice con grado de salida 0. Buscar el camino máximo desde el vértice inicial al vértice final. Los vértices asociados a números que pertenecen al camino máximo son los que permanecen en la secuencia definitiva. Todos los ejes van hacia "adelante", de modo que el grafo no tiene circuitos. El camino máximo puede hallarse con un algoritmo para camino máximo específico para grafos sin circuitos, que es $O(m) = O(n^2)$, o con un algoritmo cualquiera para camino máximo. Otra forma es considerar un vector de largo n, y ejecutar el algoritmo para camino máximo sobre el vector; la posición i del vector contiene la longitud de la subsecuencia más larga que termina en el i-ésimo número; las posiciones se calculan en orden; a cada posición se le asigna 1 más el máximo del conjunto de posiciones anteriores que están asociadas a números que pueden preceder en la subsecuencia definitiva al número actual (si el conjunto es vacío se asigna 1, o se considera un primer número ficticio que pueda preceder a todos los números); la solución es el máximo de todo el vector (o también se puede agregar un último número ficticio que pueda ser precedido por cualquiera.

Este problema es bastante conocido y puede pensarse en términos de programación dinámica, como aparece en el Ejercicio 40.

NOTA: Basado en 1.3 y 1.4 de Manber. No cambia el ejercicio si se pide orden decreciente, o no creciente, o no decreciente.

TOMADO: 2005C2P1 15-OCT-2005.

220. Se tiene una red de puertos conectados por cables, que se utilizan para mandar información de un puerto a otro, pasando eventualmente por puertos intermedios. El tiempo que demora la información para atravesar cada cable es proporcional al grosor del cable y a su longitud. Es decir, si un cable tiene grosor g_i y longitud l_i, el tiempo es $g_i \times l_i$. Sabemos que en algún momento no muy lejano, se van a cambiar *todos* los cables que conectan con el puerto 1 porque es el más antiguo y está en mal estado. Queremos saber por dónde mandar la información para que llegue lo antes posible desde un puerto cualquiera a otro cualquiera, tanto antes como luego del cambio de cables. Diseñar un algoritmo basado en grafos que resuelva el problema planteado. Cada etapa del algoritmo debe ser eficiente. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Calcular caminos mínimos usando como pesos $g_i \times l_i$, todos contra todos excepto el nodo 1, con Floyd o Dantzig. Luego agregar el nodo 1 con Dantzig, y reagregarlo también con Dantzig para los cables modificados. El primer cálculo es $O(n^3)$, y los otros pasos son $O(n^2)$ cada uno, donde n es la cantidad de nodos. Habría que analizar el costo de armar el grafo de acuerdo al formato de entrada.

NOTA: Propuesto originalmente por Marina Groshaus.

TOMADO: 2012C2R1 12-DIC-2012.

Control de proyectos

221. Dados los siguientes datos sobre un proyecto a realizar, usar el grafo de tareas en los

Variante 1: vértices

Variante 2: ejes

para determinar las tareas críticas y el tiempo mínimo que se necesita para completar el proyecto. Justificar.

Tarea	Duración	Requisitos
A	3	I, D
B	2	-
C	13	F
D	7	B
E	11	A
F	13	-
G	5	E, H
H	7	A
I	11	B

NOTA: Aparece en mi apunte de control de proyectos.

TOMADO: 2003C1P2 11-JUL-2003 (Variante 1), 2005C1R1 22-JUL-2005 (Variante 2).

Combinados I

222. Sea G un grafo conexo de n vértices que tiene k

Variante 1: ciclos simples,

Variante 2: árboles generadores,

con $k \in O(1)$.

(a) Demostrar que G tiene $O(n)$ ejes.

(b) Demostrar que si G tiene longitudes no negativas asociadas a sus ejes y está representado mediante listas de adyacentes, entonces es posible calcular los caminos mínimos desde un vértice dado hacia todos los vértices con complejidad $O(n \log n)$.

SUGERENCIA: Usar el primer punto.

(c) Demostrar que en el caso extremo en que

Variante 1: $k = 0$

Variante 2: $k = 1$

el problema del punto anterior puede resolverse con complejidad $O(n)$.

SUGERENCIA: No usar el primer punto.

SOLUCIÓN:

(a) Sea m la cantidad de ejes de G.

Para la Variante 1, el grafo tiene al menos $m - (n - 1)$ ciclos simples según el Ejercicio 157. Es decir, $k \geq m - (n - 1)$, o equivalentemente $m \leq k + n - 1 = O(n)$.

Para la Variante 2 se usa la misma idea que en el ejercicio mencionado. Como G es conexo, tiene al menos un árbol generador T. Para cada eje e que no está en T, si lo agregamos a T se forma un único ciclo que contiene a e. Si ahora sacamos cualquier eje de ese ciclo que no sea e, volvemos a obtener un árbol generador T_e, que es distinto a todos los otros árboles generadores mencionados porque contiene al eje e. Como eventualmente el grafo tiene otros árboles generadores, resulta $k \geq 1 + m - (n - 1)$, o equivalentemente $m \leq k + n - 2 = O(n)$.

Alternativamente para la Variantes 2, podemos decir que al resolver el Ejercicio 6.11.b de la Práctica se llega a la conclusión de que el conjunto de ejes de G coincide con la unión sobre sus k árboles generadores de los conjuntos de ejes de los mismos. Como cada uno de esos árboles generadores tiene $n - 1$ ejes, no necesariamente disjuntos, tenemos $m \leq k(n - 1) = O(n)$.

(b) En las condiciones del enunciado, el algoritmo de Dijkstra implementado con un heap tiene complejidad $O((m + n) \log n)$, que es $O(n \log n)$ cuando m es $O(n)$.

(c) Para la Variante 1, el grafo es conexo y sin ciclos, es decir un árbol. Para la Variante 2, el grafo es un árbol según el Ejercicio 6.11.a de la Práctica. En cualquier caso el grafo es un árbol, y pueden calcularse los caminos mínimos en $O(n)$ según el Ejercicio 214.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez. El grafo puede no ser conexo pero no aportaría mucho.

TOMADO: 2016C2P1 01-OCT-2016 (Variante 2).

Euleriano

223. Determinar para qué valores de n, p, q y h los siguientes grafos tienen camino (abierto) euleriano. Justificar.

(a) K_n

(b) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(c) $K_{p,q}$

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

SOLUCIÓN: Como los grafos son conexos, basta ver que hay exactamente dos nodos de grado impar.

K_n: Todos los nodos tienen el mismo grado, de modo que debe ser $n = 2$, y K_2 cumple lo pedido.

W_n: Ningún n sirve, ya que siempre hay al menos 3 nodos de grado 3.

$K_{p,q}$: Los nodos tienen grado p o q, de modo que al menos uno de los dos debe ser impar. Si uno sólo es impar, el otro debe ser 2, y así tenemos $K_{2,n}$ y $K_{n,2}$ con n impar. Si ambos son impares, su suma debe ser dos para que haya dos nodos de grado impar, y resulta $K_{1,1}$.

ABC h: Para $h = 0$ no hay dos nodos. El valor $h = 1$ sirve. Los valores $h >= 2$ no sirven ya que hay al menos 4 hojas que tienen grado 1.

NOTA: Encontrado originalmente en Internet (circuito euleriano y hamiltoniano). No se incluye C_n (trivial), P_n (trivial), A n (variable), ni GBC n m (variable).

TOMADO: 2016C1R2 15-JUL-2016.

224. Determinar para qué valores de n, p, q y h los siguientes grafos tienen circuito euleriano. Justificar.

(a) K_n

(b) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(c) $K_{p,q}$

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

SOLUCIÓN: Como los grafos son conexos, basta ver que todos los nodos tienen grado par.

K_n: En K_n los nodos tienen grado $n - 1$, que es par si y sólo si n es impar.

W_n: Ningún n sirve, ya que siempre hay nodos de grado impar.

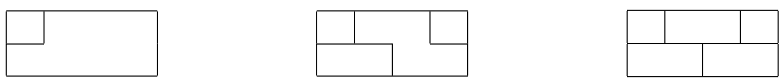
$K_{p,q}$: En $K_{p,q}$ los nodos tienen grado p o q, de modo que ambos deben ser pares.

ABC h: El valor $h = 0$ sirve. Los valores $h >= 1$ no sirven ya que hay al menos 2 hojas que tienen grado 1.

NOTA: Encontrado originalmente en Internet (circuito euleriano y hamiltoniano). No se incluye C_n (trivial), P_n (trivial), A n (variable), ni GBC n m (variable).

TOMADO: 2005C1R2 08-AGO-2005 (K_n y $K_{p,q}$).

225. Decidir para cada dibujo de la figura si existe una curva que cruce exactamente una vez a cada uno de los segmentos y no pase por ningún punto en el que se tocan dos o más segmentos. En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar. La curva puede ser abierta o cerrada, y también puede cruzarse a sí misma.



SOLUCIÓN: Modelar cada dibujo con un vértice por cada región (incluyendo la exterior), y un eje por cada segmento compartido por dos regiones. El grado de cada vértice es la cantidad de segmentos en el borde de la región. Existe la curva buscada si y sólo si el (pseudo/multi) grafo tiene un camino o circuito euleriano. El primer dibujo produce un grafo conexo con 3 vértices de grados 6, 4 y 6, de modo que la curva existe y se la puede dibujar. El segundo dibujo produce un grafo conexo con 5 vértices de grados 9, 4, 4, 5 y 8, de modo que la curva existe y se la puede dibujar. El tercer dibujo produce un grafo conexo con 6 vértices de grados 9, 4, 4, 5, 5 y 5, de modo que la curva no existe ya que hay más de dos vértices de grado impar.

Notar que cuando la curva existe, siempre se la puede dibujar abierta. La curva cerrada ocurre cuando el grafo tiene un circuito euleriano, pero como los vértices del grafo representan regiones del dibujo, es posible abrir la curva en cualquiera de las regiones. En un grafo tal cosa no es posible porque los vértices no tienen dimensión.

NOTA: Encontrado originalmente en el apunte de Lucas Andisco para la Selección para la 27ª Olimpíada Matemática del Cono Sur (2016).

TOMADO: 2016C1P2 02-JUL-2016.

226. Sea $G = (V, E)$ un grafo no necesariamente conexo con todos sus vértices de grado par. Sea $\mathbb{S} = \{C_1, C_2, \ldots C_k\}$ un conjunto de circuitos simples de G que no comparten ejes. Demostrar que existe una partición en circuitos simples de E la cual incluye a $\mathbb{S}$.

SOLUCIÓN: Si a G le sacamos todos los ejes que aparecen en $\mathbb{S}$ obtenemos un grafo con todos los vértices de grado par, por lo que existe una partición en circuitos simples de los ejes de cada una de sus componentes conexas, que junto con $\mathbb{S}$ forman una partición en circuitos simples de E.

TOMADO: 2015C2P2 30-NOV-2015 (para $k = 1$).

227. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

(a) Demostrar que si G es un grafo d-regular con una cantidad impar de vértices entonces G o G^c tienen circuito euleriano.

SUGERENCIA: Usar que si un grafo no es conexo entonces su complemento lo es.

(b) Mostrar que la propiedad no es cierta para una cantidad par de vértices. Concretamente, exhibir un grafo d-regular H con una cantidad par de vértices tal que H y H^c (tengan ejes y) no tengan circuito euleriano. Justificar.

SOLUCIÓN: Sea n la cantidad de vértices del grafo.

(a) Como la sumatoria de grados es par, d debe ser par, es decir, G tiene todos los grados pares. Según el Ejercicio 49, el complemento es $n - 1 - d$-regular, por lo que G^c también tiene todos los grados pares. Si G es conexo como tiene todos los grados pares es euleriano. Si G no es conexo, G^c es conexo, y como tiene todos los grados pares es euleriano.

(b) El problema para encontrar H es que si n es par entonces H tiene grados pares y H^c impares, o viceversa. Como no es posible evitar en ambos lados grados pares, lo que podemos hacer es que el grafo con grados pares no sea conexo. Por ejemplo, podemos tomar $H = 2K_3$ que no es euleriano porque tiene ejes y no es conexo, y su complemento tampoco es euleriano porque tiene grados impares.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2014C1P2 07-JUL-2014.

228. Sea G un grafo euleriano de 3 o más vértices, no necesariamente conexo. Demostrar que G tiene 3 o más vértices del mismo grado.

SOLUCIÓN: Es parecido al Ejercicio 5.4 de la Práctica que dice que en todo grafo no trivial hay un grado repetido. Sea n la cantidad de vértices de G. El grado mínimo es 0 y el grado máximo es $n - 1$, pero por ser euleriano todos los grados deben ser pares. Por lo tanto, si n es impar hay $(n + 1)/2$ grados posibles: $0, 2, \ldots n - 1$; sin embargo, no puede haber a la vez vértices de grado 0 y de grado $n - 1$, de manera que hay $(n - 1)/2$ grados posibles, que es menos que la mitad de n, por lo que algún grado debe aparecer al menos 3 veces. Si n es par hay $n/2$ grados posibles: $0, 2, \ldots n - 2$; para que no haya 3 vértices del mismo grado, cada grado debe aparecer exactamente 2 veces; sin embargo, si hay 2 vértices de grado 0 no puede haber vértices de grado $n - 2$.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2011C2P2 05-DIC-2011.

229. Sea G un grafo no necesariamente conexo de n vértices y m ejes. Diseñar un algoritmo eficiente que decida si G tiene un circuito euleriano. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(m + n)$.

SOLUCIÓN: Con BFS calcular las componentes conexas y simultáneamente los grados (si no están ya calculados). El grafo es euleriano si y sólo si todos los grados son pares y tiene a lo sumo una componente conexa con ejes. La complejidad es la de BFS, que con listas de adyacentes es la mencionada.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2017C2P2 01-DIC-2017.

230. Sea G un grafo conexo donde cada eje e tiene asociado un número $k_e \in \mathbb{N}$. Diseñar un algoritmo eficiente que decida si existe en el grafo un circuito que para cada eje e pasa exactamente k_e veces por ese eje. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad cuando se utiliza una estructura de datos razonable para almacenar el grafo. Justificar.

    SOLUCIÓN: Una estructura razonable es listas de adyacentes, donde además del nodo adyacente se indica el valor asociado al eje que lo une con el titular de la lista. Un posible algoritmo es sumar para cada nodo los k_e de los ejes incidentes sobre ese nodo. El circuito pedido existe si y sólo si la suma es par para todos los nodos. La demostración se basa en poner k_e copias de cada eje y usar el resultado para circuito euleriano. Notar que se necesita $k_e \geq 1$ para que el grafo modificado sea conexo. La complejidad es $O(m + n) = O(m)$ ya que el grafo es conexo, donde m es la cantidad de ejes y n es la cantidad de nodos.

    NOTA: Propuesto originalmente por Pablo Heiber. Es casi igual al Ejercicio 231.

    TOMADO: 2010C2P2 06-DIC-2010.

231. El gobierno de cierta ciudad está preocupado porque dentro de dos días es el fin del mundo y la ciudad todavía sigue intacta. El intendente ha decidido empezar la destrucción ahora mismo. 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 tiene cierto número de carriles. Se puede ir de cualquier esquina a cualquier otra recorriendo las cuadras, pasando eventualmente por esquinas intermedias. El intendente planea enviar un tractor equipado con un arado para que recorra cada carril de cada cuadra, en un sentido o en el otro, destruyendo el asfalto a su paso. El problema es que una vez destruido un carril, el tractor no puede volver a circular por el mismo. Además, el tractor no es muy maniobrable, y no puede cambiar de carril cuando se encuentra en el medio de una cuadra. El tractor está ubicado inicialmente en la esquina 1 (la intendencia), y puede terminar su trabajo en cualquier lugar (aunque necesariamente va a ser en una esquina). Diseñar un algoritmo eficiente que decida si es posible llevar a cabo el plan del intendente. La entrada del algoritmo es la cantidad n de esquinas, la cantidad m de cuadras, y para cada cuadra las esquinas que conecta y su cantidad de carriles. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

    SOLUCIÓN: Podemos modelar la ciudad con un multigrafo que tiene un nodo por cada esquina, y tantos ejes entre cada par de esquinas como carriles tiene la cuadra entre ellas (si existe). El problema consiste en determinar si ese multigrafo tiene un circuito euleriano, o un camino euleriano que tenga por extremo al nodo 1. De acuerdo al enunciado, el multigrafo armado resulta conexo, de modo que para decidir si hay circuito o camino euleriano basta analizar si todos los grados son pares en el primer caso, o si el nodo 1 y algún otro son los únicos de grado impar. Eso puede hacerse en $O(m + n) = O(m)$ simplemente calculando los grados de todos los nodos a partir de la lista de cuadras.

    NOTA: Propuesto originalmente por Michel Mizrahi. Es casi igual al Ejercicio 230.

    TOMADO: 2012C2R2 19-DIC-2012.

232. Sea G un grafo no necesariamente conexo con todos sus vértices de grado par. Demostrar que los ejes de G se pueden orientar de manera tal que cada vértice tenga el mismo grado de entrada que de salida.

    SOLUCIÓN: Basta demostrarlo para cada componente conexa. Tomemos una cualquiera, que resulta un grafo conexo con todos sus vértices de grado par. Por lo tanto existe una partición de sus ejes en circuitos simples. Si orientamos cada circuito con todos sus ejes en el mismo sentido, cada circuito aporta 1 tanto al grado de entrada como al grado de salida de los vértices por los que pasa, por lo que finalmente los grados coinciden en cada vértice.

    NOTA: Basado en 7.24 de Manber.

    TOMADO: 2005C1P2 15-JUL-2005, 2013C1P2 10-JUL-2013.

233.

Variante 1: Sea G un grafo conexo que contiene $2k$ vértices de grado impar, donde $k \in \mathbb{N}$ (el grafo podría contener otros vértices, pero de grado par). Demostrar que los ejes de G pueden particionarse en k caminos sin ejes repetidos. ¿Pasa lo mismo si G no es conexo? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

Variante 2: Sea G un grafo no necesariamente conexo formado por $2k$ vértices de grado impar, donde $k \in \mathbb{N}$ (el grafo no contiene otros vértices). Demostrar que los ejes de G pueden particionarse en k caminos sin ejes repetidos.

SOLUCIÓN: Agregar un vértice a G y hacerlo adyacente a los $2k$ vértices de grado impar. Queda un grafo conexo con todos vértices de grado par, que admite un circuito euleriano, el cual pasa k veces por el vértice agregado. Cada vez que se pasa por ahí se puede extraer un subcircuito, el cual determina un camino cuando se remueve el vértice agregado.

Para la Variante 1 con G no conexo, supongamos que G es K_3 junto con K_2. Es claro que los ejes de G no forman un camino ($k = 1$).

TOMADO: 2001C2P2 11-DIC-2001 (Variante 1), 2004C1R2 09-AGO-2004 (Variante 2).

234. Un corte de un grafo es un conjunto de ejes del grafo tal que al removerlos se obtiene un grafo con más componentes conexas que el original.

Sea G un grafo que tiene un circuito euleriano. Demostrar que cualquier corte minimal de G tiene un número par de ejes.

SOLUCIÓN: Supongamos sin pérdida de generalidad que G es conexo, ya que si no lo fuera, la única posibilidad es que a lo sumo una de sus componentes conexas tenga ejes, y las otras sean vértices ailados; los vértices aislados siguen formando las mismas componentes conexas luego de remover cualquier subconjunto de ejes (además a veces ser conexo forma parte de la definición de euleriano). Si G es conexo, puede demostrarse que un corte minimal produce un grafo con dos componenetes conexas. Tomemos cualquier corte minimal y consideremos las dos componentes conexas que produce. Por ser el corte minimal, sus ejes en el grafo original van de una componente a la otra (no conectaban nodos de la misma componente conexa). Consideremos un circuito euleriano del grafo y comencemos a recorrerlo desde cualquier vértice de una de las componentes conexas. Al recorrer cada eje del corte, se pasa de una componente a la otra. Como el circuito comienza y termina en la misma componente conexa, la cantidad de ejes del corte debe ser par.

NOTA: Analizar si puede extenderse a grafos no necesariamente conexos con todos sus nodos de grado par.

TOMADO: 2001C2R2 26-DIC-2001, 2012C1R2 08-AGO-2012.

235. Sea $G = (V, E)$ un grafo euleriano.

(a) Demostrar que si G es arbitrariamente atravesable desde alguno de sus vértices entonces toda partición de E en circuitos simples contiene la misma cantidad de circuitos simples.

(b) ¿Vale la recíproca del primer punto? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

(c) ¿Sigue valiendo la propiedad del primer punto si G es sólo euleriano en vez de arbitrariamente atravesable desde alguno de sus vértices? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Por ser G euleriano, el grado de cualquier nodo es dos veces la cantidad de circuitos simples que pasan por ese nodo en cualquier partición en circuitos simples de E. Por ser G arbitrariamente atravesable desde algún vértice v, todos los circuitos simples de G pasan por v. Consideremos cualquier partición de E en circuitos simples. Por el segundo argumento todos los circuitos de esa partición pasan por v, y por el primer argumento la cantidad que hay es la mitad del grado de v. El resultado también surge de manera implícita de la demostración del Ejercicio 8.7.b de la Práctica.

(b) No vale; un contraejemplo es una secuencia de tres círculos con cada par consecutivo compartiendo un vértice; hay una única partición formada por los tres círculos, pero el grafo no es arbitrariamente atravesable porque ningún nodo pertenece a todos ellos.

(c) No vale; un contraejemplo es tres círculos que se intersectan de a pares y entre los tres; el grafo es euleriano pero existe una partición que contiene tres circuitos simples y otra que contiene cuatro.

TOMADO: 2003C1P2 11-JUL-2003 (excepto el segundo punto), 2017C1R2 21-JUL-2017.

236. Sea $G = (V, E)$ un grafo euleriano. Sean v y w dos vértices de G.

   (a) Demostrar que si G es arbitrariamente atravesable desde v y desde w entonces $d(v) = d(w)$.

   (b) ¿Vale la recíproca del primer punto? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

   (c) Demostrar que si $d(v) = d(w)$, entonces G es arbitrariamente atravesable desde v si y sólo si lo es desde w.

   SUGERENCIA: Dado cualquier ciclo simple C, encontrar una partición P de E tal que $C \in P$.

   SOLUCIÓN:

   (a) Por el Ejercicio 8.7.b de la Práctica los dos grados son máximos, y por lo tanto iguales.

   (b) No vale. No tiene sentido. Un contraejemplo es K_5.

   (c) Ida y vuelta son iguales, intercambiando los papeles de v y w. Lo que sigue es la ida. Por el Ejercicio 8.7.a de la Práctica sabemos que v está en todo ciclo simple, y también sabemos que basta ver que w lo está. Sea C cualquier ciclo simple de G. Sea P una partición de E en ciclos simples tal que $C \in P$ (esto siempre es posible según el Ejercicio 226). El grado de cada vértice es dos veces la cantidad de ciclos de P en los cuales está el vértice. Como v está en todo ciclo simple de G, en particular está en todos los de P, y por lo tanto $d(v) = 2|P|$. Dado que $d(v) = d(w)$, resulta $d(w) = 2|P|$, lo que significa que w está en todos los ciclos de P, en particular en C.

   NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

   TOMADO: 2015C2R2 18-DIC-2015.

237. Sea G un grafo. Demostrar que son equivalentes:

   (a) G tiene ejes, y es arbitrariamente atravesable desde todos sus vértices.

   (b) G no tiene vértices aislados, y es arbitrariamente atravesable desde al menos tres vértices distintos.

   (c) G tiene circuitos simples, y todos ellos son circuitos hamiltonianos de G.

   (d) G es C_n (ciclo simple de $n \geq 3$ vértices).

   SUGERENCIA: Salen (237b) $\Rightarrow$ (237c), (237b) $\Rightarrow$ (237d), y (237d) $\Rightarrow$ (cualquiera), entre otras.

   SOLUCIÓN: Vamos a usar que si G es euleriano entonces existe una partición de los ejes en circuitos simples. Eso en principio lo hemos visto para grafos conexos, lo cual puede verse que vale en este caso, pero de todos modos también vale en el caso no conexo. Además, si G tiene ejes esa partición no es vacía.

(237a) $\Rightarrow$ (237b): G no tiene vértices aislados: Si tuviera vértices aislados no sería arbitrariamente atravesable desde ellos, ya que no sería posible recorrer los ejes que por hipótesis existen.

G es arbitrariamente atravesable desde al menos tres vértices distintos: Basta ver que G tiene al menos tres vértices distintos, lo cual ocurre porque al haber ejes, el circuito euleriano tiene que pasar por al menos tres vértices distintos.

(237a) $\Rightarrow$ (237c): G tiene circuitos simples: Como G es arbitrariamente atravesable en particular es euleriano, por lo que existe una partición de los ejes en circuitos simples. Tal partición no es vacía porque G tiene ejes.

Todos los circuitos simples de G son hamiltonianos: Sea C cualquier ciclo simple de G. Por el Ejercicio 8.7.a de la Práctica cada vértice desde el cual G es arbitrariamente atravesable debe estar en todos los ciclos simples de G. Por lo tanto, todos los vértices de G están en C, que resulta hamiltoniano.

(237a) $\Rightarrow$ (237d): Probablemente similar a (237a) $\Rightarrow$ (237c).

(237b) $\Rightarrow$ (237a): Pendiente, aunque innecesario.

(237b) ⇒ (237c), (237d): Sean v_1, v_2 y v_3 tres vértices distintos desde los cuales G es arbitrariamente atravesable. Como G es arbitrariamente atravesable en particular es euleriano, por lo que existe una partición de los ejes en circuitos simples. Tal partición no es vacía porque G tiene ejes, ya que no hay vértices aislados. Sea $C \in P$. Por el Ejercicio 8.7.a de la Práctica cada vértice desde el cual G es arbitrariamente atravesable debe estar en todos los ciclos simples de G. Por lo tanto, v_1, v_2 y v_3 están en C. Si P tuviera otro circuito C', también v_1, v_2 y v_3 deberían estar en C', y habría algún otro circuito simple en G que no contendría a los tres vértices. Por lo tanto, no hay otros circuitos en P, lo cual implica que todos los ejes de G están en C. Además, no puede haber otros nodos fuera de C, ya que por hipótesis todos los nodos tienen ejes incidentes, y como acabamos de ver todos los ejes de G están en C. En definitiva, C es el único ciclo de G y es hamiltoniano por definición, lo que equivale a decir que G es C_n.

(237c) ⇒ (237a), (237b): Como G tiene circuitos simples hamiltonianos, entonces tiene al menos tres vértices, tiene ejes y no tiene vértices aislados. Vamos a ver que G es arbitrariamente atravesable desde cada uno de sus vértices (que son al menos tres). Por el Ejercicio 8.7.a de la Práctica basta ver que cada vértice está en todo circuito simple de G, lo cual es cierto porque por hipótesis todos los circuitos simples son hamiltonianos.

(237c) ⇒ (237d): Sea C cualquier circuito simple de G, el cual por hipótesis existe y es hamiltoniano. Por lo tanto, no hay más vértices fuera de C, y sólo resta ver que no hay más ejes en G que los de C. Si hubiera cualquier otro eje, debería necesariamente conectar dos nodos de C, lo que produciría dos ciclos no hamiltonianos, contradiciento la hipótesis.

(237d) ⇒ (cualquiera): Trivial.

TOMADO: 2015C1P2 04-JUL-2015.

238. (a) Sea G un grafo. Demostrar que G es C_n (ciclo simple de $n \geq 3$ vértices) si y sólo si G tiene un cubrimiento de vértices por aristas y es arbitrariamente atravesable desde al menos tres vértices distintos.

(b) Mostrar que para la vuelta la cota de tres vértices no puede ser mejorada. Concretamente, exhibir un grafo que tenga un cubrimiento de vértices por aristas, sea arbitrariamente atravesable desde al menos dos vértices distintos, y no sea C_n. Justificar.

SOLUCIÓN:

(a) La ida es trivial. La vuelta es la parte (237b) ⇒ (237d) del Ejercicio 237.

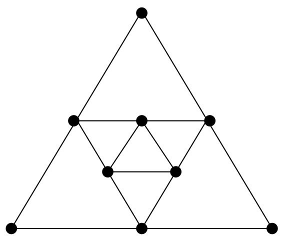
(b) Dos ciclos que coinciden en exactamente dos nodos (tipo dos conjuntos con su intersección).

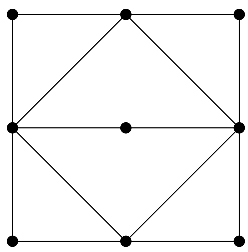
NOTA: Es mayormente una parte del Ejercicio 237 (el primer punto es una de las equivalencias de ese ejercicio). Lo del cubrimiento es una forma complicada de decir que no tiene vértices aislados.

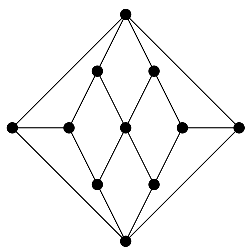
TOMADO: 2015C1R2 17-JUL-2015.

239. (a) Determinar para cada uno de los grafos mostrados cuál es la mínima cantidad de ejes (eventualmente 0) que es necesario agregarle para obtener un grafo simple que tenga un camino (abierto) euleriano. En caso de que no pueda lograrse que lo tenga, indicarlo. Exhibir un grafo conexo para el cual no pueda lograrse que lo tenga, o demostrar que tal grafo no existe. Justificar.

(b) Repetir para circuito euleriano.







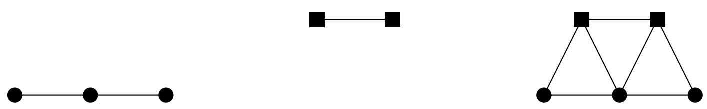
SOLUCIÓN:

(a) Hay que lograr que haya exactamente 2 nodos de grado impar, eliminando los que sobran, o creandolos si no hubiera ninguno. En el primer grafo no hay nodos de grado impar, por lo que hay que crearlos agregando un eje entre dos nodos no adyacentes. En el segundo grafo no hace falta agregar nada. En el tercer grafo hay que agregar 3 ejes para eliminar 6 nodos de grado impar. No puede lograrse por ejemplo para K_4, porque sus 4 nodos de grado impar ya son adyacentes entre sí.

(b) Hay que agregar ejes entre nodos no adyacentes de grado impar, de modo que todos los nodos tengan grado par. En el primer grafo no hace falta agregar nada, en el segundo grafo hay que agregar un eje entre los 2 nodos de grado impar, y en el tercero hay que agregar 4 ejes entre nodos de grado impar. No puede lograrse por ejemplo para K_4, porque sus 4 nodos de grado impar ya son adyacentes entre sí.

TOMADO: 2010C2R2 22-DIC-2010, 2019C1R2 19-JUL-2019.

240. 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 $T(G)$.



Demostrar que si G es un grafo conexo y con ejes entonces son equivalentes:

(a) Todos los grados de los vértices de G tienen la misma paridad (son todos pares o todos impares).

(b) $L(G)$ tiene un circuito euleriano.

(c) $T(G)$ tiene un circuito euleriano.

SOLUCIÓN:

(240a) $\Rightarrow$ (240b): Basta ver que $L(G)$ es conexo y todos sus vértices tienen grado par. $L(G)$ es conexo porque G lo es (demostrarlo). Cada vértice $e = (v, w)$ de $L(G)$ tiene grado $d(v) + d(w) - 2$, que es par si todos los grados en G tienen la misma paridad.

(240a) $\Rightarrow$ (240c): Basta ver que $T(G)$ es conexo y todos sus vértices tienen grado par. $T(G)$ es conexo porque G lo es (demostrarlo). Cada vértice $e = (v, w)$ de $T(G)$ (correspondiente a un eje de G) tiene grado $d(v) + d(w)$, mientras que cada vértice x de $T(G)$ (correspondiente a un vértice de G) tiene grado $2d(x)$. Ambos tipos de valores son pares si todos los grados en G tienen la misma paridad.

(240b) $\Rightarrow$ (240a): Si $L(G)$ es euleriano, tiene todos los vértices de grado par. Eso significa que para cada eje $e = (v, w)$ de G, se cumple que $d(v) + d(w) - 2$ es par, lo que equivale a decir que $d(v)$ y $d(w)$ tienen la misma paridad. Como G es conexo, hay un camino entre cualquier par de vértices de G, y por lo anterior la paridad se mantiene a lo largo de cualquier camino entre ellos, lo que implica que todos los grados tienen la misma paridad.

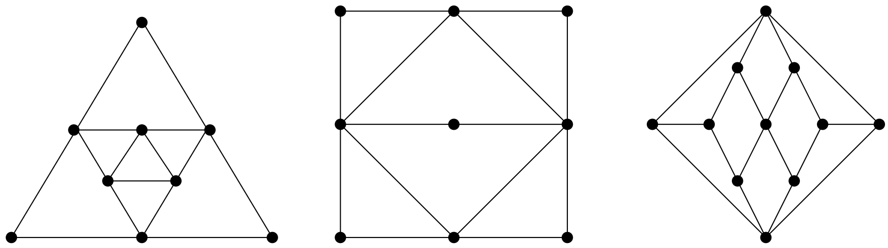
(240c) $\Rightarrow$ (240a): Análogo a (240b) $\Rightarrow$ (240a), usando $d(v) + d(w)$.

NOTA: Propuesto originalmente por Leandro Montero. Aparece en el libro de Harary. Si G no es conexo, aparentemente vale el resultado análogo considerando cada componente conexa de G, $L(G)$ y $T(G)$ por separado (por ejemplo, si $G = K_3 + K_4$.)

TOMADO: 2009C1R2 26-AGO-2009.

Euleriano y hamiltoniano

241. Decidir para cada uno de los grafos mostrados si tiene el camino o circuito indicado. Justificar.



(a) Camino (abierto) euleriano.
(b) Circuito euleriano.
(c) Camino (abierto) hamiltoniano.
(d) Circuito hamiltoniano.

SOLUCIÓN:

(a) Únicamente el segundo grafo, que es el único que tiene exactamente dos vértices de grado impar (y es conexo).
(b) Únicamente el primer grafo, que es el único que tiene todos sus vértices de grado par (y es conexo).
(c) El primer y el segundo grafo tienen, empezando en una esquina, haciendo primero el borde, y luego lo de adentro. El tercer grafo tiene, empezando y terminando en las puntas izquierda y derecha, y en el resto haciendo una especie de S.
(d) El primer y el segundo grafo no tienen, ya que hay una única manera de incluir los nodos de grado 2, lo cual forma un circuito cerrado que luego no permite agregar a los otros nodos. El tercer grafo es bipartito (por ejemplo porque su número cromático es 2, debido a que tiene ejes, es planar y toda región tiene una cantidad par de aristras frontera); como la cantidad de nodos es 11 (impar), no puede ser hamiltoniano. Alternativamente, el primer grafo no tiene porque al sacar los 3 nodos del medio de los lados del triángulo más grande se producen 4 componentes conexas, el segundo grafo no tiene porque al sacar los 2 nodos adyacentes al nodo central se producen 3 componentes conexas, y el tercer grafo no tiene porque al sacar los 3 nodos del medio junto con el de arriba y el de abajo se producen 6 componentes conexas.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2008C1P2 11-JUL-2008.

242. Un puente de un grafo es un eje del mismo tal que al removerlo se obtiene un grafo con más componentes conexas.

(a) Demostrar que si G tiene un circuito euleriano entonces no tiene puentes.
(b) Demostrar que si G tiene un circuito hamiltoniano entonces no tiene puentes.

SOLUCIÓN: Consideremos cualquier eje de G. Al removerlo los caminos que lo usaban pueden rearmarse para usar el camino euleriano o hamiltoniano que quedó (salvo para hamiltoniano y $G = K_2$, si K_2 fuera hamiltoniano). Los caminos en otras componentes conexas no se ven afectados (aunque en realidad esos caminos no existen). Para euleriano, otra forma de demostrarlo es notar que al remover el eje, quedarían componentes conexas con exactamente un nodo de grado impar, por lo que la sumatoria de grados en cada componente conexa no sería par, lo cual no puede ocurrir porque en cualquier grafo es 2 veces la cantidad de ejes. Para hamiltoniano, otra forma de demostrarlo es usando esa propiedad que dice que si sacamos un conjunto de nodos y se obtienen más componentes conexas que la cantidad de nodos, el grafo no es hamiltoniano; en este caso, habría que sacar un extremo del eje, del lado en que hay otros nodos además del que sacamos (para que se forme una componente conexa de ese lado).

NOTA: Encontrado originalmente en Internet. En vez de euleriano puede pedirse que todos los vértices tengan grado par.

TOMADO: 2007C1P2 10-JUL-2007.

243. En cada caso exhibir un grafo conexo con ejes que cumpla lo indicado, o demostrar que tal grafo no existe. Justificar. (Considerar que los caminos eulerianos y los caminos hamiltonianos son abiertos.)
   (a) Con camino euleriano y sin circuito euleriano.
   (b) Con circuito euleriano y sin camino euleriano.
   (c) Con camino hamiltoniano y sin circuito hamiltoniano.
   (d) Con circuito hamiltoniano y sin camino hamiltoniano.
   (e) Con camino euleriano y sin camino hamiltoniano.
   (f) Con camino hamiltoniano y sin camino euleriano.
   (g) Con circuito euleriano y sin circuito hamiltoniano.
   (h) Con circuito hamiltoniano y sin circuito euleriano.

   SOLUCIÓN:
   (a) K_2.
   (b) K_3.
   (c) K_2.
   (d) No existe.
   (e) Un grafo bipartito con muchos nodos de grado par de un lado, y pocos nodos del otro, dos de grado impar y el resto de grado par.
   (f) K_4.
   (g) $K_{2,4}$.
   (h) K_4.

   TOMADO: 2007C1R2 08-AGO-2007.

244. (a) Para cada $n \in \{2, 3, 4\}$ exhibir todos los grafos no isomorfos de n vértices que tienen a la vez camino euleriano y camino hamiltoniano. Justificar.
   (b) Sea $f(n)$ la cantidad de grafos no isomorfos de n vértices que tienen a la vez camino euleriano y camino hamiltoniano. Decidir para cada entero $n \geq 2$ si $f(n)$ vale 0, 1, 2 o al menos 3. Justificar.
   (c) Para cada $n \in \{2, 3, 4, 5\}$ exhibir todos los grafos no isomorfos de n vértices que tienen a la vez circuito euleriano y circuito hamiltoniano. Justificar.
   (d) Sea $f(n)$ la cantidad de grafos no isomorfos de n vértices que tienen a la vez circuito euleriano y circuito hamiltoniano. Decidir para cada entero $n \geq 2$ si $f(n)$ vale 0, 1, 2 o al menos 3. Justificar.

   (Considerar que los caminos eulerianos y los caminos hamiltonianos son abiertos.)

   SOLUCIÓN:
   (a) Cada grafo debe tener a P_n como subgrafo a fin de que pueda tener camino hamiltoniano. Luego se pueden agregar o no ejes libremente para que tenga también camino euleriano, sin que deje de tener camino hamiltoniano.
   Para $n = 2$ no es posible agregar ejes a P_2 sin que deje de ser simple, de modo que P_2 es el único.
   Para $n = 3$ se obtiene $K_3 = C_3$ si se agrega un eje a P_3, y no es posible seguir agregando ejes. Como C_3 no tiene nodos de grado impar, no tiene camino euleriano, de modo que el único grafo es P_3.
   Para $n = 4$ hay dos formas distintas de agregar un eje a P_4: una es conectar las puntas para formar C_4, y la otra es conectar una punta con el nodo adyacente a la otra punta de modo de formar C_3 con una pata. El grafo C_4 no tiene nodos de grado impar de modo que no tiene camino euleriano. El otro grafo es conexo y tiene exactamente 2 nodos de grado impar, de modo que tiene camino euleriano. Si agregamos otro eje a cualquiera de esos dos grafos, obtenemos $K_4 - e$ (diamante), que es conexo y tiene exactamente 2 nodos de grado impar, de modo que también tiene camino euleriano. Si agregamos otro eje obtenemos K_4 que no tiene camino euleriano porque no tiene los grados apropiados. No es posible seguir agregando ejes. En resumen, hay tres grafos que cumplen lo pedido: P_4, C_3 con una pata, y $K_4 - e$.

(b) Por el punto anterior $f(2) = 1$, $f(3) = 1$ y $f(4) = 3$. Para $n \geq 5$ el grafo P_n cumple lo pedido. Se le puede agregar un eje que conecte una punta con un nodo que no sea la otra punta. Esto da como resultado un grafo conexo y con 2 nodos de grado impar, de modo que tiene camino euleriano. Si ahora agregamos un eje que conecte la otra punta con un nodo que no sea la primera punta del P_5 original, volvemos a obtener un grafo conexo con 2 nodos de grado impar, que también tiene camino euleriano. En consecuencia $f(n) \geq 3$ para también para $n \geq 5$.

(c) Para $n = 2$ no hay ningún grafo porque debería ser conexo (por hamiltoniano) y tener todos los nodos de grado par.

Para el resto de los valores, cada grafo debe tener a C_n como subgrafo a fin de que pueda ser hamiltoniano. Luego se pueden agregar o no ejes libremente para que sea también euleriano, sin que deje de ser hamiltoniano.

Para $n = 3$ no es posible agregar ejes a C_3 de modo que es el único.

Para $n = 4$ se obtienen nodos de grado impar si se agrega un eje a C_4, por lo que el grafo resultante no es euleriano. Si se agregan dos ejes a C_4 se obtiene K_4 que tiene nodos de grado impar, y tampoco sirve. En consecuencia el único grafo es C_4.

Para $n = 5$ se obtienen 2 nodos de grado impar si se agrega un eje a C_5. Hay una única forma de corregir cualquiera de ellos para que sea par, agregando un eje incidente al nodo que se corrige, el cual también estropea el nodo del otro extremo del eje agregado, que debe ser corregido luego. Eso ocurre hasta que el eje agregado corrige dos nodos a la vez, que es cuando se forma K_5, que es euleriano. En resumen, los únicos grafos que cumplen lo pedido son C_5 y K_5.

(d) Por el punto anterior $f(2) = 0$, $f(3) = 1$, $f(4) = 1$ y $f(5) = 2$. Para $n \geq 6$ el grafo C_n cumple lo pedido. Para encontrar otros grafos es necesario agregar ejes manteniendo los grados pares. Una posibilidad es tomar cualquier conjunto independiente de 3 nodos y formar con ellos un triángulo. Luego se puede repetir con otro conjunto independiente de 3 nodos distintos de los primeros. Cada uno de los grafos cumple lo pedido, de modo que $f(n) \geq 3$ para $n \geq 6$.

NOTA: Basado en una idea de Elisa Orduna.

TOMADO: 2014C2P2 01-DIC-2014.

245. A, B, C, D, E y F son agentes secretos. Algunos de ellos deben reunirse de a pares en un reducto subterráneo para intercambiar información. Por razones de seguridad sólo un par de agentes puede reunirse al mismo tiempo. Asimismo se ha decidido que por cada reunión (excepto la última), uno de los dos agentes que participen en la misma también debe participar en la siguiente. A continuación aparecen en un orden arbitrario los pares de agentes que deben reunirse: A con B, A con F, B con C, B con D, B con E, C con E, C con F, D con F, E con F. Modelar la situación como un problema sobre grafos, y encontrar una sucesión de reuniones que cumpla con los requerimientos. Justificar.

SOLUCIÓN: Armar un grafo con un vértice por cada agente, uniendo los pares de agentes que deben reunirse; luego buscar un camino euleriano. Alternativamente, armar un grafo con un vértice por cada reunión, conectando las reuniones que tienen un agente en común; luego buscar un camino hamiltoniano en el grafo. Aparentemente el segundo grafo es el grafo de líneas del primero.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2005C1R2 08-AGO-2005.

246. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

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.

Sea G un grafo con ejes.

(a) Determinar el grado de cada vértice de $L(G)$ en función de los grados de los vértices de G.

(b) Demostrar que si G es regular entonces $L(G)$ es regular. ¿Vale la recíproca?

(c) Demostrar que si G es conexo entonces $L(G)$ es conexo. ¿Vale la recíproca?

(d) Demostrar que si G es regular y conexo entonces $L(G)$ tiene un circuito euleriano. ¿Vale la recíproca?

(e) Demostrar que si G tiene un circuito euleriano entonces $L(G)$ tiene un circuito euleriano. ¿Vale la recíproca?

(f) Demostrar que si G tiene un circuito euleriano entonces $L(G)$ tiene un circuito hamiltoniano. ¿Vale la recíproca?

En todos los casos justificar las respuestas.

SOLUCIÓN:

(a) Ver Ejercicio 123a.

(b) Surge del primer punto. Si el grado de todos los nodos de G es d, el grado de cualquier nodo de $L(G)$ es $d + d - 2$. La recíproca es falsa, por ejemplo si G tuviera algunos nodos de grado d y los adyacentes a esos de grado $d + 1$, como en $K_{1,2}$, que no es regular, mientras que $L(K_{1,2}) = K_2$ sí lo es.

(c) Para obtener un camino entre dos nodos en $L(G)$ basta seguir un camino entre los extremos de los ejes correspondientes de G. La recíproca es falsa, por ejemplo si G tiene nodos aislados.

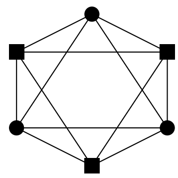
(d) Por los puntos anteriores, si G es regular de grado d entonces $L(G)$ es regular de grado $2d-2$, que es par. Además, si G es conexo entonces $L(G)$ es conexo. En definitiva $L(G)$ tiene todos sus nodos de grado par y es conexo, de modo que tiene un circuito euleriano.

(e) Si G tiene un circuito euleriano entonces tiene todos sus nodos de grado par y una única componente conexa con ejes. Los nodos aislados que pudiera tener G no modifican $L(G)$. Por los puntos anteriores, $L(G)$ también tiene todos sus nodos de grado par y es conexo, de modo que tiene un circuito euleriano. La recíproca es falsa, por ejemplo si G no es euleriano pero es regular y conexo, como en K_2 que no tiene circuito euleriano, mientras que $L(K_2) = K_1$ sí lo tiene.

(f) Si G tiene un circuito euleriano, los nodos de $L(G)$ correspondientes a los ejes de G, en el mismo orden, forman un circuito hamiltoniano. La recíproca es falsa, porque en $L(G)$ se forman circuitos que no provienen de circuitos en G, como ocurre con $K_{1,n}$, que no tiene circuito euleriano, mientras que $L(K_{1,n}) = K_n$ tiene circuito hamiltoniano.

TOMADO: 2005C2P2 06-DIC-2005 (último punto), 2005C2R1 15-DIC-2005 (tres primeros puntos), 2005C2R2 22-DIC-2005 (puntos a, d y e).

247. 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. En la siguiente figura aparece $T(K_3)$; los vértices correspondientes a vértices de K_3 son círculos, mientras que los vértices correspondientes a ejes de K_3 son cuadrados.



Sea G un grafo conexo.

(a) Demostrar que si G tiene un circuito euleriano entonces $T(G)$ también. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SUGERENCIA: Usar caracterización de grafos eulerianos.

(b) Demostrar que si G tiene un circuito hamiltoniano entonces $T(G)$ también. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

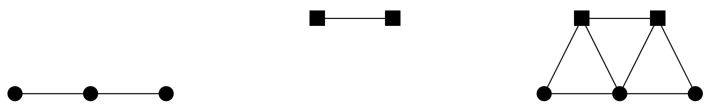
(a) En el Ejercicio 122a se demuestra que si G es conexo entonces $T(G)$ también. En el Ejercicio 123c se calculan los grados en $T(G)$ en base a los de G, que resultan pares si los de G lo eran, lo cual ocurre por ser euleriano. Por lo tanto, $T(G)$ es conexo y con grados pares, lo que implica que es euleriano. La recíproca es falsa, porque de acuerdo al Ejercicio 240 los grados en $T(G)$ son pares si y sólo si los grados en G tienen la misma paridad (podrían ser todos impares).

(b) Si $G = K_2$ (suponiendo que K_2 es hamiltoniano) entonces $T(G) = K_3$ que es hamiltoniano. Lo que sigue es para G hamiltoniano que no es K_2, por lo que su circuito hamiltoniano es simple. G es subgrafo de $T(G)$, de modo que existe un circuito en $T(G)$ que pasa por todos los vértices que eran vértices de G. Para pasar también por los vértices de $T(G)$ que eran ejes en G, se puede remplazar cada eje de ese circuito por dos ejes, uno de ida y otro de vuelta al vértice de $T(G)$ correspondiente al eje de G (digamos picos o puentes que suben y bajan). Falta pasar por los vértices de $T(G)$ que eran ejes de G que no forman parte del circuito hamiltoniano de G. Sea vx uno de esos ejes de G incidente sobre el nodo v, y sean u y v los vértices anterior y siguiente a v en el circuito hamiltoniano de G. Hasta ahora, tenemos en $T(G)$ un circuito de la forma $\ldots, u, uv, v, vw, w, \ldots$. Como en $T(G)$ el vértice vx es adyacente a uv y a v, podemos insertarlo entre esos nodos obteniendo $\ldots, u, uv, vx, v, vw, w, \ldots$. Notar que este cambio no nos impide insertar posteriormente cualquier otro eje pendiente de G que también incida sobre v, digamos el eje vy, ya que el nodo correspondiente en $T(G)$ es adyacente a vx y a v. De esta manera podemos ir agregando todos los vértices faltantes, hasta obtener un circuito hamiltoniano en $T(G)$. La recíproca es falsa; por ejemplo, $T(K_{1,2})$ es hamiltoniano.

NOTA: Propuesto originalmente por Leandro Montero.

TOMADO: 2009C1P2 10-AGO-2009.

248. 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 conexo con ejes.

(a) Demostrar que G y $L(G)$ son subgrafos de $T(G)$.

(b) Demostrar que $T(G)$ tiene un subgrafo generador conexo euleriano.
SUGERENCIA: El subgrafo es $T(G) - E(L(G))$.

(c) Demostrar que si G tiene un subgrafo generador conexo euleriano entonces $T(G)$ es hamiltoniano.
SUGERENCIA: Transformar el circuito euleriano del subgrafo de G, en un circuito hamiltoniano de $T(G)$, corrigiéndolo mediante "desviaciones" de ida y vuelta que utilizan porciones de $L(G)$.

(d) Demostrar que si G es euleriano entonces $T(G)$ es hamiltoniano.

(e) Demostrar que $T(T(G))$ es hamiltoniano.

SOLUCIÓN:

(a) Surge de la definición.

(b) El grafo propuesto es subgrafo de $T(G)$. Es generador porque no se le quitan vértices. Es conexo porque G lo es y cada vértice de $T(G)$ correspondiente a un eje de G es adyacente en $T(G)$ a quienes eran sus dos extremos en G. Al ser conexo, es euleriano porque todos sus vértices tienen grado par: los vértices correspondientes a ejes de G tienen grado 2 en el subgrafo, mientras que los vértices correspondientes a vértices de G tienen el doble del grado que tienen en G.

(c) Si recorremos el subgrafo generador conexo euleriano de G en $T(G)$, pasamos por todos los vértices de G, eventualmente más de una vez. Para que eso se vuelva un circuito hamiltoniano en $T(G)$, hay que evitar repetir vértices, e incluir los vértices de $T(G)$ que son ejes en G. Para evitar repetir vértices, cada vez que vamos a repetir un vértice podemos subir a $L(G)$ y continuar por los vértices de $L(G)$ correspondientes a los ejes que recorreríamos en G, bajando en cuanto aparece un vértice no repetido. Al final nos queda un circuito en $T(G)$ que no repite vértices, y que eventualmente utiliza algunos vértices de $L(G)$. Para los vértices de $L(G)$ que pudieran faltar, remplazamos un eje en G por un eje de ida y otro de vuelta que pasan por el vértice correspondiente de $L(G)$. El eje remplazado tiene que estar en el circuito, porque si no está significa que el vértice de $L(G)$ ya fue agregado en el paso anterior.

(d) Al ser G euleriano (y conexo), es un subgrafo generador conexo euleriano de G mismo. Por el punto anterior $T(G)$ es hamiltoniano.

(e) Sea $H = T(G)$. Por el segundo punto H tiene un subgrafo generador conexo euleriano, de modo que por el tercer punto $T(H)$ es hamiltoniano.

NOTA: Propuesto originalmente por Leandro Montero.

TOMADO: 2013C2R2 21-DIC-2013 (excepto el tercer punto).

Hamiltoniano

249. Determinar para qué valores de n, p, q y h los siguientes grafos tienen camino (abierto) hamiltoniano. Justificar.

(a) K_n

(b) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(c) $K_{p,q}$

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

SOLUCIÓN:

K_n: K_1 y K_2 tienen camino hamiltoniano. Respecto de K_n con $n \geq 3$, también tiene camino hamiltoniano porque tiene circuito hamiltoniano.

W_n: Cualquier n sirve porque tiene circuito hamiltoniano.

$K_{p,q}$: Los caminos alternan nodos de cada una de las partes. Los valores $|p-q| = 0$ (es decir, $p = q$) sirven porque siempre se puede ir a un nodo no visitado de la otra parte, hasta visitarlos todos. Los valores $|p - q| = 1$ también sirven porque puede lograrse lo dicho empezando el camino en la parte con más nodos. Los valores $|p - q| \geq 2$ no sirven porque se acaban los nodos de la parte más chica antes de haber visitado todos los de la parte más grande.

ABC h: Los valores $h \leq 1$ sirven. Los valores $h \geq 2$ no sirven porque el camino debería empezar o terminar en al menos 4 hojas distintas.

NOTA: Encontrado originalmente en Internet (circuito euleriano y hamiltoniano). No se incluye C_n (trivial), P_n (trivial), A n (variable), ni GBC n m (variable).

TOMADO: 2018C1P2 06-JUL-2018.

250. Determinar para qué valores de n, p, q y h los siguientes grafos tienen circuito hamiltoniano. Justificar.

(a) K_n

(b) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(c) $K_{p,q}$

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

SOLUCIÓN:

K_n: Los valores $n \leq 2$ son discutibles. Para $n \geq 3$ el grafo K_n es hamiltoniano, porque siempre se puede armar un circuito yendo a un nodo no visitado y finalmente volviendo al origen.

W_n: Cualquier n sirve, ya que se puede ir del nodo universal a uno del circuito, recorrer todo el circuito, y volver al nodo universal.

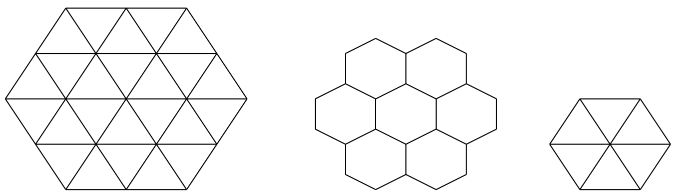
$K_{p,q}$: Debe ser $p = q$ ya que caso contrario habría una parte con menos nodos que la otra, y al elimar esa parte, quedarían demasiadas componentes conexas. En tal caso es sencillo armar el circuito empezando en cualquier nodo y yendo siempre a un nodo no visitado, para finalmente cerrar el circuito. El caso $p = q = 1$ es discutible.

ABC h: El valor $h = 0$ es discutible. Los valores $h \geq 1$ no sirven porque eliminando los padres de las hojas quedarían demasiadas componentes conexas.

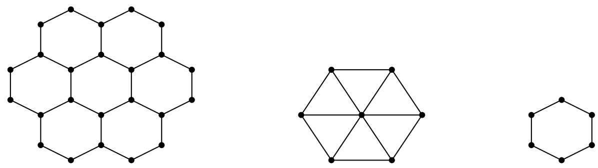
NOTA: Encontrado originalmente en Internet (circuito euleriano y hamiltoniano). No se incluye C_n (trivial), P_n (trivial), A n (variable), ni GBC n m (variable).

TOMADO: 2005C1R2 08-AGO-2005 (K_n y $K_{p,q}$).

251. (a) Decidir para cada dibujo de la figura si es posible recorrerlo empezando y terminando en un mismo polígono y pasando exactamente una vez por cada uno de los polígonos que lo forman. El recorrido puede pasar de un polígono a otro si y sólo si ambos polígonos comparten al menos un lado. Justificar.



(b) Decidir para cada grafo de la figura si tiene un circuito hamiltoniano. Justificar.

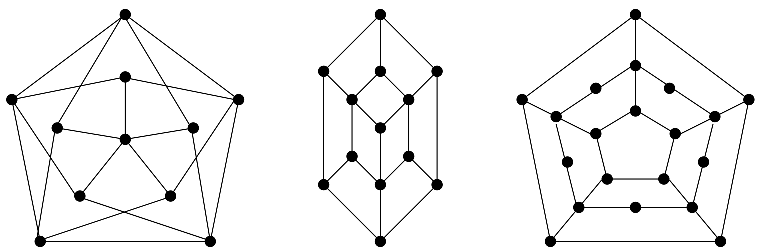


SOLUCIÓN: Los grafos modelan las adyacencias de los dibujos, de manera tal que ambos puntos son equivalentes. En el primer grafo los vértices de la periferia pueden formar parte de una única manera de un posible circuito hamiltoniano, que excluye a los vértices internos, de modo que no es hamiltoniano; otra forma de argumentarlo es notar que si se eliminan los 6 vértices de grado 3 de la periferia, se obtiene un grafo con 7 componentes conexas. Es fácil encontrar circuitos hamiltonianos en los otros dos grafos.

NOTA: Las figuras son las mismas que en el Ejercicio 319.

TOMADO: 2013C2P2 02-DIC-2013.

252. Demostrar que el primer grafo tiene un circuito hamiltoniano pero los otros dos grafos no.



SUGERENCIA: Para el primer grafo, analizar cómo podría formar parte del circuito el vértice central.

SOLUCIÓN: Para el primer grafo, el vértice central puede formar parte del circuito usando dos ejes contiguos de la estrella, o dos ejes no contiguos; en cualquier caso hay 3 ejes de la estrella que no son usados, lo que hace que 3 de los vértices interiores puedan formar parte del circuito de una única manera, lo cual a su vez permite deducir que otros ejes pertenecen o no al circuito; si se eligen ejes contiguos para el nodo central finalmente se obtiene un circuito hamiltoniano, y si se eligen ejes no contiguos, no se puede formar un circuito. El segundo grafo es bipartito (por ejemplo porque su número cromático es 2, debido a que tiene ejes, es planar y toda región tiene una cantidad par de aristras frontera); como la cantidad de nodos es 13 (impar), no puede ser hamiltoniano. En el tercer grafo hay una única manera de incluir los nodos de grado 2, lo cual forma un circuito cerrado que luego no permite agregar a los otros nodos; otra forma de argumentarlo es notar que si se eliminan los 5 nodos de grado 4 se obtiene un grafo con más de 5 componentes conexas.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2017C1R2 21-JUL-2017.

253. Un torneo (también llamado digrafo completamente conexo) es una orientación de un grafo completo. Más precisamente, un torneo es un digrafo que se obtiene asignando una dirección a cada eje de un grafo completo no dirigido.

Sea G un torneo de $n \geq 3$ vértices. Demostrar que G es fuertemente conexo o puede volverse fuertemente conexo invirtiendo exactamente un arco.

SOLUCIÓN: Por el Ejercicio 8.14 de la Práctica, G tiene un camino hamiltoniano dirigido. Si $n \geq 3$ hay un eje adicional entre sus extremos, que si está orientado igual que el camino hace que el digrafo tenga un circuito hamiltoniano dirigido, y caso contrario hace que lo tenga al invertirlo. Una vez conseguido el circuito, el digrafo es fuertemente conexo. También sale por inducción, pero es más complicado.

NOTA: Propuesto originalmente por Federico Lebrón.

TOMADO: 2012C1R2 08-AGO-2012.

254. Dado un digrafo, un orden $v_1, v_2, \ldots$ de sus vértices se dice topológico si y sólo si para todo eje (v_i, v_j) del digrafo se cumple que $i < j$ (es decir, si hay un eje dirigido de v_i a v_j, entonces v_i aparece antes que v_j en el orden).

Sea G un digrafo sin ciclos dirigidos. Demostrar que G tiene un camino hamiltoniano dirigido si y sólo si existe un único orden topológico de sus vértices.

SUGERENCIA: No omitir existencia ni unicidad.

SOLUCIÓN: Para la ida, tomar un camino hamiltoniano; ese orden es topológico, ya que todos los arcos de un vértice al siguiente deben existir, y si hubiera ejes hacia atrás entonces habría ciclos; por otro lado, es único, ya que si hubiera otro, podemos tomar el primer vértice en el cual difieren, digamos v_i en el camino hamiltoniano versus v_j en el otro orden, con $j > i$, y el eje (v_{j-1}, v_j) iría hacia atrás en ese otro orden supuestamente topológico. Para la vuelta, tomar el único orden topológico que existe; debe existir un arco de cada nodo al siguiente, ya que caso contrario se podrían invertir y obtener otro orden topológico.

NOTA: Propuesto originalmente por Xavier Warnes. La propiedad permite obtener un algoritmo lineal para resolver el problema del camino hamiltoniano en grafos dirigidos sin ciclos, ya que encontrar un orden topológico es lineal.

TOMADO: 2012C2P2 05-DIC-2012.

255. Se tiene un tablero de $n \times n$ casillas, con $n \geq 2$. Se desea que una torre de ajedrez recorra el tablero mediante movimientos válidos, pasando una sola vez sobre cada casilla y volviendo a la casilla original. Determinar los valores de n para los cuales esto es posible. Justificar.

SOLUCIÓN: Para n par el recorrido es posible. Para $n = 2$ el recorrido es trivial. Para $n \geq 4$, podemos considerar un recorrido como el pedido para las $n-2$ primeras filas y columnas, eliminamos el movimiento que va de la primera casilla de la fila $n-2$ a la segunda casilla de esa fila, y agregamos ahí un camino que recorra las dos últimas filas y columnas.

Para n impar el recorrido no es posible. Podemos armar un grafo con un nodo por cada casilla, y un eje entre los nodos correspondientes a casillas que comparten un lado. Estos ejes representan los movimientos posibles de la torre. Encontrar un recorrido como el pedido equivale a encontrar un circuito hamiltoniano en el grafo. Este grafo resulta bipartido (las casillas blancas forman un conjunto, y las negras otro). El grafo tiene n^2 nodos, que resulta impar por ser n impar. De acuerdo al Ejercicio 8.15 de la Práctica, los grafos bipartitos con un número impar de nodos no admiten un circuito hamiltoniano.

TOMADO: 2002C1P2 10-JUL-2002.

256. El grafo K_3 tiene un único circuito hamiltoniano. Determinar cuántos tiene K_n, para $n > 3$. Justificar.

   SOLUCIÓN: Inductivamente, si elegimos cualquier vértice de K_n, tenemos $n-1$ posibles lugares para agregarlo en cada uno de los circuitos hamiltonianos de K_{n-1}. Otra forma es considerar los $n!$ posibles órdenes diferentes de los vértices de K_n, y notar que cada circuito hamiltoniano está asociado a $2n$ órdenes diferentes.

   TOMADO: 2002C1R2 07-AGO-2002.

257. Sea G un grafo no conexo de $n \geq 3$ vértices.

   (a) Demostrar que si G tiene todos sus vértices de grado a lo sumo $n/2-1$, entonces G^c tiene un circuito hamiltoniano.

   (b) Demostrar que si G tiene todos sus vértices de grado al menos $n/2-1$, entonces G^c tiene un circuito hamiltoniano.

   SUGERENCIA: Evitar ser engañados.

   SOLUCIÓN:

   (a) El resultado es cierto aunque el grafo sea conexo. Si todos los vértices tienen grado a lo sumo $n/2-1$, en G^c tienen grado al menos $n-1-(n/2-1) = n/2$, por lo que de acuerdo al teorema de Dirac, G^c es hamiltoniano.

   (b) Si todos los vértices tienen grado al menos $n/2-1$, cada componente conexa tiene al menos $n/2$ nodos de grado al menos $n/2-1$, por lo que G es la unión de dos copias de $K_{n/2}$, de modo que G^c es $K_{n/2,n/2}$ que es hamiltoniano. También puede demostrarse usando el teorema de Dirac (debido a que lo que está afuera de cada componente conexa tienen al menos $n/2$ nodos), pero no vale la pena.

   TOMADO: 2005C2R2 22-DIC-2005.

258. (a) Para cada $n \geq 3$ impar exhibir un grafo de n vértices, todos ellos de grado al menos $(n-1)/2$, y sin circuitos hamiltonianos. Justificar.

   (b) Para cada $n \geq 3$ par repetir lo anterior o demostrar que no es posible hacerlo.

   SOLUCIÓN:

   (a) $K_{k,k+1}$, con $k \in \mathbb{N}$.

   (b) No es posible porque el grado, al ser entero, sería al menos $n/2$, y estaríamos en las condiciones del Teorema de Dirac.

   NOTA: Encontrado originalmente en Internet. Por el Ejercicio 84 sabemos que los grafos buscados deben ser conexos, razón por la cual el enunciado no lo pide.

   TOMADO: 2008C1R2 25-JUL-2008.

259. Sea G un grafo sin circuitos.

   (a) Demostrar que G tiene un camino hamiltoniano si y sólo si G es un camino.

   (b) Diseñar un algoritmo eficiente que decida si G tiene un camino hamiltoniano, y en tal caso encontrarlo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n)$, donde n es la cantidad de vértices de G.

   SOLUCIÓN: Se puede contar cuántos nodos de grado 1 y de grado 2 hay. Si hay respectivamente 2 y $n-2$ es un camino, que luego se puede buscar por BFS empezando por uno de los nodos de grado 1. También se puede hacer BFS directamente, verificando en cada paso.

   NOTA: Puede pedirse uno solo de los puntos.

   TOMADO: 2011C2R2 21-DIC-2011.

260. Supongamos que se dispone de un algoritmo que dado un grafo, informa en tiempo polinomial si el mismo tiene un circuito hamiltoniano o no. En base a ese algoritmo, diseñar un nuevo algoritmo que dado un grafo G, encuentre en tiempo polinomial un circuito hamiltoniano del mismo, o informe que tal circuito no existe. Mostrar que el algoritmo propuesto es correcto y determinar la cantidad de veces que invoca al algoritmo original. Justificar. La cantidad de veces que el mejor algoritmo que conocemos invoca al algoritmo original es $O(m)$, donde m es la cantidad de ejes de G.

    SOLUCIÓN: Sea A el algoritmo original. Si A responde no, responder no. Caso contrario, para cada eje e de G, si A responde sí en $G - e$, sacar e de G. Al final se obtiene un grafo hamiltoniano (porque A respondió sí para la última versión del grafo). El grafo debe ser exactamente un circuito, ya que si tuviera más ejes A debería haber respondido sí al intentar sacarlos. Todos los ejes de ese circuito estaban en el grafo original, de modo que forman un circuito hamiltoniano en el mismo.

    NOTA: Propuesto originalmente por Federico Lebrón. Es parecido al Ejercicio 367.

    TOMADO: 2011C2P2 05-DIC-2011 (sin invocaciones).

261. Sea $G = K_n$ ($n \geq 2$) con pesos arbitrarios asociados a sus ejes. Sea $e_{\min}(G)$ cualquier eje de G que tenga peso mínimo y $e_{\max}(G)$ cualquier eje de G que tenga peso máximo. ¿Es cierto que…

    (a) $e_{\min}(G)$ pertenece a algún camino hamiltoniano de peso mínimo de G?
    (b) $e_{\min}(G)$ pertenece a todo camino hamiltoniano de peso mínimo de G?
    (c) $e_{\max}(G)$ no pertenece a algún camino hamiltoniano de peso mínimo de G?
    (d) $e_{\max}(G)$ no pertenece a ningún camino hamiltoniano de peso mínimo de G?

    Demostrar o dar un (contra) ejemplo y justificar según corresponda.

    Repetir suponiendo que los pesos cumplen la desigualdad triangular.

    SOLUCIÓN:

    (a) Falso; para encontrar un contraejemplo, pensemos que si tenemos un camino hamiltoniano que contiene a $e_{\min}(G)$, ese camino puede contener a lo sumo a dos ejes adyacentes a $e_{\min}(G)$; por el contrario, un camino hamiltoniano que no contiene a $e_{\min}(G)$ puede contener hasta cuatro de esos ejes; si todos los ejes adyacentes a $e_{\min}(G)$ tienen poco peso, y los demás tienen peso mayor, el circuito hamiltoniano de peso mínimo no contendrá a $e_{\min}(G)$; con esta idea, es posible armar un contraejemplo, consistente en K_5, donde los ejes de un triángulo tienen peso 2, y los otros ejes tienen peso 1; si elegimos $e_{\min}(G)$ como el eje que no tiene extremos en el triángulo, puede verse que cualquier camino hamiltoniano que contenga a ese eje, necesariamente contendrá a un eje de peso 2, resultando un camino de peso al menos 5, mientras que el camino hamiltoniano de peso mínimo tiene peso 4.
    (b) Falso; por ejemplo para K_3 con todos ejes de igual peso.
    (c) Falso; por ejemplo para K_2. Si hay más vértices es verdadero: sea cualquier camino hamiltoniano de peso mínimo; si no contiene a $e_{\max}(G)$, listo; si lo contiene, sacamos $e_{\max}(G)$ y agregamos el eje que une los extremos del camino original, el cual no tiene peso mayor que $e_{\max}(G)$.
    (d) Ídem segundo punto.

    TOMADO: 2004C1P2 14-JUL-2004.

262. Sea $G = K_n$ ($n \geq 3$) con pesos arbitrarios asociados a sus ejes. Sea $e_{\min}(G)$ cualquier eje de G que tenga peso mínimo y $e_{\max}(G)$ cualquier eje de G que tenga peso máximo. ¿Es cierto que…

    (a) $e_{\min}(G)$ pertenece a algún circuito hamiltoniano de peso mínimo de G?
    (b) $e_{\min}(G)$ pertenece a todo circuito hamiltoniano de peso mínimo de G?
    (c) $e_{\max}(G)$ no pertenece a algún circuito hamiltoniano de peso mínimo de G?
    (d) $e_{\max}(G)$ no pertenece a ningún circuito hamiltoniano de peso mínimo de G?

    Demostrar o dar un (contra) ejemplo y justificar según corresponda.

    Repetir suponiendo que los pesos cumplen la desigualdad triangular.

    SOLUCIÓN:

(a) Falso; por ejemplo para K_4, con un eje de peso 2 y todos los demás de peso 1; si elegimos $e_{\min}(G)$ como el eje que no es adyacente al de peso 2 puede verse que cualquier circuito hamiltoniano que contenga a ese eje, necesariamente contendrá al eje de peso 2, resultando un circuito de peso al menos 5, mientras que el circuito hamiltoniano de peso mínimo tiene peso 4.

(b) Ídem punto anterior.

(c) Falso; por ejemplo para K_3.

(d) Ídem punto anterior.

TOMADO: 2004C1R2 09-AGO-2004.

263. Sea $G = K_n$ ($n \geq 3$) con pesos arbitrarios asociados a sus ejes. ¿Es cierto que...

(a) si un circuito hamiltoniano de peso mínimo de G contiene un camino hamiltoniano de peso mínimo, el camino se obtiene eliminando del circuito un eje de mayor peso?

(b) algún circuito hamiltoniano de peso mínimo de G contiene un camino hamiltoniano de peso mínimo?

(c) todo circuito hamiltoniano de peso mínimo de G contiene un camino hamiltoniano de peso mínimo?

Demostrar o dar un (contra) ejemplo y justificar según corresponda.

Repetir suponiendo que los pesos cumplen la desigualdad triangular.

SOLUCIÓN:

(a) Verdadero.

(b) Falso; para encontrar un contraejemplo, la idea es armar un grafo con un circuito que tenga un eje de mucho peso y los otros de poco peso, de tal modo que estos últimos ejes sean el único camino hamiltoniano de peso mínimo; por otro lado se agregan ejes intermedios de tal modo que el circuito hamiltoniano de peso mínimo sea otro; por ejemplo, podemos tomar K_4 con dos ejes no adyacentes de peso 1.4, un eje de peso 2, y los demás de peso 1.

(c) Ídem punto anterior.

TOMADO: 2003C1R2 04-AGO-2003 (sin la parte de desigualdad triangular).

264. Sea $G = (V, E)$ un grafo completo con costos no negativos asociados a sus ejes, los cuales cumplen la desigualdad triangular. Dado $W \subseteq V$ definimos $T(W)$ como el costo de una solución óptima al Problema del Viajante de Comercio (TSP) en el subgrafo inducido por W.

(a) Demostrar que $T(W_1) \leq T(W_2)$ para W_1 y W_2 cualesquiera tales que $W_1 \subseteq W_2 \subseteq V$. SUGERENCIA: Modificar una solución óptima para W_2.

(b) ¿Sigue valiendo la propiedad del punto anterior si los costos no cumplen la desigualdad triangular? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN: Considerar un circuito óptimo para W_2. Si eliminamos los nodos que no están en W_1, y cada vez que lo hacemos conectamos el nodo anterior con el nodo siguiente, obtenemos en cada paso un circuito de costo menor o igual que el anterior. Finalmente, obtenemos una solución factible para el problema sobre W_1, y el óptimo sobre W_1 es menor o igual que eso. La propiedad no vale en general sin desigualdad triangular. Por ejemplo, si $G = K_4$ con un eje de costo M y todos los demás de costo 1, si W_1 es V sin el nodo que tiene todos sus ejes de costo 1, mientras que $W_2 = V$, resulta $T(W_1) = M + 2$ y $T(W_2) = 4$.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2013C1R2 05-AGO-2013.

265. Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia. El costo de una correspondencia es la suma de los costos de los ejes de la correspondencia. Una correspondencia perfecta de costo mínimo de un grafo es una correspondencia perfecta del grafo que tiene costo menor o igual que el de cualquier correspondencia perfecta del grafo.

Sea G un grafo completo con un número par de vértices, y costos asociados a sus ejes. Sea M^* cualquier correspondencia perfecta de costo mínimo de G, y sea C^* cualquier solución óptima al Problema del Viajante de Comercio (TSP) en G. Demostrar que $c(C^*) \geq 2c(M^*)$, donde $c(X)$ es el costo de X.

SUGERENCIA: Obtener dos correspondencias perfectas a partir de C^*.

SOLUCIÓN: C^* tiene un número par de ejes y nodos. Si tomamos ejes intercalados, podemos formar dos conjuntos de ejes M_1 y M_2 que son correspondencias perfectas, cada una de costo mayor o igual que el de la óptima M^*, y cuya suma de costos es la de C^*. En definitiva, $c(C^*) = c(M_1) + c(M_2) \geq 2c(M^*)$.

NOTA: Encontrado originalmente en Internet. Con la sugerencia es muy fácil.

TOMADO: 2010C2P2 06-DIC-2010.

Planaridad

266. Sea G_n el grafo de vértices $v_1, v_2, \ldots v_n$ tal que v_i y v_j son adyacentes si y sólo si $|i-j| \leq 3$. Demostrar que G_n es planar.

SUGERENCIA: Inducción en n.

SOLUCIÓN: Probar algo más fuerte para $n \geq 3$: que existe una representación planar de G_n en la cual los 3 últimos vértices forman una región triángular.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2018C2R2 19-DIC-2018.

267. Dado un grafo G definimos G^3 como el grafo que se obtiene al tomar tres copias de G y agregar los tres ejes posibles entre las tres copias de cada vértice de G. Por ejemplo, $K_1^3 = K_3$ y K_2^3 es un prisma de base triangular.

Sea G un grafo de n vértices y m ejes.

(a) Demostrar sin usar otros puntos de este ejercicio que si G^3 es planar entonces $m \leq 2n-2$.

(b) Demostrar que si G^3 es planar entonces $m \leq n-1$.

(c) Demostrar que si G tiene ciclos simples entonces G^3 no es planar.
SUGERENCIA: Demostrar que C_k^3 no es planar.

(d) Demostrar que si $\Delta(G) \geq 3$ entonces G^3 no es planar.
SUGERENCIA: Demostrar que $K_{1,3}^3$ no es planar.

(e) Demostrar que P_k^3 es planar.

(f) Demostrar que G^3 es planar si y sólo si cada componente conexa de G es un camino simple.

SOLUCIÓN:

(a) Sean $n' = 3n$ y $m' = 3m + 3n$ las cantidades de vértices y ejes de G^3, respectivamente. Como $n' \geq 3$ y G^3 es planar, se cumple $m' \leq 3n' - 6$. Si remplazamos n' y m' de acuerdo a las igualdades mencionadas al principio resulta $3m + 3n \leq 9n - 6$, que equivale a lo que pide el enunciado.

(b) Por el tercer punto G no tiene ciclos, es decir, es un bosque, los cuales tienen a lo sumo $n-1$ ejes.

(c) Si G tiene un ciclo simple entonces C_k^3 es subgrafo de G^3, de modo que para demostrar que G^3 no es planar es suficiente con argumentar que C_k^3 no es planar. Sean u_1, v_1 y w_1 tres vértices que son consecutivos en C_k. Además, u_1 y w_1 son adyacentes si $k = 3$, o hay más vértices entre ellos si $k > 3$. Esos vértices también están en C_k^3, siendo sus segundas copias u_2, v_2 y w_2, mientras que las terceras copias son u_3, v_3 y w_3. Se cumple que C_k^3 tiene un subgrafo homeomorfo a $K_{3,3}$, de modo que no es planar por Kuratowski. El subgrafo homeomorfo está formado por v_1, w_2 y w_3 en una parte, y u_1, w_1 y v_2 en la otra, junto con ejes y vértices adicionales que permiten conectar a cada vértice de una parte con cada vértice de la otra. Hay un eje directo entre v_1 y cada vértice de la otra parte. El vértice w_2 tiene un eje directo tanto a w_1 como a v_2, y se conecta a u_1 pasando por u_2. El vértice w_3 tiene un eje directo a w_1, se conecta a u_1 pasando por u_3, y se conecta a v_2 pasando por v_3.

(d) Si $\Delta(G) \geq 3$ entonces $K_{1,3}$ es subgrafo de G y por lo tanto $K_{1,3}^3$ es subgrafo de G^3, de modo que para demostrar que G^3 no es planar es suficiente con argumentar que $K_{1,3}^3$ no es planar. Sea v el vértice de grado 3 en $K_{1,3}$, y sean x, y y z los tres vértices de grado 1. Si contraemos en $K_{1,3}^3$ las tres copias de x, las tres copias de y, y las tres copias de z, esos tres vértices quedan conectados a las tres copias de v. Es así que $K_{1,3}^3$ tiene un subgrafo contraíble a $K_{3,3}$, de modo que no es planar.

(e) Basta exhibir una representación planar de P_k^3. Una posibilidad es trazar k triángulos (C_3) concéntricos, cada uno unido por sus tres vértices con los triángulos que tiene más próximos.

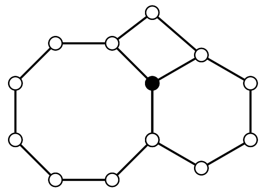
(f) Para la ida, por el tercer punto G no tiene ciclos, es decir, es un bosque. Además, por el cuarto punto $\Delta(G) \leq 2$, de modo que cada componente conexa de G es un árbol con vértices de grado a lo sumo 2, es decir, un camino simple.

La vuelta es cierta por el punto anterior.

NOTA: Propuesto originalmente por Brian Bokser y Agustín Santiago Gutiérrez.

TOMADO: 2017C2P2 01-DIC-2017 (cuatro últimos puntos), 2017C2R2 18-DIC-2017 (tres primeros puntos).

268. El grafo de un cuboctaedro truncado es un grafo conexo que admite una representación planar donde cada vértice está en el borde de exactamente tres regiones, cuyos bordes son los ciclos simples C_4, C_6 y C_8. En la siguiente figura se muestran las tres regiones en cuyos bordes está un vértice dado.



Sean n y m las cantidades de vértices y ejes del grafo, respectivamente. Sea r la cantidad total de regiones en la representación planar mencionada, de las cuales r_i tienen como borde al ciclo simple C_i para $i = 4, 6, 8$.

(a) Demostrar que $r_i = n/i$ para $i = 4, 6, 8$.

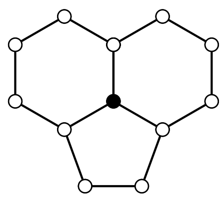
(b) Demostrar que $r = 13n/24$.

(c) Demostrar que $m = 3n/2$.

(d) Determinar los valores exactos de n, m, r y r_i para $i = 4, 6, 8$. Justificar.

SOLUCIÓN: Es el Ejercicio 271 instanciado con el cuboctaedro truncado. La solución es la de ese ejercicio instanciada con los datos del cuboctaedro truncado. Resulta $n = 48$, $m = 72$, $r = 26$, $r_4 = 12$, $r_6 = 8$ y $r_8 = 6$.

269. El grafo de una pelota de fútbol (llamado formalmente grafo de un icosaedro truncado) es un grafo conexo que admite una representación planar donde cada vértice está en el borde de exactamente tres regiones: una pentagonal y dos hexagonales (una región pentagonal es aquella cuyo borde es un ciclo simple de 5 vértices; análogamente para hexagonal). En la siguiente figura se muestran las regiones en cuyos bordes está un vértice dado.



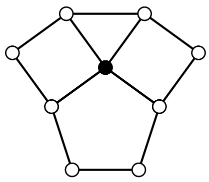
Sean n y m las cantidades de vértices y ejes del grafo, respectivamente. Sea r la cantidad total de regiones en la representación planar mencionada, de las cuales r_5 son pentagonales y r_6 son hexagonales.

(a) Demostrar que $r_5 = n/5$ y $r_6 = 2n/6$.
(b) Demostrar que $r = 8n/15$.
(c) Demostrar que $m = 3n/2$.
(d) Determinar los valores exactos de n, m, r, r_5 y r_6. Justificar.

SOLUCIÓN: Es el Ejercicio 271 instanciado con el icosaedro truncado. La solución es la de ese ejercicio instanciada con los datos del icosaedro truncado. Resulta $n = 60$, $m = 90$, $r = 32$, $r_5 = 12$ y $r_6 = 20$.

TOMADO: 2017C1R2 21-JUL-2017.

270. El grafo de un rombicosidodecaedro (esta palabra existe de verdad) es un grafo conexo que admite una representación planar donde cada vértice está en el borde de exactamente cuatro regiones: una triangular, una pentagonal, y dos opuestas con forma de cuadrilátero (una región triangular es aquella cuyo borde es un ciclo simple de 3 vértices; análogamente para los otros tipos de regiones). En la siguiente figura se muestran las regiones en cuyos bordes está un vértice dado.



Sean n y m las cantidades de vértices y ejes del grafo, respectivamente. Sea r la cantidad total de regiones en la representación planar mencionada, de las cuales r_3 son triangulares, r_5 son pentagonales, y r_4 tienen forma de cuadrilátero.

(a) Demostrar que $r_3 = n/3$, $r_4 = 2n/4$ y $r_5 = n/5$.
(b) Demostrar que $r = 31n/30$.
(c) Demostrar que $m = 4n/2$.
(d) Determinar los valores exactos de n, m, r, r_3, r_4 y r_5. Justificar.

SOLUCIÓN: Es el Ejercicio 271 instanciado con el rombicosidodecaedro. La solución es la de ese ejercicio instanciada con los datos del rombicosidodecaedro. Resulta $n = 60$, $m = 120$, $r = 62$, $r_3 = 20$, $r_4 = 30$ y $r_5 = 12$.

TOMADO: 2015C2R2 18-DIC-2015 (excepto el tercer punto).

271. El grafo de un sólido de Arquímedes es un grafo conexo que admite una representación planar donde cada vértice está en el borde de exactamente p regiones de las cuales p_i tienen como borde a un ciclo simple de i ejes, para i en cierto conjunto $S \subset \mathbb{N}$. Notar que $p = \sum_{i \in S} p_i$.

Sean n y m las cantidades de vértices y ejes del grafo, respectivamente. Sea r la cantidad total de regiones en la representación planar mencionada, de las cuales r_i tienen como borde a un ciclo simple de i ejes para $i \in S$.

(a) Demostrar que $r_i = np_i/i$ para $i \in S$.
(b) Demostrar que $r = n \sum_{i \in S} p_i/i$.
(c) Demostrar que $m = np/2$.
(d) Determinar los valores exactos de n, m, r y r_i para $i \in S$. Justificar.

SOLUCIÓN:

(a) Cada nodo aporta a p_i regiones cuyo borde es C_i, de modo que el total de aportes para esas regiones es np_i. Cada región cuyo borde es C_i requiere i aportes, de modo que el total de requerimientos de esas regiones es ir_i. Como todo lo aportado es requerido y todo lo requerido es aportado, se cumple $ir_i = np_i$, o equivalentemente $r_i = np_i/i$.
(b) Como sólo existen las regiones mencionadas, por el punto anterior tenemos $r = \sum_{i \in S} r_i = \sum_{i \in S} np_i/i = n \sum_{i \in S} p_i/i$.

(c) Dado que el grafo es p-regular, por sumatoria de grados tenemos $2m = \sum_v d(v) = \sum_v p = np$. O también, dado que el grafo es planar, por sumatoria de bordes y el primer punto tenemos $2m = \sum_{i \in S} ir_i = \sum_{i \in S} inp_i/i = n \sum_{i \in S} p_i = np$. En cualquier caso resulta $2m = np$, o equivalentemente $m = np/2$.

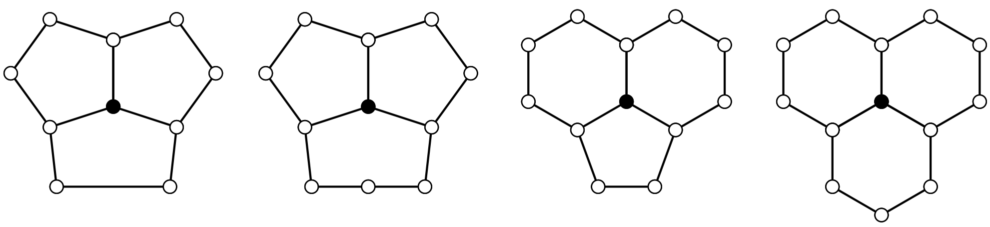
(d) Dado que el grafo es planar y conexo, cumple la ecuación de Euler $n - m + r = 2$. Si remplazamos m y r de los puntos anteriores en la ecuación, nos queda $n - np/2 + n \sum_{i \in S} p_i/i = 2$, lo que equivale a $n \left(1 - p/2 + \sum_{i \in S} p_i/i\right) = 2$, y en definitiva a $n = 2/\left(1 - p/2 + \sum_{i \in S} p_i/i\right)$. Una vez determinado n a partir de los datos, por medio de los tres primeros puntos se pueden determinar m, r y r_i para $i \in S$.

NOTA: Encontrado originalmente en Internet (rombicosidodecaedro). Este es en realidad un meta-ejercicio que puede instanciarse con cualquiera de los sólidos de Arquímedes. A continuación pueden verse los datos y los resultados para los distintos sólidos existentes.

nombre	p_i			i			p	n	m	r	r_i		
tetraedro truncado (*)	1	2		3	6		3	12	18	8	4	4	
cuboctaedro (*)	2	2		3	4		4	12	24	14	8	6	
cubo truncado	1	2		3	8		3	24	36	14	8	6	
octaedro truncado (*)	1	2		4	6		3	24	36	14	6	8	
rombicuboctaedro (*)	1	3		3	4		4	24	48	26	8	18	
cuboctaedro truncado	1	1	1	4	6	8	3	48	72	26	12	8	6
cubo romo	4	1		3	4		5	24	60	38	32	6	
icosidodecaedro	2	2		3	5		4	30	60	32	20	12	
dodecaedro truncado	1	2		3	10		3	60	90	32	20	12	
icosaedro truncado (*, fútbol)	1	2		5	6		3	60	90	32	12	20	
rombicosidodecaedro	1	2	1	3	4	5	4	60	120	62	20	30	12
icosidodecaedro truncado	1	1	1	4	6	10	3	120	180	62	30	20	12
dodecaedro romo	4	1		3	5		5	60	150	92	80	12	

El significado de (*) puede consultarse en la solución del Ejercicio 272.

272. El grafo de una pelota de fútbol generalizada es un grafo conexo que admite una representación planar donde cada vértice está en el borde de exactamente tres regiones, cada una de las cuales puede ser pentagonal o hexagonal (una región pentagonal es aquella cuyo borde es un ciclo simple de 5 vértices; análogamente para hexagonal). En la siguiente figura se muestran las cuatro posibles situaciones en las que puede estar cada vértice del grafo.



Sean n y m las cantidades de vértices y ejes del grafo, respectivamente. Sea r la cantidad total de regiones en la representación planar mencionada, de las cuales r_5 son pentagonales y r_6 son hexagonales.

(a) Demostrar que $m = (5r_5 + 6r_6)/2$.

(b) Demostrar que $n = (5r_5 + 6r_6)/3$.

(c) Determinar el valor exacto de r_5. Justificar.

(d) ¿Es posible que todas las regiones sean pentagonales? En caso afirmativo exhibir un grafo que lo cumpla y justificar; en caso negativo demostrar.

(e) ¿Es posible que todas las regiones sean hexagonales? En caso afirmativo exhibir un grafo que lo cumpla y justificar; en caso negativo demostrar.

SOLUCIÓN:

(a) Por sumatoria de bordes de regiones.

(b) Por el punto anterior y sumatoria de grados (el grafo es 3-regular).

(c) Hay que remplazar m, n y $r = r_5 + r_6$ en la ecuación de Euler. Resulta $r_5 = 12$.

(d) Si $r_6 = 0$ por los tres primeros puntos tenemos $m = 30$ y $n = 20$. Como sea, el grafo resulta platónico, y concretamente es el grafo del dodecaedro (el del juego de Hamilton).

(e) No es posible porque $r_5 \neq 0$ como ya vimos.

En este ejercicio se trata de ver que la cantidad de determinado polígono (en este caso pentágonos) en una generalización de cierto sólido de Arquímedes (en este caso el icosaedro truncado), sólo depende del grado de los vértices (en este caso 3) y de los tipos de polígonos involucrados (en este caso pentágonos y exágonos). Se puede ver que lo mismo ocurre para otros sólidos de Arquímedes. En el caso general, tenemos un grafo p-regular con r_i regiones cuyo borde es C_i, para i en cierto conjunto $S \subset \mathbb{N}$. En tal caso por sumatoria de bordes y de grados tenemos $\sum_{i \in S} ir_i = 2m = np$, de modo que $m = \sum_{i \in S} ir_i/2$ y $n = \sum_{i \in S} ir_i/p$; además, $r = \sum_{i \in S} r_i$. Si remplazamos los tres valores en la ecuación de Euler resulta $2 = n - m + r = \sum_{i \in S} ir_i/p - \sum_{i \in S} ir_i/2 + \sum_{i \in S} r_i = \sum_{i \in S} r_i(i/p - i/2 + 1)$, es decir, $2 = \sum_{i \in S} r_i(i/p - i/2 + 1)$. Para que una cantidad de regiones esté unívocamente determinada, deben anularse los coeficientes $i/p - i/2 + 1$ de las otras regiones. Es decir, todos los $i \in S$, excepto uno, deben cumplir $i/p - i/2 + 1 = 0$, o equivalentemente $i = 2p/(p - 2)$. Como la expresión $2p/(p - 2)$ depende únicamente de p, existe a lo sumo un coeficiente que puede anularse, de modo que debemos restringirnos al caso $|S| = 2$. Además, es necesario que $2p/(p-2) \in S$, ya que caso contrario el coeficiente que podría anularse no aparecería en la ecuación de Euler. Estas dos condiciones (es decir, $|S| = 2$ y $2p/(p - 2) \in S$) se cumplen para cinco de los sólidos de Arquímedes, marcados con (*) en la tabla del Ejercicio 271. Como vimos, estas condiciones aseguran que en la ecuación de Euler se anulan todos los coeficientes excepto uno, de modo que la misma se reduce a $2 = r_i(i/p - i/2 + 1)$, y es así que la cantidad de regiones que queda unívocamente determinada es $r_i = 2/(i/p - i/2 + 1)$. Notar que este valor es necesariamente entero, ya que existe al menos un grafo con esas características (el del sólido de Arquímedes correspondiente). Podemos preguntarnos como en el cuarto punto si es posible que no haya otras regiones más que las unívocamente determinadas. Para los cinco casos marcados con (*) en la tabla del Ejercicio 271 la respuesta es afirmativa: la generalización sin C_6 del tetraedro truncado sólo tiene C_3 y es el tetraedro, la generalización sin C_4 del cuboctaedro sólo tiene C_3 y es el octaedro, la generalización sin C_6 del octaedro truncado sólo tiene C_4 y es el cubo, la generalización sin C_4 del rombicuboctaedro sólo tiene C_3 y es el octaedro, y como vimos en este ejercicio, la generalización sin C_6 del icosaedro truncado sólo tiene C_5 y es el dodecaedro. Notar que aparecen cuatro de los cinco sólidos platónicos (únicamente falta el icosaedro).

NOTA: Encontrado originalmente en Internet (tres primeros puntos).

TOMADO: 2017C1P2 07-JUL-2017.

273. Sea G un grafo planar no necesariamente conexo de $n \geq 3$ vértices. Sea r la cantidad de regiones en cualquier representación planar de G. Demostrar que $r \leq 2n - 4$.

SUGERENCIA: Considerar primero el caso de G conexo.

SOLUCIÓN: Caso G conexo: Sea m la cantidad de ejes de G. Como G es planar y conexo se cumple la ecuación de Euler $n - m + r = 2$. Además, como G es planar y $n \geq 3$ se cumple $m \leq 3n - 6$. Por lo tanto $r = 2 - n + m \leq 2 - n + 3n - 6 = 2n - 4$, es decir, $r \leq 2n - 4$.

Caso G no conexo: Sean $G_1, G_2, \dots G_k$ las componentes conexas de G. Para $i = 1, 2, \dots k$, consideremos una representación planar de G_i, y un nodo v_i en la región exterior de tal representación. Consideremos la representación planar de G que se obtiene por yuxtaposición de las representaciones planares de $G_1, G_2, \dots G_k$ (se puede demostrar que la cantidad de regiones no depende de la representación). Si en esa representación agregamos de manera planar el eje (v_i, v_{i+1}) para $i = 1, 2, \dots k - 1$, obtenemos un grafo G' planar conexo. Por la primera parte, la desigualdad vale para G', y por lo tanto para G ya que ambos grafos tienen la misma cantidad de regiones y de nodos. Probablemente haya demostración alternativa aplicando la primera parte a cada componente conexa de al menos 3 nodos y sumando. Las componentes conexas con a lo sumo 2 nodos no afectan la desigualdad, ya que no modifican el valor de r y hacen crecer el de n.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2016C2R2 19-DIC-2016.

274. Un grafo (simple) se dice 1-árbol si es un árbol con un eje agregado.

   (a) Sea G un grafo con ejes, e un eje de G, y H_e el grafo que se obtiene al subdividir e en G. Demostrar que G y H_e tienen la misma cantidad de circuitos simples. Concluir que la aplicación de sucesivas subdivisiones preserva la cantidad de circuitos simples.

   (b) Usar el punto anterior para demostrar que cualquier 1-árbol es planar.

   (c) Usar la ecuación de Euler para demostrar que cualquier 1-árbol es planar.

   SOLUCIÓN:

   (a) Los circuitos simples de G que no contienen al eje subdividido, no se ven afectados. Por cada circuito simple de G que sí lo contiene, en H_e hay un circuito simple que en vez de usar el eje subdividido usa los ejes resultado de la subdivisión; los circuitos simples son distintos porque difieren los originales. Recíprocamente, por cada circuito simple que en H_e usa los ejes resultado de la subdivisión, en G hay un circuito simple que usa al eje original. Por lo tanto, hay una biyección entre los circuitos simples de ambos grafos, de modo que la cantidad es la misma.

   (b) Un 1-árbol tiene exactamente un circuito simple, lo que implica que cualquier subgrafo de un 1-árbol tiene a lo sumo un circuito simple, y por el primer punto al subdividir cualquier cantidad de veces tal subgrafo se obtiene un grafo con a lo sumo un circuito simple. Por otro lado la cantidad de circuitos simples de K_5 y $K_{3,3}$ es mayor que uno, y por el primer punto al subdividir cualquier cantidad de veces a cualquiera de ellos se obtiene un grafo con más de un circuito simple. Por el Ejercicio 5.19.d de la Práctica, dos grafos isomorfos tienen la misma cantidad de circuitos simples, lo que implica que un subgrafo de un 1-árbol no puede ser homeomorfo ni a K_5 ni a $K_{3,3}$, y por lo tanto cualquier 1-árbol es planar por Kuratowski.

   (c) Sea T el árbol al cual se le agrega un eje, n su cantidad de vértices, y $m = n - 1$ su cantidad de ejes. Por el Ejercicio 9.2 de la Práctica, T es planar. Consideremos entonces cualquier representación planar de T, y sea r la cantidad de regiones de la representación. Como todo árbol es conexo, entonces T cumple la ecuación de Euler, de modo que $r = 2 - n + m = 2 - n + (n - 1) = 1$. Es así que los extremos del eje agregado están en el borde una misma región de la representación (la única región), por lo que es posible agregar el eje manteniendo la planaridad.

   NOTA: Propuesto originalmente por Guido Tagliavini Ponce. Es similar al Ejercicio 275.

   TOMADO: 2016C1P2 02-JUL-2016 (dos primeros puntos).

275. (a) Sea G un grafo con ejes, e un eje de G, y H_e el grafo que se obtiene al subdividir e en G. Demostrar que las secuencias gráficas de G y H_e son iguales, excepto por el hecho de que la segunda tiene un elemento agregado de valor 2. Concluir que la aplicación de sucesivas subdivisiones preserva la secuencia gráfica original, excepto por el hecho de que aparecen elementos agregados de valor 2.

   (b) Sea G un grafo que tiene a lo sumo 4 nodos de grado mayor o igual a 3. Usar el punto anterior para demostrar que G es planar.

   SOLUCIÓN:

   (a) Trivial.

   (b) Cualquier subgrafo de G tiene a lo sumo 4 nodos de grado al menos 3, y por el primer punto al subdividir cualquier cantidad de veces a cualquier subgrafo de G se obtiene un grafo que tiene a lo sumo 4 nodos de grado al menos 3. Por otro lado K_5 y $K_{3,3}$ tienen al menos 5 nodos de grado al menos 3, y por el primer punto al subdividir cualquier cantidad de veces a cualquiera de ellos se obtiene un grafo que tiene al menos 5 nodos de grado al menos 3. Por el Ejercicio 5.19.c de la Práctica, dos grafos isomorfos tienen los mismos grados, lo que implica que un subgrafo de G no puede ser homeomorfo ni a K_5 ni a $K_{3,3}$, y por lo tanto G es planar por Kuratowski.

   NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez. Es similar al Ejercicio 274.

   TOMADO: 2018C1R2 20-JUL-2018.

276. Decidir si cada uno de los grafos indicados es planar, donde M_k es una correspondencia formada por k ejes. Justificar.

   (a) $K_{4,4} - M_4$
   (b) $K_{4,4} - M_3$
   (c) $K_6 - M_3$
   (d) $K_6 - M_2$
   (e) el grafo que se obtiene subdividiendo un eje en $K_6 - M_3$

   SOLUCIÓN: Sean n y m las cantidades de nodos y ejes de cada grafo, respectivamente. Notemos que en cada punto el grafo indicado es único, salvo isomorfismos.

   (a) Es $n = 8$ y $m = 16 - 4 = 12$. El grafo es bipartito porque $K_{4,4}$ lo es y $n \geq 2$. Como tiene suficientes nodos/ejes/ciclos, si fuera planar debería cumplirse $m \leq 2n - 4 = 12$, lo cual ocurre. Con un poco de esfuerzo es posible encontrar una representación planar. El grafo es el cubo Q_3.
   (b) Es $n = 8$ y $m = 16 - 3 = 13$. El grafo es bipartito porque $K_{4,4}$ lo es y $n \geq 2$. Como tiene suficientes nodos/ejes/ciclos, si fuera planar debería cumplirse $m \leq 2n - 4 = 12$, lo cual no ocurre. Por lo tanto, no es planar.
   (c) Es $n = 6$ y $m = 6 \times 5/2 - 3 = 12$. Como $n \geq 3$, si fuera planar debería cumplirse $m \leq 3n - 6 = 12$, lo cual ocurre. Con bastante esfuerzo es posible encontrar una representación planar. El grafo es el octaedro.
   (d) Es $n = 6$ y $m = 6 \times 5/2 - 2 = 13$. Como $n \geq 3$, si fuera planar debería cumplirse $m \leq 3n - 6 = 12$, lo cual no ocurre. Por lo tanto, no es planar.
   (e) Subdividir ejes no cambia la planaridad. Como $K_6 - M_3$ es planar, la subdivisión también lo es.

   NOTA: Encontrado originalmente en Internet.

   TOMADO: 2015C2P2 30-NOV-2015.

277. Un grafo se dice no-planar crítico si y sólo si es no planar pero al sacarle cualquier vértice se obtiene un grafo planar.

   Sea $f(n)$ la cantidad de grafos no isomorfos de n vértices que son no-planares críticos. Decidir para cada entero $n = 1, 2, \ldots 6$ si $f(n)$ vale 0, 1 o al menos 2. Justificar.

   SOLUCIÓN: Para $n \leq 4$ todo grafo es planar, de modo que $f(1) = f(2) = f(3) = f(4) = 0$. Para $n = 5$ el único grafo no planar es K_5, ya que todo otro grafo de 5 vértices es subgrafo de $K_5 - e$, que es planar; como $K_5 - v = K_4$ que es planar, resulta $f(5) = 1$. Para $n = 6$ $K_{3,3}$ es no-planar crítico ya que es no planar pero $K_{3,3} - v = K_{2,3}$ que es planar; otro ejemplo es K_6 sin dos ejes que formen una correspondencia, el cual es no planar porque no cumple que $13 = m \leq 3n - 6 = 12$, pero si se le saca cualquier vértice resulta un subgrafo de $K_5 - e$, que es planar; por lo tanto, $f(6) \geq 2$.

   NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

   TOMADO: 2015C1R2 17-JUL-2015.

278. Dada una representación planar de un grafo planar G, se define su (pseudo/multi) grafo dual G^* de la siguiente manera: por cada región de la representación planar de G se agrega un vértice en G^*; por cada eje de la representación planar de G se agrega un eje en G^* entre los vértices correspondientes a las regiones que están a ambos lados de e en G. Un grafo planar G se dice autodual si es isomorfo a su dual.

   Exhibir todos los árboles no isomorfos que son autoduales. Justificar.

   SOLUCIÓN: Por Euler cualquier representación planar de un árbol tiene una única región. Por lo tanto, el dual de un árbol de n nodos tiene un único nodo con $m = n - 1$ self-loops, el cual no es isomorfo al árbol original mientras haya algún self-loop, por lo que debe ser $n = 1$.

   Otra forma de argumentarlo es usando el Ejercicio 279c que dice que si un grafo es autodual entonces $2n - m = 2$, ya que para árboles debe ser además $m = n - 1$, lo que implica $n = 1$. Falta comprobar que $n = 1$ efectivamente sirve, lo cual es inmediato.

   TOMADO: 2015C1P2 04-JUL-2015.

279. Dada una representación planar de un grafo planar G, se define su (pseudo/multi) grafo dual G^* de la siguiente manera: por cada región de la representación planar de G se agrega un vértice en G^*; por cada eje de la representación planar de G se agrega un eje en G^* entre los vértices correspondientes a las regiones que están a ambos lados de e en G. Un grafo planar G se dice autodual si es isomorfo a su dual. Si G es autodual, entonces G^* no es pseudo ni multi grafo.

Sea G un grafo planar.

(a) Demostrar que G^* es conexo.

(b) Demostrar que si G es autodual entonces es conexo.

(c) Demostrar que si $G = (V, E)$ es autodual entonces $2|V| - |E| = 2$.

(d) Exhibir todos los grafos no isomorfos de 4 o menos vértices que son autoduales. Justificar.

SOLUCIÓN:

(a) Si consideramos la representación planar de G, tomamos cualquier punto de la región exterior, y trazamos una recta hacia cualquier punto de una región dada, esa recta va a cortar determinados ejes, lo cual representa un camino en G^*. Es decir, en G^* hay un camino desde el vértice que representa la región exterior de G hacia cualquier otro vértice, por lo que G^* es conexo.

(b) Por definición, G es isomorfo a G^*, que es conexo.

(c) Como G y G^* son isomorfos, resulta $n(G) = n(G^*)$. Pero por definición de G^* es $n(G^*) = r(G)$, de modo que $n(G) = r(G)$. Si remplazamos en la ecuación de Euler r por n, resulta la igualdad pedida.

(d) Hay que usar el punto anterior. Para $n = 1$ resulta $m = 0$, por lo que $G = K_1$, que es autodual. Para $n = 2$ resulta $m = 2$, por lo que no existe grafo (simple) G. Para $n = 3$ resulta $m = 4$, por lo que no existe grafo (simple) G. Para $n = 4$ resulta $m = 6$, por lo que $G = K_4$, que es autodual.

NOTA: Propuesto originalmente por Federico Lebrón.

TOMADO: 2011C2P2 05-DIC-2011 (con un punto adicional).

280. Dada una representación planar de un grafo planar G, se define su (pseudo/multi) grafo dual G^* de la siguiente manera: por cada región de la representación planar de G se agrega un vértice en G^*; por cada eje de la representación planar de G se agrega un eje en G^* entre los vértices correspondientes a las regiones que están a ambos lados de e en G. Un grafo planar G se dice autodual si es isomorfo a su dual.

Determinar para qué valores de n, p y q los siguientes grafos son planares. Para cada grafo que sea planar, obtener una representación planar, calcular su dual, y decidir si el grafo es autodual. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) árbol de n vértices

(e) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(f) $K_{p,q}$

SOLUCIÓN:

K_n: Para $n \geq 5$ el grafo K_n no es planar porque contiene a K_5 que no es planar. Cada caso con $n \leq 4$ es un caso particular de alguno de los puntos siguientes, ya que $K_1 = P_1$, $K_2 = P_2$, $K_3 = C_3$ y $K_4 = W_3$.

C_n: Es planar. El dual tiene dos nodos unidos por n ejes. Para $n \geq 3$ resulta un multigrafo, de modo que no es autodual.

P_n: Es planar. El dual tiene un único nodo con $n-1$ self-loops. Sólo es autodual cuando $n-1 = 0$, es decir, el grafo trivial.

W_n: Es planar, ya que basta dibujar el ciclo de manera planar y agregar el vértice universal en el medio. El dual es W_n. Es autodual.

$K_{p,q}$: Para $p \geq 3$ y $q \geq 3$ el grafo $K_{p,q}$ no es planar porque contiene a $K_{3,3}$ que no es planar. Para $p \leq 2$ o $q \leq 2$ es planar (basta dibujarlo); en el caso particular de $p = 2$ o $q = 2$ se puede dibujar haciendo un sandwich donde cada pan es un nodo. El dual siempre es pseudo o multi grafo, de modo que no es autodual.

NOTA: No se incluye GBC n m (variable), ni ABC h (no aporta).

TOMADO: 2011C2R2 21-DIC-2011 (cierta parte), 2012C1P2 11-JUL-2012 (cierta parte).

281. (a) Exhibir un grafo no planar que tenga la menor cantidad posible de nodos. Justificar.
(b) Exhibir un grafo no planar que tenga la menor cantidad posible de ejes. Justificar.

SOLUCIÓN:

(a) K_5. No es planar por teoremas de caracterización. Cualquier grafo con menos nodos es planar por estar incluído en K_4, que es planar; también puede argumentarse usando los teoremas de caracterización.

(b) $K_{3,3}$. No es planar por teoremas de caracterización. Cualquier grafo con menos ejes es planar, porque de no serlo, debería contener un subgrafo contraíble a $K_{3,3}$ o K_5; el supuesto grafo tendría a lo sumo 8 ejes, su supuesto subgrafo también a lo sumo 8 ejes, y el subgrafo contraído también a lo sumo 8 ejes (cada contracción elimina al menos un eje); el resultado no sería $K_{3,3}$ ni K_5, que tienen 9 y 10 ejes respectivamente; también puede argumentarse usando homeomorfismo en vez de contracción.

TOMADO: 2004C1R2 09-AGO-2004.

282. Sea G un grafo planar no necesariamente conexo. ¿Es cierto que cualquier representación planar de G tiene la misma cantidad de regiones? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN: Es cierto, porque en la ecuación de Euler la cantidad de nodos y de ejes no cambia al variar la representación.

TOMADO: 2004C1P2 14-JUL-2004.

283. Sea G un grafo planar. Diseñar un algoritmo polinomial que encuentre un subgrafo completo máximo de G. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SUGERENCIA: Usar que K_5 no es subgrafo de G.

SOLUCIÓN: Examinar todos los conjuntos posibles de hasta 4 nodos.

TOMADO: 2007C1R2 08-AGO-2007, 2012C2P2 05-DIC-2012.

284. Sea G un grafo planar de n vértices y m ejes.

(a) Demostrar que si $n < 6$ entonces existe un vértice v tal que $d(v) < 4$.
(b) Demostrar que si $m < 30$ entonces existe un vértice v tal que $d(v) < 5$.
(c) Demostrar que si $m < 12$ entonces existe un vértice v tal que $d(v) < 4$.

SOLUCIÓN: Todos los puntos se resuelven de manera similar. Si se pide demostrar que existe un vértice v tal que $d(v) < d$, por absurdo suponemos que para todo vértice $d(v) \geq d$, lo que implica $2m = \sum_{i=1}^{n} d(v_i) \geq \sum_{i=1}^{n} d = nd$, y por lo tanto $2m \geq nd$. Además, $n \geq d(v) + 1 \geq d + 1 \geq 3$, de modo que $n \geq 3$, y al ser G planar se cumple $m \leq 3n - 6$. Si despejamos m de $2m \geq nd$ obtenemos $m \geq nd/2$ de modo que $nd/2 \leq m \leq 3n - 6$, y por lo tanto $nd/2 \leq 3n - 6$ que si $d \leq 5$ equivale a $n \geq 12/(6 - d)$; para n cualquiera y $d = 6$ resulta $3n \leq 3n - 6$, lo cual es absurdo (Ejercicio 9.5.a de la Práctica); para $n < 12$ y $d = 5$ resulta $n \geq 12/(6 - 5) = 12$, lo cual es absurdo (Ejercicio 9.5.b de la Práctica); para $n < 6$ y $d = 4$ resulta $n \geq 12/(6 - 4) = 6$, lo cual es absurdo (primer punto). Si despejamos n de $2m \geq nd$ obtenemos $n \leq 2m/d$ de modo que $m \leq 3n - 6 \leq 6m/d - 6$, y por lo tanto $m \leq 6m/d - 6$ que si $d \leq 5$ equivale a $m \geq 6d/(6 - d)$; para m cualquiera y $d = 6$ resulta $m \leq m - 6$, lo cual es absurdo (Ejercicio 9.5.a de la Práctica); para $m < 30$ y $d = 5$ resulta $m \geq 6 \times 5/(6 - 5) = 30$, lo cual es absurdo (segundo punto); para $m < 12$ y $d = 4$ resulta

$m \geq 6 \times 4/(6-4) = 12$, lo cual es absurdo (tercer punto). Los valores $d = 3$ y $d = 2$ se pueden analizar de manera similar, aunque no producen resultados interesantes en relación a planaridad. El valor $d = 1$ no se puede analizar con este mecanismo porque no se puede asegurar $n \geq 3$, y tampoco es interesante.

NOTA: Propuesto originalmente por Michel Mizrahi (un punto).

TOMADO: 2013C1P2 10-JUL-2013 (incluso $m < 6$).

285. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

(a) Determinar todos los n para los cuales existe al menos un grafo planar 3-regular de n vértices. Justificar.

(b) Exhibir todos los grafos no isomorfos planares 3-regulares de $n \leq 6$ vértices. Justificar.

SOLUCIÓN:

(a) Debe ser n par porque en todo grafo la suma de grados es par. Debe ser $n \geq 4$ para que el grado pueda ser 3. Para cada $n \geq 4$ par existe un grafo como el pedido: para $n = 4$ se puede tomar K_4, y para $n \geq 6$ par se puede tomar un prisma de base $C_{n/2}$. Alternativamente para $n \geq 8$ se puede tomar un grafo no conexo formado por cierta cantidad de copias del grafo mencionado para $n = 4$ junto con a lo sumo una copia del grafo mencionado para $n = 6$.

(b) Por el primer punto debe ser $n = 4$ o $n = 6$.

Para $n = 4$, como el grafo debe ser 3-regular cada vértice es adyacente a los otros 3, de modo que el grafo es K_4 que es planar.

Para $n = 6$, el complemento es 2-regular, es decir, colección de ciclos. Las únicas posibilidades para el complemento son $2C_3$ y C_6. Los grafos correspondientes sin complementar son $K_{3,3}$ y un prisma de base triangular. El primero no es planar por Kuratowski, mientras que el segundo sí lo es por el primer punto.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez. Ver en Ejercicio 289 la lista de ejercicios donde se consideran grafos planares regulares.

TOMADO: 2018C1P2 06-JUL-2018.

286. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

(a) Demostrar que si un grafo es planar y 4-regular entonces tiene al menos 6 vértices.

(b) Exhibir todos los grafos no isomorfos 4-regulares de 6 vértices. Justificar. Demostrar que *son* planares.

SUGERENCIA: ¿Cómo es el complemento de un grafo 4-regular de 6 vértices?

(c) Exhibir todos los grafos no isomorfos 4-regulares de 7 vértices. Justificar. Demostrar que *no son* planares.

(d) Exhibir todos los n para los cuales existe al menos un grafo planar 4-regular de n vértices. Justificar.

SOLUCIÓN:

(a) Sea n la cantidad de nodos y m la cantidad de ejes. Como el grafo es planar y tiene al menos 3 vértices (porque todos los vértices tienen grado 4), se cumple que $m \leq 3n - 6$. Como la suma de los grados es $2m$, y todos los vértices tienen grado 4, resulta $m = 2n$, y reemplazando m por $2n$ en la desigualdad anterior, se obtiene $n \geq 6$.

Alternativamente, si todo vértice v es tal que $d(v) = 4$, entonces todo vértice v es tal que $d(v) \geq 4$, y usando una contrarecíproca del Ejercicio 284 resulta $n \geq 6$.

Alternativamente, por absurdo supongamos que $n \leq 5$. Sin embargo, debe ser $n \geq 5$, porque todos los nodos tienen grado 4. De tal modo, el grafo está formado por exactamente 5 nodos de grado 4, es decir, el grafo es K_5, que no es planar, lo cual es absurdo.

(b) El complemento es 1-regular, es decir, sus componentes conexas son K_2, y como tiene 6 nodos es $3K_2$. El grafo 4-regular es el complemento de ese. El grafo resulta ser un antiprisma de base triangular, también conocido como octaedro (uno de los grafos platónicos).

Hay que demostrar que el complemento de $3K_2$ es planar. Se lo puede dibujar como indica el último punto, o de cualquier otra manera y comprobar que es planar.

(c) El complemento es 2-regular, es decir, sus componentes conexas son ciclos simples. Como cada componente conexa debe tener al menos 3 vértices, el complemento es C_7 o $C_3 \cup C_4$. Los grafos 4-regulares son los complementos de esos.

Hay que demostrar que los complementos de C_7 y de $C_3 \cup C_4$ no son planares. Eso es cierto, ya que ambos complementos contienen un subgrafo homeomorfo a $K_{3,3}$ (para el caso del complemento de $C_3 \cup C_4$, el subgrafo es exactamente $K_{3,3}$).

(d) Por los puntos anteriores, debe ser $n \geq 6$ y $n \neq 7$.

Para los $n \geq 6$ pares, un posible grafo planar 4-regular de n vértices es C_n, junto con un $C_{n/2}$ interior formado conectando uno de cada dos nodos, junto con un $C_{n/2}$ exterior formado conectando los nodos que no se usaron en el otro $C_{n/2}$. El grafo resulta ser un antiprisma de base $C_{n/2}$.

Para los $n \geq 9$ impares, armar el grafo para $n-1$ nodos, elegir 2 arcos no consecutivos de uno de los $C_{(n-1)/2}$, suprimir esos arcos, y conectar sus extremos a un nuevo nodo.

NOTA: Ver en Ejercicio 289 la lista de ejercicios donde se consideran grafos planares regulares.

TOMADO: 2005C2P2 06-DIC-2005 (dos primeros puntos), 2005C2R2 22-DIC-2005 (excepto el segundo punto).

287. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Variante 1: Sea G un grafo de n vértices, planar y d-regular.

(a) Demostrar que $d \leq 5$.
(b) Demostrar que si $d = 5$ entonces $n \geq 12$.
(c) Exhibir un grafo planar 5-regular que tenga la menor cantidad posible de vértices. Justificar.

Variante 2: Determinar todos los $d \in \mathbb{N}_0$ tales que existe un grafo planar d-regular. Para cada uno de ellos exhibir un grafo planar d-regular que tenga la menor cantidad posible de vértices. Justificar.

SOLUCIÓN: La Variante 1 es más fácil porque se restringe al caso $d = 5$ y el enunciado da más información.

Se puede demostrar que tiene que ser $d \leq 5$ utilizando el Ejercicio 9.5.a de la Práctica o como en el Ejercicio 284.

Para $d = 5$ se puede demostrar que tiene que ser $n \geq 12$ utilizando el Ejercicio 9.5.b de la Práctica o como en el Ejercicio 284. Se puede conseguir un grafo planar 5-regular de 12 vértices tomando un antiprisma de base pentagonal y agregando un vértice en cada base, adyacente a todos los vértices de esa base; este grafo también es conocido como icosaedro (uno de los grafos platónicos). El grafo tiene la menor cantidad posible de vértices porque para $d = 5$ es $n \geq 12$.

Para $d = 4$ se puede demostrar que tiene que ser $n \geq 6$ como en el Ejercicio 284 o el Ejercicio 286a. Se puede conseguir un grafo planar 4-regular de 6 vértices como en el Ejercicio 286b. El grafo tiene la menor cantidad posible de vértices porque para $d = 4$ es $n \geq 6$.

Para $d \leq 3$ tiene que ser $n \geq d + 1$ porque dado un vértice cualquiera, hay otros d adyacentes al mismo. Un grafo planar d-regular de $d + 1$ vértices es K_{d+1}, que por lo dicho tiene la menor cantidad posible de vértices.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Ver en Ejercicio 289 la lista de ejercicios donde se consideran grafos planares regulares.

TOMADO: 2018C2P2 03-DIC-2018 (Variante 1), 2019C1P2 05-JUL-2019 (Variante 2).

288. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Sea G un grafo de 7 vértices.

(a) Demostrar que si G es 4-regular entonces no es planar.
(b) Demostrar que si G es regular, entonces G no es planar o G^c no es planar.

SOLUCIÓN:

(a) Ver Ejercicio 286c.

(b) El grafo G tiene 7 vértices y es d-regular. Por la cantidad de vértices que tiene el grafo, d puede tomar valores entre 0 y 6, y debe ser par porque la sumatoria de grados es par. Además, por el Ejercicio 49 el grafo G^c es $(6-d)$-regular. Para $d=0$ o $d=6$ tenemos respectivamente que G^c o G es isomorfo a K_6, que no es planar. Para $d=2$ o $d=4$ tenemos respectivamente que G^c o G es 4-regular, que no es planar por el primer punto.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Ver en Ejercicio 289 la lista de ejercicios donde se consideran grafos planares regulares.

289. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Sea G un grafo de n vértices, planar y d-regular.

(a) Demostrar que $d \leq 5$.

(b) Demostrar que si $d = 5$ entonces $n \geq 12$.

(c) Demostrar que si G^c también es planar entonces $d \leq 4$.

(d) Demostrar que si G^c también es planar entonces $n \leq 9$.

SOLUCIÓN: Por el Ejercicio 49 el grafo G^c es $(n-1-d)$-regular.

(a) Ver Ejercicio 287a.

(b) Ver Ejercicio 287b.

(c) Supongamos por absurdo $d > 4$, lo que implica $d = 5$ teniendo en cuenta el primer punto, y por lo tanto $n \geq 12$ según el segundo punto. Si aplicamos el primer punto a G^c tenemos que $n-1-d = n-6 \leq 5$, lo que implica $n \leq 11$, un absurdo.

(d) Se cumple $n \leq 10$ por la contrarecíproca del Ejercicio 9.5.c de la Práctica. Supongamos por absurdo $n > 9$, lo que implica entonces $n = 10$. Si aplicamos el tercer punto a G tenemos que $d \leq 4$, y si lo aplicamos a G^c tenemos que $n-1-d = 9-d \leq 4$, lo que implica $d \geq 5$, un absurdo.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

En varios ejercicios se consideran grafos de n vértices, planares y d-regulares: $d = 3$ en el Ejercicio 285, $d = 4$ en el Ejercicio 286, $d = 5$ en el Ejercicio 287 (donde también se demuestra que debe ser $d \leq 5$), y $n = 7$ en el Ejercicio 288.

Los grafos coplanares son los grafos planares con complemento planar. En el último punto de este ejercicio se demuestra que si G es un grafo de n vértices, coplanar y regular, entonces $n \leq 9$. Se puede comprobar que tales grafos existen para cada n entre 1 y 6. En el Ejercicio 288 se demuestra que no hay para $n = 7$. Desconozco si existen para $n = 8$ o $n = 9$.

TOMADO: 2019C1R2 19-JUL-2019.

290. (a) ¿Existe un grafo no planar $G = (V,E)$ tal que $|E| \leq 3|V| - 6$?

(b) ¿Existe un grafo no planar bipartito $G = (V,E)$ tal que $|E| \leq 2|V| - 4$?

En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar.

TOMADO: 2001C1P2 10-JUL-2001.

291. Dado un grafo $G = (V,E)$ se define su grosor $t(G)$ (en inglés, *thickness*) como la mínima cantidad de grafos planares cuya unión da por resultado G. Formalmente, $t(G)$ es el mínimo $t \in \mathbb{N}$ para el cual existen t conjuntos $E_1, E_2, \dots E_t$ tales que $E = \cup_{i=1}^{t} E_i$ y $G_i = (V, E_i)$ es planar para $i = 1, 2, \dots t$.

(a) Determinar $t(K_n)$ para $1 \leq n \leq 8$, donde K_n es el grafo completo de n vértices. Justificar.

(b) Demostrar que si $9 \leq n \leq 10$ entonces $2 \leq t(K_n) \leq 3$.

(c) Demostrar que si $n \geq 11$ entonces $t(K_n) \geq 3$.

(d) Demostrar que si H es subgrafo de G entonces $t(H) \leq t(G)$.

(e) Demostrar que si G es un grafo de $n \geq 3$ vértices y m ejes entonces $t(G) \geq \lceil m/(3n-6) \rceil$.

SOLUCIÓN:

(a) Para $1 \leq n \leq 4$ es $t(K_n) = 1$ ya que K_n es planar.
Para $5 \leq n \leq 8$ es $t(K_n) \geq 2$ ya que K_n no es planar. Pero $t(K_8) \leq 2$ ya que K_8 es la unión de dos grafos planares, y por el cuarto punto también vale para los otros. En consecuencia, resulta $t(K_n) = 2$.

(b) Es $t(K_n) \geq 2$ ya que K_n no es planar. Para demostrar $t(K_n) \leq 3$ alcanza con formar K_{10} con la unión de 3 grafos planares, lo cual es posible.

(c) Por el cuarto punto, alcanza con demostrarlo para K_{11}, y para ese grafo el resultado es cierto por el quinto punto.
Alternativamente, por absurdo supongamos que $n \geq 11$ y sin embargo $t(K_n) \leq 2$. Sin pérdida de generalidad podemos suponer que formamos K_n con exactamente 2 grafos planares de n vértices (eventualmente con nodos aislados), y sin ejes repetidos. Por lo tanto, cada grafo planar tiene a lo sumo $3n - 6$ ejes, y entre ambos tienen la cantidad de ejes de K_n, es decir, $n(n-1)/2$. Eso significa que $n(n-1)/2 \leq 6n - 12$, lo que implica $n \leq 10$.

(d) Podemos tomar los mismos grafos que forman G y suprimir todo lo que no esté en H.

(e) Supongamos que podemos formar G con la unión de t grafos planares. Sin pérdida de generalidad, podemos suponer que todos esos grafos tienen exactamente n vértices (eventualmente con nodos aislados), y que no tienen ejes repetidos. Por lo tanto, cada grafo planar tiene a lo sumo $3n - 6$ ejes, y entre todos tienen m ejes. Eso significa que $m \leq t(3n - 6)$, o lo que es lo mismo $t \geq m/(3n - 6)$, lo que equivale a $t \geq \lceil m/(3n - 6) \rceil$ ya que t es entero. Como la desigualdad vale para cualquier t, en particular para el mínimo.

TOMADO: 2001C2P2 11-DIC-2001 (parte del segundo punto), 2008C1P2 11-JUL-2008 (puntos a y c), 2008C1R2 25-JUL-2008 (tres últimos puntos).

292. Dado un grafo $G = (V, E)$ se define su grosor $t(G)$ (en inglés, *thickness*) como la mínima cantidad de grafos planares cuya unión da por resultado G. Formalmente, $t(G)$ es el mínimo $t \in \mathbb{N}$ para el cual existen t conjuntos $E_1, E_2, \ldots E_t$ tales que $E = \cup_{i=1}^{t} E_i$ y $G_i = (V, E_i)$ es planar para $i = 1, 2, \ldots t$.

(a) Sea G un grafo. Demostrar que G es planar si y sólo si $t(G) = 1$.

(b) Demostrar que si G es un grafo de $n \geq 3$ vértices y m ejes entonces $t(G) \geq \lceil m/(3n - 6) \rceil$.

(c) Demostrar que $t(K_n) \geq \lceil n/6 \rceil$.

SOLUCIÓN:

(a) Para la ida basta tomar $G_1 = G$. Para la vuelta debe ser $G = G_1$, de modo que G es planar porque lo es G_1.

(b) Ver Ejercicio 291e.

(c) Para $n \leq 4$ por el primer punto es $t(K_n) = 1 = \lceil n/6 \rceil$. Para $n \geq 3$ por el segundo punto es $t(K_n) \geq \lceil n(n-1)/(3n-6)/2 \rceil = \lceil n(n-1)/(n-2)/6 \rceil \geq \lceil n/6 \rceil$.

NOTA: Propuesto originalmente por Leopoldo Taravilse.

TOMADO: 2016C1R2 15-JUL-2016.

293. Dado un grafo $G = (V, E)$ se define su grosor $t(G)$ (en inglés, *thickness*) como la mínima cantidad de grafos planares cuya unión da por resultado G. Formalmente, $t(G)$ es el mínimo $t \in \mathbb{N}$ para el cual existen t conjuntos $E_1, E_2, \ldots E_t$ tales que $E = \cup_{i=1}^{t} E_i$ y $G_i = (V, E_i)$ es planar para $i = 1, 2, \ldots t$.

(a) Sea $G = (V, E)$ un grafo. Sean M_1 y M_2 dos correspondencias de G. Demostrar que $H = (V, M_1 \cup M_2)$ es planar.

(b) ¿Sigue valiendo la propiedad del punto anterior para tres correspondencias en vez de dos? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

(c) Demostrar que si G es subgrafo de G' entonces $t(G) \leq t(G')$.

(d)

Variante 1: Demostrar que si G es un grafo de n vértices entonces $t(G) \leq \lceil n/2 \rceil$.
SUGERENCIA: Conseguir una partición en correspondencias de los ejes de K_n

Variante 2: Demostrar que si G es un grafo no planar de n vértices entonces $t(G) \leq \lceil (n-1)/2 \rceil$. SUGERENCIA: Conseguir una partición en correspondencias de los ejes de $K_n - E(H_n)$ donde $H_n = C_{n-1} + K_1$ (grafo junta).

Variante 3: Demostrar que si G es un grafo no planar de n vértices entonces $t(G) \leq \lceil (n-2)/2 \rceil$. SUGERENCIA: Conseguir una partición en correspondencias de los ejes de $K_n - E(H_n)$ donde $H_n = C_{n-2} + 2K_1$ (grafo junta).

SOLUCIÓN:

(a) En (V, M_1) y en (V, M_2) cada nodo tiene grado a lo sumo 1, de modo que en H cada nodo tiene grado a lo sumo 2.

En cualquier subgrafo de H cada nodo también tiene grado a lo sumo 2, y en cualquier subdivisión de tal subgrafo ocurre lo mismo (porque subdividir agrega nodos de grado 2). Por otro lado, subdividir a K_5 o $K_{3,3}$ no elimina los nodos de grado mayor o igual que 3 que tienen esos grafos. En consecuencia, los subgrafos de H no pueden ser homeomorfos a K_5 ni a $K_{3,3}$, de modo que H es planar por Kuratowski.

Alternativamente, dado que en H cada nodo tiene grado a lo sumo 2, puede demostrarse que cada componente conexa de H es un nodo aislado, o un camino simple, o un circuito simple. Todas esas componentes conexas son planares, y por lo tanto lo es H.

(b) No, por ejemplo para $G = K_{3,3}$. Si tomamos tres correspondencias M_1, M_2 y M_3 que sean disjuntas de a pares y contengan tres ejes cada una, resulta $H = G = K_{3,3}$ que no es planar.

(c) Ver Ejercicio 291d.

(d) Por el tercer punto $t(G) \leq t(K_n)$, por lo que alcanza con acotar $t(K_n)$.

Para la Variante 1 consideremos un coloreo óptimo de los ejes de K_n, es decir, que usa $\chi'(K_n)$ colores. Los colores particionan a los ejes en correspondencias. Si agrupamos esas correspondencias de a pares (queda una sin aparear porque $\chi'(K_n)$ es impar), y para cada uno de los $\lceil \chi'(K_n)/2 \rceil$ grupos armamos un grafo con esos ejes y los nodos de K_n, obtenemos grafos planares según el primer punto. Como en conjunto todos esos grafos planares tienen a todos los ejes de K_n, resulta $t(K_n) \leq \lceil \chi'(K_n)/2 \rceil$. Por Vizing $\chi'(K_n) \leq \Delta(K_n) + 1 = n$, y por lo tanto $t(K_n) \leq \lceil n/2 \rceil$.

La Variante 2 y la Variante 3 son similares a la Variante 1. Uno de los grafos planares es C. Los otros se obtienen de aparear las correspondencias de $K_n - E(H_n)$ definidas por un coloreo óptimo de los ejes de ese grafo. Resulta $t(K_n) \leq 1 + \lceil \chi'(K_n - E(H_n))/2 \rceil \leq 1 + \lceil (\Delta(K_n - E(H_n)) + 1)/2 \rceil = \lceil (\Delta(K_n - E(H_n)) + 3)/2 \rceil$. Para la Variante 2 tenemos $\Delta = n - 4$ y por lo tanto $t(K_n) \leq \lceil (n-4+3)/2 \rceil = \lceil (n-1)/2 \rceil$, mientras que para la Variante 3 se cumple $\Delta = n - 5$ y por lo tanto $t(K_n) \leq \lceil (n-5+3)/2 \rceil = \lceil (n-2)/2 \rceil$.

Para la Variante 2 la demostración falla para $n \leq 3$, porque H_n no está definido (más aún, la propiedad es falsa para $n = 1$). Eso no es un problema porque al ser G no planar, se tiene $n \geq 5$.

Para la Variante 3 la demostración falla para $n \leq 4$, porque H_n no está definido (más aún, la propiedad es falsa para $n \leq 2$). La demostración incluso falla para $n = 5$, porque $\Delta(K_n - E(H_n)) \neq n - 5 = 0$ ya que en $K_n - E(H_n)$ hay dos vértices adyacentes de grado 1 (y el resto de grado 0). En consecuencia, el caso $n = 5$ debe analizarse por separado, lo cual es sencillo ya que K_5 puede formarse a parir de dos copias de C_5.

Alternativamente, una demostración por inducción es posible, al menos para la Variante 3.

NOTA: Propuesto originalmente por Leopoldo Taravilse.

TOMADO: 2016C1R2 15-JUL-2016 (excepto el tercer punto, Variante 1), 2018C1P2 06-JUL-2018 (excepto el segundo punto, Variante 3).

294. (a) Sea G un grafo planar bipartito. ¿Puede ocurrir que G tenga todos sus vértices de grado mayor o igual que 4?

(b) Sea G un grafo planar bipartito con menos de 8 vértices. ¿Puede ocurrir que G tenga todos sus vértices de grado mayor o igual que 3?

En caso afirmativo dar un ejemplo y justificar; en caso negativo demostrar.

SOLUCIÓN: No puede ocurrir. Si G no tiene ciclos (es decir, es un bosque), siempre tiene al menos un nodo de grado a lo sumo 1. Si G tiene ciclos, vale $m \leq 2n - 4$; por absurdo supongamos que todos los vértices tienen grado mayor o igual a lo dicho; resulta entonces $2m = \sum d(v) \geq kn$, es decir, $m \geq kn/2$, donde $k = 4$ o $k = 3$; si juntamos esta desigualdad con la anterior obtenemos $kn/2 \leq 2n - 4$; para $k = 4$ eso equivale a $4 \leq 0$, mientras que para $k = 3$ equivale a $n \geq 8$. Otra forma de demostrar el segundo punto es notando que debe haber al menos 3 nodos en cada parte, por lo que los únicos casos posibles son $K_{3,3}$ y $K_{3,4}$ que no son planares.

NOTA: Es similar al Ejercicio 9.5 de la Práctica (puntos a y b), salvo que acá los grafos son bipartitos.

TOMADO: 2007C1P2 10-JUL-2007, 2012C2R2 19-DIC-2012.

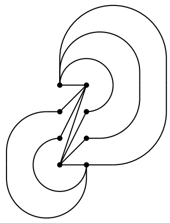
295. (a) Sea G un grafo bipartito de 9 o más vértices. Demostrar que G no es planar o G^c no es planar.
SUGERENCIA: Evitar ser engañados.

(b) Exhibir un grafo bipartito H de 8 vértices tal que tanto H como H^c sean planares. Justificar.

SOLUCIÓN:

(a) Al menos una de las partes contiene 5 o más vértices, por lo que G^c contiene a K_5. Otra forma (más complicada) de resolverlo es similar a la solución del Ejercicio 9.5.c de la Práctica, suponiendo que ambos son planares y por lo tanto para G vale $m \leq 2n - 4$, mientras que para G^c vale $m^c \leq 3n - 6$; si se suman ambas desigualdades se obtiene que debe ser $n \leq 8$.

(b) Cada parte del conjunto de nodos de H debe tener 4 vértices (caso contrario H^c contendría a K_5). Una posibilidad es el siguiente grafo.



El grafo es evidentemente planar. Para justificar que el complemento también es planar alcanza con calcularlo y dibujarlo de manera planar.

NOTA: Lo de ser engañados es porque en el primer punto, G^c siempre es no planar, independientemente de lo que ocurra con G. Aún así, puede resolverse de manera similar al Ejercicio 9.5.c de la Práctica (ver antes).

TOMADO: 2002C1R2 07-AGO-2002, 2005C1P2 15-JUL-2005.

296. (a) Sea G un bosque de 8 o más vértices. Demostrar que G^c no es planar.

(b) Exhibir un bosque de 7 vértices tal que su complemento sea planar. Justificar.

SOLUCIÓN:

(a) Sean n y m la cantidad de vértices y ejes de G, respectivamente. Por ser un bosque, se cumple $m \leq n - 1$. La cantidad de ejes de G^c es $n(n-1)/2 - m \geq (n-1)(n-2)/2$. Si fuera planar, debería ser $n(n-1)/2 - m \leq 3n - 6$, y por lo tanto $(n-1)(n-2)/2 \leq 3n - 6$, lo que no ocurre para $n \geq 8$.

(b) Si vemos el bosque como un grafo bipartito, ambas partes del conjunto de nodos deben tener menos de 5 vértices (caso contrario el complemento contendría a K_5). De tal modo, una parte debe tener 4 vértices y la otra 3. Asimismo, conviene que el bosque tenga la mayor cantidad posible de ejes (que sea un árbol) para que el complemento tenga la menor cantidad posible, y tenga más posibilidades de ser planar. Si elegimos una estrella de 3 rayos de longitud 2, cumple con lo anterior y su complemento es planar. Para justificarlo alcanza con calcular el complemento y dibujarlo de manera planar.

TOMADO: 2002C1P2 10-JUL-2002, 2005C1R2 08-AGO-2005, 2014C2P2 01-DIC-2014.

297. Un grafo planar se dice outerplanar si y sólo si existe una representación planar del mismo que tiene alguna región (usualmente la exterior) a la cual pertenecen todos los vértices.

(a) Demostrar que si G es outerplanar entonces todo subgrafo de G es outerplanar.

(b) Demostrar que si G es un bosque entonces es outerplanar.

(c) Demostrar que si G es un grafo de $n \leq 3$ vértices entonces es outerplanar.

(d) Para cada $n \geq 4$ exhibir un grafo de n vértices que sea outerplanar sin ser un bosque. Justificar.

(e) Para cada $n \geq 4$ exhibir un grafo de n vértices que no sea outerplanar. Justificar.

(f) Demostrar que si G es outerplanar no necesariamente conexo de $n \geq 2$ vértices y m ejes entonces $m \leq 2n - 3$. Demostrar que si además G es bipartito la cota puede mejorarse a $m \leq (3n-4)/2$.

SOLUCIÓN:

(a) Al remover ejes o vértices con sus ejes, las regiones sólo pueden agrandarse.

(b) Todos los vértices están en la región exterior, que es la única que existe.

(c) Por el punto anterior, basta demostrarlo para los que no son bosques. El único grafo de $n \leq 3$ vértices que no es un bosque es K_3, que es claramente outerplanar.

(d) El ciclo de $n \geq 4$ vértices C_n es claramente outerplanar.

(e) El completo de $n \geq 4$ vértices K_n no es outterplanar. Por el primer punto, basta ver que K_4 no lo es. Se puede argumentar diciendo que en una supuesta representación outerplanar habría que dibujar un ciclo con 3 de los vértices, y el cuarto no podría ubicarse ni en la región interior ni en la exterior de manera outerplanar. Otra forma para justificarlo es usar el último punto, ya que si K_n fuera outerplanar debería ser $n(n-1)/2 \leq 2n-3$, que no se cumple para $n = 4$. De hecho se cumple sólo para $n \in [2, 3]$, lo que permite argumentar que K_n no es outerplarnar para $n \geq 4$, sin usar el primer punto.

(f) La demo sigue las mismas ideas usadas para demostrar las propiedades similares en grafos planares comunes. Si G es un bosque, vale $m \leq n-1$, y para $n \geq 2$ se cumple $n - 1 \leq (3n-4)/2 \leq 2n-3$. Si G no es un bosque, es como sigue. Sea r la cantidad de regiones en toda representación outerplanar de G (o en una cualquiera, si no queremos argumentar unicidad de la cantidad de regiones). La región exterior tiene al menos n ejes para poder alojar a los n vértices, y como hay ciclos cualquier otra región tiene al menos $k = 3$ ejes, o al menos $k = 4$ si el grafo es bipartito. Como dos veces la cantidad de ejes es la suma de todos los bordes, tenemos $2m \geq n + k(r-1)$ lo que equivale a $r \leq (2m - n + k)/k$. Por Euler sabemos que $n - m + r = 1 + c$ lo que equivale a $m - n + 1 + c = r$ donde c es la cantidad de componentes conexas de G. Si remplazamos r utilizando la desigualdad anterior obtenemos $m - n + 1 + c \leq (2m - n + k)/k$ lo que equivale a $m \leq (kn - n - kc)/(k-2)$, y como $c \geq 1$ resulta $m \leq (kn - n - k)/(k-2)$, que es la desigualdad buscada para cada uno de los valores de k.

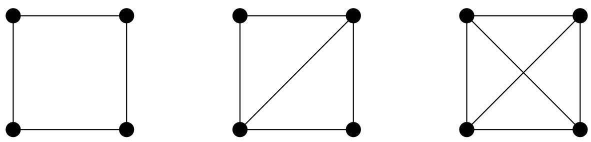
NOTA: Propuesto originalmente por Leandro Montero (segunda parte del último punto). En el anteúltimo punto podría pedirse que el grafo fuera planar. En tal caso, aparentemente una rueda de $n-1$ rayos serviría. La justificación parece que sale con el último punto.

TOMADO: 2009C1P2 10-AGO-2009.

298. Un grafo planar se dice outerplanar si y sólo si existe una representación planar del mismo que tiene alguna región (usualmente la exterior) a la cual pertenecen todos los vértices.

Un grafo se dice outerplanar maximal si y sólo si es outerplanar pero deja de serlo si se le agrega cualquier eje entre vértices no adyacentes.

En la siguiente figura aparecen, de izquierda a derecha, un grafo outerplanar no maximal (C_4), un grafo outerplanar maximal (diamante), y un grafo planar no outerplanar (K_4).



Sea G un grafo outerplanar maximal de $n \geq 3$ vértices.

(a) Demostrar que G es conexo.

(b) Demostrar que G tiene al menos un ciclo simple.

(c) Considerar una representación planar de G que tenga todos los vértices en la región exterior. Demostrar que hay otras regiones, y que el borde de cada una de ellas es un ciclo simple de 3 ejes.

(d) Demostrar que en toda representación planar de G hay $r = n - 1$ regiones y que G tiene $m = 2n - 3 = 2r - 1$ ejes.

SOLUCIÓN:

(a) Informal. Si no lo fuera, podríamos aumentar la cantidad de ejes uniendo dos componentes conexas, sin perder outerplanaridad.

(b) Informal. Si no los tuviera sería un árbol, con una única región (la exterior). Como $n \geq 3$ entonces G no sería completo, y al agregarle un eje se formaría un único ciclo, afuera del cual podríamos dibujar todo el resto del grafo, que seguiría siendo outerplanar.

(c) Informal. Como tiene ciclos, el borde de toda región es un ciclo, y hay por lo menos 2 regiones. Si alguna región no exterior no fuera un triángulo, podríamos agregar ejes en ella sin perder outerplanaridad.

(d) Basta ver que vale para cualquier representación que tenga todos los vértices en la región exterior, ya que por Euler cualquier representación planar (al menos de un grafo conexo) tiene la misma cantidad de regiones. Como $2m$ es la suma de todos los bordes, por el punto anterior tenemos $2m = n + 3(r - 1)$. Además, por Euler sabemos que $n - m + r = 2$. Despejando n, m o r en la primera ecuación y remplazando en la segunda se obtienen las igualdades buscadas.

NOTA: Propuesto originalmente por Leandro Montero.

TOMADO: 2009C1R2 26-AGO-2009 ($r = n - 1$ en el cuarto punto).

299. Un grafo se dice planar maximal si y sólo si es simple y planar pero deja de serlo si se le agrega cualquier eje entre vértices existentes.

(a) Demostrar que si G es planar maximal entonces es conexo.

(b) Exhibir todos los grafos no isomorfos de $n \leq 5$ vértices que son planares maximales. Justificar.

SOLUCIÓN:

(a) Informal. Si no lo fuera, podríamos aumentar la cantidad de ejes uniendo dos componentes conexas, sin perder planaridad.

(b) K_n con $n \leq 4$ es planar y completo, de modo que es maximal; no hay otros con la misma cantidad de nodos porque cualquier subgrafo generador propio de un grafo planar, no puede ser maximal. K_5 no es planar pero $K_5 - e$ lo es para cualquier eje e, de modo que es maximal; no hay otros por el mismo motivo que antes.

TOMADO: 2013C2R2 21-DIC-2013.

300. Sea G un grafo planar de $n \geq 3$ vértices y $m \geq 3n - 7$ ejes. Demostrar que G es conexo.

SOLUCIÓN: Por absurdo supongamos que G no es conexo. Sea G_1 una componente conexa de G, y G_2 el resto de G. Consideremos una representación planar de ambos subgrafos. Al menos una de las partes tiene al menos dos nodos v_1 y v_2 en la región exterior, y la otra al menos un nodo w. Podemos agregar ejes de w a v_1 y a v_2 manteniendo la planaridad, lo cual es absurdo porque el grafo tendría al menos $3n - 5$ ejes, a pesar de que el máximo para un grafo planar de $n \geq 3$ vértices es $3n - 6$.

Otra forma de demostrarlo es como sigue. Por Euler $n - m + r = 1 + c$, donde c es la cantidad de componentes conexas del grafo. Al ser $n \geq 3$ el borde de cada región tiene tamaño al menos 3, y por lo tanto $2m \geq 3r$, es decir $r \leq 2m/3$. Como además $m \geq 3n - 7$ tenemos $c = n - m + r - 1 \leq n - m + 2m/3 - 1 = n - m/3 - 1 \leq n - (3n - 7)/3 - 1 = n - n + 7/3 - 1 = 4/3$, es decir $c \leq 4/3$, lo que equivale a $c = 1$ porque es entero no nulo.

NOTA: Propuesto originalmente por Federico Lebrón. Aparentemente está relacionado y se puede probar de manera similar que todo grafo planar maximal de $n \geq 3$ vértices no tiene puentes.

TOMADO: 2014C1P2 07-JUL-2014.

Coloreo

301.

Variante 1: Sean G un grafo y $k \in \mathbb{N}$.

(a) Demostrar que el polinomio cromático de G evaluado en k vale 1 si y sólo si G no tiene ejes y $k = 1$.

(b) Demostrar que el polinomio cromático de G evaluado en k vale 2 si y sólo si G es trivial o bipartito conexo, y $k = 2$.

Variante 2: (a) Encontrar todos los pares (G, k) con G grafo y $k \in \mathbb{N}$ tales que el polinomio cromático de G evaluado en k vale 1. Justificar.

(b) Repetir para 2.

SOLUCIÓN:

(a) Para la ida, por absurdo supongamos que G tiene al menos un eje e y consideremos un coloreo de G. Los extremos de e deben tener dos colores dintintos. Podemos intercambiar esos colores en el coloreo de G, obteniendo un coloreo distinto al original, lo cual es absurdo porque $P_G(k) = 1$. En consecuencia, G no tiene ejes, lo cual implica que su polinomio cromático es $P_G(k) = k^n$, que sólo vale 1 si $k = 1$.

Alternativamente para la ida, por absurdo supongamos $k \geq 2$ y consideremos un coloreo de G. Sea c un color usado en el coloreo, y sea c' cualquier otro color disponible (usado o no). Podemos intercambiar esos colores en el coloreo de G, obteniendo un coloreo distinto al original, lo cual es absurdo porque $P_G(k) = 1$. En consecuencia, $k \leq 1$, pero no puede ser $k = 0$ porque $P_G(k) = 1$, de modo que se cumple $k = 1$. Al ser $P_G(k) = 1$ significa que G admite ser coloreado con un único color, por lo que no puede tener ejes.

Para la vuelta, sabemos que $P_G(k) = k^n$, que evaluado en $k = 1$ vale 1.

(b) Para la ida, consideremos los 2 coloreos de G que usan a lo sumo k colores. Ninguno de esos coloreos puede usar 3 o más colores porque entonces se podrían permutar los colores y habría más de 2 coloreos. Por lo tanto los coloreos usan a lo sumo 2 colores, lo que implica que G es trivial o bipartito. Si G es trivial sabemos que $P_G(k) = k$, de modo que $k = 2$. Si G es bipartito no puede ser $k = 1$ porque $P_G(1)$ vale 0 o 1 para cualquier G, y tampoco puede ser $k \geq 3$ porque se podrían elegir 2 de esos 3 colores para colorear G de 3 formas diferentes. Es así que en este caso también tenemos $k = 2$. Notemos además que si G es bipartito debe ser conexo, ya que caso contrario habría al menos dos formas de colorear cada componente conexa si hay $k = 2$ colores disponibles, y al combinar las posibilidades habría más de 2 formas de colorear G.

Para la vuelta, si G es trivial sabemos que $P_G(k) = k$, que evaluado en $k = 2$ vale 2. El otro caso es cuando G es bipartito conexo y hay $k = 2$ colores disponibles. Sea v un nodo cualquiera de G. Si asignamos uno de los colores a v, sus adyacentes deben recibir el otro, y así sucesivamente hasta que todos los nodos son coloreados porque el grafo es conexo. El coloreo queda unívocamente determinado por el color asignado a v. La alternativa es asignar el otro color disponible a v, lo que también determina el resto del coloreo. En total hay dos posibles coloreos, lo que equivale a decir que el polinomio cromático de G evaluado en k vale 2.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma para 1, y por Min Chih Lin para 2.

TOMADO: 2018C2P2 03-DIC-2018 (Variante 1, primer punto), 2018C2R2 19-DIC-2018 (Variante 1, segundo punto).

302. Sea G un grafo de n vértices, m ejes, y formado por c componentes conexas.

(a) Demostrar que si G es un árbol entonces su polinomio cromático es $P_G(k) = k(k-1)^{n-1}$.

(b) Demostrar que si G es un bosque entonces su polinomio cromático es $P_G(k) = k^c(k-1)^{n-c} = k^{n-m}(k-1)^m = k^c(k-1)^m$.

SOLUCIÓN:

(a) Se demuestra por inducción sacándole a T una de sus hojas en el paso inductivo.

(b) Las tres expresiones son equivalentes ya que en un bosque $m = n - c$. El polinomio cromático de un grafo no conexo es la productoria de los polinomios cromáticos de cada una de las componentes conexas, que en este caso son árboles. Si llamamos G_i a la i-ésima componente conexa, y n_i a su cantidad de nodos, por el primer punto tenemos $P_G(k) = \Pi_{i=1}^{c} P_{G_i}(k) = \Pi_{i=1}^{c} k(k-1)^{n_i - 1} = k^c(k-1)^{\sum_{i=1}^{c} n_i - 1} = k^c(k-1)^{n-c}$.

NOTA: La recíproca del primer punto también es cierta.

TOMADO: 2002C1R2 07-AGO-2002 (primer punto).

303. Se puede demostrar que si dos grafos son isomorfos entonces tienen el mismo polinomio cromático. ¿Vale la recíproca? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN: No vale. Según el Ejercicio 302 cualquier árbol de n vértices tiene el mismo polinomio cromático, de modo que es suficiente tomar dos árboles no isomorfos de la misma cantidad de vértices. Una posibilidad es tomar P_4 y $K_{1,3}$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2017C2R2 18-DIC-2017.

304. Sea C_n el grafo que es un ciclo simple de $n \geq 3$ vértices, y sea $W_n = C_n + K_1$ (grafo junta). Es sabido que el polinomio cromático de C_n es $(k-1)^n + (-1)^n(k-1)$. Demostrar que el polinomio cromático de W_n es $k[(k-2)^n + (-1)^n(k-2)]$.

SOLUCIÓN: Hay k posibilidades para pintar el nodo universal de W_n que proviene de K_1. Los otros nodos forman C_n y hay $k-1$ colores disponibles para pintarlos, ya que todos son adyacentes al nodo proveniente de K_1. La cantidad de formas de pintar los nodos de C_n con $k-1$ colores es por definición el polinomio cromático de C_n evaluado en $k-1$, es decir, $(k-2)^n + (-1)^n(k-2)$. Cada una de esas formas, combinada con las k formas de pintar el primer nodo, define una forma posible de pintar W_n, de modo que la cantidad de formas de pintar W_n es el producto entre k y $(k-2)^n + (-1)^n(k-2)$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

305. Sean $x, x' \in \mathbb{N}$.

Variante 1: (a) Determinar la *mínima* cantidad de vértices que puede tener un grafo G tal que $\chi(G) \geq x$ y $\chi(G^c) \geq x'$. Expresar el resultado en función de x y x'.
SUGERENCIA: Demostrar una cota *inferior* para la cantidad de vértices y exhibir un grafo en el cual se alcanza esa cota.

(b) Determinar la *máxima* cantidad de vértices que puede tener un grafo G tal que $\chi(G) \leq x$ y $\chi(G^c) \leq x'$. Expresar el resultado en función de x y x'.
SUGERENCIA: Demostrar una cota *superior* para la cantidad de vértices y exhibir un grafo en el cual se alcanza esa cota.

(c) Exhibir todos los grafos G no isomorfos tales que $\chi(G) \leq 2$ y $\chi(G^c) \leq 2$. Justificar.

Variante 2: (a) Demostrar que si G es un grafo de n vértices tal que $\chi(G) \geq x$ y $\chi(G^c) \geq x'$, entonces $n \geq x + x' - 1$. Demostrar que la cota para n no puede ser mejorada exhibiendo un grafo G de n vértices y valores x, x' tales que $\chi(G) \geq x$, $\chi(G^c) \geq x'$, y $n = x + x' - 1$. Justificar.

(b) Demostrar que si G es un grafo de n vértices tal que $\chi(G) \leq x$ y $\chi(G^c) \leq x'$, entonces $n \leq x \times x'$. Demostrar que la cota para n no puede ser mejorada exhibiendo un grafo G de n vértices y valores x, x' tales que $\chi(G) \leq x$, $\chi(G^c) \leq x'$, y $n = x \times x'$. Justificar.

(c) Exhibir todos los grafos G no isomorfos tales que $\chi(G) \leq 2$ y $\chi(G^c) \leq 2$. Justificar.

SOLUCIÓN: Las dos variantes en definitiva piden lo mismo.

(a) Por el Ejercicio 10.12.a de la Práctica $n+1 \geq \chi(G) + \chi(G^c) \geq x + x'$, de modo que $n \geq x + x' - 1$. El Ejercicio 10.12.b provee otra cota ya que según el mismo $((n+1)/2)^2 \geq \chi(G) \times \chi(G^c) \geq x \times x'$, de modo que $n \geq 2\sqrt{x \times x'} - 1$; sin embargo, esta cota es peor (menor) o igual ya que la media aritmética es mayor o igual que la media geométrica (excepto para $x = x'$ en cuyo caso coinciden).

Para cada par $x, x' \in \mathbb{N}$ se pueden construir grafos que alcanzan la cota. Un grafo posible es $G = K_x \cup (x'-1)K_1$. Es claro que G tiene $x + x' - 1$ vértices. Además, se cumple $\chi(G) \geq x$ porque K_x es subgrafo de G, y se cumple $\chi(G^c) \geq x'$ porque G tiene un conjunto independiente de x' vértices. Otro grafo posible es el complemento de $G^c = K_{x'} \cup (x-1)K_1$, que es la misma idea de recién pero pensada desde el lado del complemento.

(b) Por el Ejercicio 10.12.b de la Práctica $n \leq \chi(G) \times \chi(G^c) \leq x \times x'$, de modo que $n \leq x \times x'$. El Ejercicio 10.12.c provee otra cota ya que según el mismo $2\sqrt{n} \leq \chi(G) + \chi(G^c) \leq x + x'$, de modo que $n \leq ((x + x')/2)^2$; sin embargo, esta cota es peor (mayor) o igual ya que la media aritmética es mayor o igual que la media geométrica (excepto para $x = x'$ en cuyo caso coinciden).

Para cada par $x, x' \in \mathbb{N}$ se pueden construir grafos que alcanzan la cota. Un grafo posible es $G = x'K_x$. Es claro que G tiene $x \times x'$ vértices. Además, se cumple $\chi(G) \leq x$ porque cada K_x puede colorearse de manera independiente, y se cumple $\chi(G^c) \leq x'$ porque los nodos que en G forman cada K_x, en G^c pueden colorearse del mismo color. Otro grafo posible es el complemento de $G^c = xK_{x'}$, que es la misma idea de recién pero pensada desde el lado del complemento.

(c) Por el punto anterior es $n \leq 4$, y al analizar casos resulta que G puede ser K_1, $2K_1$, K_2, $K_1 \cup K_2$, P_3, $2K_2$, P_4 o C_4. Alternativamente, se puede hacer lo mismo que en el Ejercicio 91c con el agregado del grafo trivial, o una combinación de ambas estrategias.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez (excepto el primer punto).

TOMADO: 2019C1P2 05-JUL-2019 (Variante 2).

306. Para cada uno de los siguientes grafos determinar el número cromático de su complemento. Expresar los resultados en función de n, p y q. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(e) $K_{p,q}$

SUGERENCIA: Demostrar que si G es un grafo de n vértices y $\alpha(G)$ es la cantidad de vértices de un conjunto independiente máximo de G, entonces $\chi(G) \geq \lceil n/\alpha(G) \rceil$. Usarlo al menos para $G = C_n^c$ con $n \geq 5$ impar.

SOLUCIÓN: Si partimos un grafo en partes sin ejes entre ellas (por ejemplo, componentes conexas), el número cromático del grafo es el máximo de los números cromáticos de cada parte.

K_n: El complemento tiene n componentes conexas iguales a K_1. Por lo tanto tenemos $\chi(K_n^c) = \chi(K_1) = 1$.

C_n: Para $n = 3$ es $C_3 = K_3$. Por el primer punto tenemos $\chi(C_3^c) = \chi(K_3^c) = 1$.

Para $n \geq 4$ par vamos a demostrar que $\chi(C_n^c) = \lceil n/2 \rceil = n/2$ probando la doble desigualdad. Nodos alternados del ciclo de C_n forman un conjunto independiente de $n/2$ elementos, los cuales en el complemento inducen un completo, de modo que $\chi(C_n^c) \geq n/2$ (esta desigualdad también se puede probar con la sugerencia o con el Ejercicio 10.12.b de la Práctica). Además nodos consecutivos en el ciclo de C_n no son adyacentes en el complemento. Eso implica que si aplicamos el algoritmo secuencial de coloreo al complemento siguiendo el orden del ciclo, el algoritmo va a usar al menos dos veces cada color, es decir, a lo sumo $n/2$ colores en total. Como un coloreo óptimo es mejor o igual que uno arbitrario, resulta $\chi(C_n^c) \leq n/2$.

Para $n \geq 5$ impar vamos a demostrar que $\chi(C_n^c) = \lceil n/2 \rceil = (n+1)/2$ probando la doble desigualdad. En el Ejercicio 327 se demuestra que $n \leq \chi(G)\alpha(G)$, lo que equivale a la sugerencia ya que $\chi(G)$ es entero. Como $\alpha(G) = \omega(G^c)$, es $\alpha(C_n^c) = \omega(C_n) = 2$, y de acuerdo a la sugerencia tenemos $\chi(C_n^c) \geq \lceil n/2 \rceil$. De manera similar al caso par, el algoritmo secuencia para n impar usa a lo sumo $\lceil n/2 \rceil$ colores, por lo que $\chi(C_n^c) \leq \lceil n/2 \rceil$.

P_n: Vamos a demostrar que $\chi(P_n^c) = \lceil n/2 \rceil$ probando la doble desigualdad. Comenzando desde un extremo de P_n, nodos alternados forman un conjunto independiente de $\lceil n/2 \rceil$ elementos, los cuales en el complemento inducen un completo, de modo que $\chi(P_n^c) \geq \lceil n/2 \rceil$ (para $n \geq 2$ esta desigualdad también se puede probar con la sugerencia o con el Ejercicio 10.12.b de la Práctica). El algoritmo secuencial aplicado a P_n^c siguiendo el orden del camino de P_n, usa al menos dos veces cada color. Como vimos antes eso implica $\chi(P_n^c) \leq \lceil n/2 \rceil$.

W_n: El complemento tiene dos partes: K_1 y C_n^c. Por lo tanto tenemos $\chi(W_n^c) = \max\{\chi(K_1), \chi(C_n^c)\} = \chi(C_n^c)$, que por el segundo punto es 1 si $n = 3$, o $\lceil n/2 \rceil$ si $n \geq 4$.

$K_{p,q}$: El complemento tiene dos componentes conexas: K_p y K_q. Por lo tanto tenemos $\chi(K_{p,q}^c) = \max\{\chi(K_p), \chi(K_q)\} = \max\{p, q\}$.

Los resultados mostrados permiten determinar para qué valores de n se cumple que C_n^c es color crítico. Se tiene $C_n^c - v = (C_n - v)^c = P_{n-1}^c$, de modo que $\chi(C_n^c - v) = \chi(P_{n-1}^c) = \lceil (n-1)/2 \rceil$, es decir, $(n-1)/2$ si n es impar, y $n/2$ si n es par. Por otro lado $\chi(C_n^c)$ vale 1 si $n = 3$, $n/2$ si $n \geq 4$ par, y $(n+1)/2$ si $n \geq 5$ impar. Por lo tanto, C_n^c es color crítico si y sólo si $n \geq 5$ impar, ya que en esos casos disminuye el número cromático al eliminar cualquier vértice.

NOTA: Basado en una pregunta de un alumno sobre $\chi(C_n^c)$. No se incluye A n (variable), GBC n m (variable), ni ABC h (difícil).

TOMADO: 2015C2P2 30-NOV-2015.

307. Sea C_n el grafo que es un ciclo simple de $n \geq 3$ vértices.

(a) Demostrar que $\chi(C_3^c) = 1$ y $\chi(C_n^c) = \lceil n/2 \rceil$ para $n \geq 4$.

(b) Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Sea G un grafo de n vértices, d-regular y tal que $\chi(G) + \chi(G^c) = n + 1$. Demostrar que $G = K_n$ o $G^c = K_n$ o $G = G^c = C_5$.

SUGERENCIA: Usar o no el primer punto.

SOLUCIÓN:

(a) Demostrado en el Ejercicio 306.

(b) Si $G = K_n$ o $G^c = K_n$ no hay nada más que probar.

Si $G \neq K_n \neq G^c$, por el Ejercicio 10.13 de la Práctica, G es un ciclo impar. No puede ser $G = C_3$ porque entonces sería $G = K_3$ que estamos suponiendo que no ocurre. No puede ser $G = C_n$ con $n \geq 7$ porque entonces por el primer punto sería $n + 1 = \chi(G) + \chi(G^c) = 3 + \lceil n/2 \rceil = 3 + (n+1)/2$, y la igualdad $n + 1 = 3 + (n+1)/2$ no se cumple para $n \geq 7$. Por lo tanto es $G = C_5$, cuyo complemento es también C_5.

Alternativamente, si $G \neq K_n \neq G^c$, por el Ejercicio 10.13 de la Práctica, G es un ciclo impar. El ejercicio también se puede aplicar a G^c, lo que permite deducir que también G^c es un ciclo impar. Por lo tanto, ambos son 2-regulares, y como la suma de los grados en G y en G^c es $n - 1$, tenemos que $2 + 2 = n - 1$ o equivalentemente $n = 5$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2016C2R2 19-DIC-2016.

308. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

Sean G y H dos grafos.

(a) Demostrar que si G es homomorfo a H entonces $\chi(G) \leq \chi(H)$.

(b) Demostrar que G es k-coloreable si y sólo si G es homomorfo a K_k.

(c) Demostrar que $\chi(G)$ es el mínimo $k \in \mathbb{N}$ tal que G es homomorfo a K_k.

SOLUCIÓN:

(a) Sea f el homomorfismo de G a H. Consideremos un coloreo óptimo para H. A cada nodo v de G, le asignamos el color que tiene $f(v)$. Si dos nodos son adyacentes en G, como sus imágenes lo son en H, van a recibir distinto color. Por lo tanto el coloreo de G es válido, y tiene a lo sumo $\chi(H)$ colores.

(b) Para la vuelta, como $\chi(K_k) = k$, por el punto anterior resulta $\chi(G) \leq k$, es decir, G es k-coloreable. Para la ida, consideremos un coloreo de G que usa a lo sumo k colores; definimos la función de $V(G)$ en $V(K_k)$ asignando a cada nodo su color (es decir, cada nodo de K_k representa un color entre 1 y k); veamos que esa función define un homomorfismo; sea $(v, w) \in E(G)$; como deben tener colores diferentes, $f(v)$ y $f(w)$ son distintos, por lo que esos nodos son adyacentes en K_k que es completo.

(c) Por el punto anterior, equivale a probar que $\chi(G)$ es el mínimo k tal que G es k-coloreable. Pero esa es justamente la definición de $\chi(G)$.

NOTA: El espíritu es el mismo que en el Ejercicio 342.

TOMADO: 2011C2R2 21-DIC-2011.

309. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

Sea G un grafo. Demostrar que G es 2-coloreable si y sólo si es homomorfo a K_2.

SOLUCIÓN: Es un caso particular del segundo punto del Ejercicio 308. Sigue una solución específica.

Para la ida, consideremos un 2-coloreo de G; sea V_1 el conjunto de nodos que tienen un color, y V_2 (eventualmente vacío) el conjunto de nodos que tienen el otro; definamos una función que manda todos los nodos de V_1 a uno de los nodos de K_2, y todos los nodos de V_2 al otro; como no hay ejes dentro de V_1 ni dentro de V_2, la función resulta un homomorfismo. Para la vuelta, sea V_1 la preimagen de uno de los nodos de K_2, y V_2 la preimagen del otro; no puede haber ejes dentro de V_1 ni dentro de V_2 porque entonces la función no sería un homorfismo; por lo tanto, es posible colorear a V_1 con un color y a V_2 con otro.

NOTA: Es esencialmente el Ejercicio 64.

TOMADO: 2017C1P2 07-JUL-2017.

310. Sea G un grafo conexo de n vértices y m ejes. Demostrar que $\chi(G) \leq m - n + 3$.

SUGERENCIA: Usar un árbol generador y que $m - n + 3 = 2 + [m - (n - 1)]$.

SOLUCIÓN: Consideremos un árbol generador T de G, que tiene $n-1$ ejes y puede pintarse con 2 colores ya que no tiene ciclos impares. Para conseguir un coloreo válido de G, agregamos cada uno de los $m - (n - 1)$ ejes que no están en T, y al hacerlo cambiamos el color de uno de sus extremos por un color completamente nuevo.

NOTA: Propuesto originalmente por Federico Lebrón.

TOMADO: 2012C2R2 19-DIC-2012.

311. Sea G un grafo. Demostrar que los ejes de G pueden orientarse de manera tal de obtener un digrafo que cumpla todo lo siguiente.

(a) No tenga ciclos dirigidos.

(b) Tenga todos sus caminos dirigidos formados por a lo sumo $\chi(G) - 1$ arcos.

(c)

Variante 1: Tenga al menos un camino dirigido formado por exactamente $\chi(G) - 1$ arcos.
SUGERENCIA: Organizar los vértices en $\chi(G)$ niveles.

Variante 2: Para cada $i = 1, 2, \ldots \chi(G) - 1$, tenga al menos un camino dirigido maximal formados por exactamente i arcos.
SUGERENCIA: Algoritmo secuencial de coloreo.

SOLUCIÓN: Considerar un coloreo óptimo de G, y poner en el i-ésimo nivel los nodos que tienen el color i. Dirigir los arcos de cada nivel hacia el siguiente. Notar que no hay arcos de un nivel a sí mismo, porque cada nivel forma un conjunto independiente. El digrafo resultante es claramente acíclico, con todos sus caminos de largo a lo sumo $\chi(G) - 1$. Para demostrar que existe un camino de ese largo podemos considerar un coloreo óptimo particular en vez de cualquiera. Por ejemplo, podemos tomar un coloreo óptimo producido por el algoritmo secuencial de coloreo (por el Ejercicio 10.14.d de la Práctica, existe un orden de los nodos que lo produce). Si orientamos los ejes como antes, hay un camino de longitud $X(G) - 1$ porque cada nodo es adyacente al menos a un nodo del nivel anterior (caso contrario, el algoritmo secuencial le hubiera asignado un color menor). Para conseguir el camino es suficiente tomar un nodo del último nivel y retroceder repetidamente a un predecesor. Más aún, hay caminos dirigidos maximales de largos $1, 2, \ldots \chi(G) - 1$, porque cada nodo es adyacente al menos a un nodo de cada nivel menor al suyo. Otra posibilidad es tomar cualquier coloreo óptimo y luego corregirlo como sigue. Para cada clase de color tratamos de mover cada nodo a la clase anterior, a la cual lo movemos siempre que no sea adyacente a ningún nodo de la clase destino. Al terminar sabemos que cada clase forma un conjunto independiente (porque lo eran al principio y al mover un nodo no violamos esa propiedad), hay al menos un nodo en cada clase (caso contrario el coloreo no era óptimo), y todo nodo que originalmente estaba en una clase es adyacente al menos a un nodo que originalmente estaba en la clase anterior (ya que no puede ser adyacente a los nodos que originalmente estaban en su propia clase, y es adyacente al menos a un nodo de la clase anterior). Por lo tanto, finalmente se puede armar el camino de largo máximo igual que antes. Alternativamente, se puede mover cada nodo al mínimo nivel posible menor que el propio, lo que produce que al finalizar se cumpla que cada nodo de un nivel es adyacente al menos a un nodo de cada uno de los niveles anteriores, lo que produce caminos maximales de todos los largos posibles, como en el caso del algoritmo secuencial de coloreo.

NOTA: Propuesto originalmente por Michel Mizrahi (Variante 1).

TOMADO: 2012C1P2 11-JUL-2012 (Variante 1), 2012C1R2 08-AGO-2012 (Variante 2).

312. Sea G un grafo no trivial y v uno de sus vértices. Demostrar que si $\chi(G - v) < \chi(G)$ entonces $d(v) \geq \chi(G) - 1$.

SOLUCIÓN: Si fuera $d(v) \leq \chi(G) - 2$ podría colorearse G con menos de $\chi(G)$ colores, tomando un coloreo óptimo de $G - v$ y pintando v con un color no usado por sus adyacentes.

NOTA: Es esencialmente el Ejercicio 10.7.b de la Práctica, salvo que la precondición es para un vértice particular.

TOMADO: 2015C2R2 18-DIC-2015.

313.

Variante 1: Sea G un grafo no trivial, conteniendo un vértice universal v. Demostrar que G es color crítico si y sólo si $G - v$ es color crítico o trivial.

Variante 2: Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, se define su grafo junta como $G_1 + G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$, es decir, el grafo que contiene a G_1 y a G_2 como subgrafos, y además contiene un eje entre cada vértice de G_1 y cada vértice de G_2. Se dice que un grafo J es un grafo junta si y sólo si existen grafos G_1 y G_2 tales que $J = G_1 + G_2$.

Sean G y H dos grafos. Demostrar que $G + H$ es color crítico si y sólo si G es color crítico o trivial, y H es color crítico o trivial.

SOLUCIÓN: La Variante 1 es un caso particular de la Variante 2 tomando $G' = G - v$ y $H' = (\{v\}, \emptyset)$, de modo que basta demostrar la Variante 2.

Sabemos que un grafo es color crítico si y sólo si al sacarle cualquier nodo su número cromático disminuye (debe disminuir exactamente en 1). Por el Ejercicio 10.8 de la Práctica, $\chi(G + H) = \chi(G) + \chi(H)$, y si G y H son color críticos, entonces $G + H$ también lo es. Notar además que si G es no trivial y v es uno de sus nodos entonces $(G - v) + H = (G + H) - v$, y lo mismo para H.

Para la ida, basta demostrar que G cumple lo pedido, ya que G y H son intercambiables. Si G es trivial no hay nada más que probar. Si G no es trivial, sea v cualquier nodo de G. Usando las propiedades mencionadas tenemos que $\chi(G-v) = \chi(G-v)+\chi(H)-\chi(H) = \chi((G-v)+H)-\chi(H) = \chi((G + H) - v) - \chi(H) = \chi(G + H) - 1 - \chi(H) = \chi(G) + \chi(H) - 1 - \chi(H) = \chi(G) - 1$, por lo que G es color crítico.

Para la vuelta, por el Ejercicio 10.8 de la Práctica la implicación se cumple si G y H son color críticos, y en cualquier caso sabemos que $\chi(G+H) = \chi(G) + \chi(H)$. Si tanto G como H son triviales, entonces $G + H = K_2$ que es color crítico. Resta analizar el caso es que G es trivial y H es color crítico (no trivial), ya que G y H son intercambiables. Sea v cualquier nodo de $G+H$. Si v está en G, como es su único nodo resulta $(G+H)-v = H$, y por lo tanto $\chi((G+H)-v) = \chi(H) = \chi(H) + \chi(G) - \chi(G) = \chi(G+H) - 1$. Si v está en H que es color crítico, resulta $\chi((G+H)-v) = \chi(G+(H-v)) = \chi(G) + \chi(H-v) = \chi(G) + \chi(H) - 1 = \chi(G+H) - 1$. Como al sacar cualquier nodo de $G+H$ el número cromático disminuye, $G+H$ es color crítico.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2011C2P2 05-DIC-2011 (Variante 2).

314. Sea G un grafo. Demostrar que G es color crítico y 3-cromático si y sólo si es un ciclo impar (ciclo simple de $n \geq 3$ vértices con n impar).

SOLUCIÓN: Para la ida, G debe tener un ciclo impar ya que si no lo tuviera sería 2-coloreable. Por lo tanto, tiene un ciclo simple impar, y sea C cualquiera de ellos. Si hay nodos fuera de C, al sacar cualquiera de ellos subsistiría C, por lo que G no sería color crítico. Si C tiene alguna cuerda, uno de los dos circuitos que forma la cuerda con C tiene que ser impar, y al sacar un vértice del otro circuito no disminuiría el número cromático, por lo que G no sería color crítico (alternativamente, elegir como C a un circuito simple impar de longitud mínima).

Para la vuelta, sabemos que un ciclo impar es 3-cromático. Además, al sacarle cualquier vértice queda un camino simple que es 2-coloreable, de modo que el ciclo es color crítico.

NOTA: Propuesto originalmente por Federico Lebrón. Es parecido al Ejercicio 62. Está relacionado con el Ejercicio 48.

TOMADO: 2012C2P2 05-DIC-2012.

315. Sea G un grafo de 3 o más vértices. Sea v un vértice de G tal que $\chi(G-v) < \chi(G)$.

(a) ¿Es cierto que si $G-v$ es color crítico entonces G también lo es?

(b) ¿Es cierto que si G es color crítico entonces $G-v$ también lo es?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SUGERENCIA: Cierto y falso.

SOLUCIÓN:

(a) Es cierto. Sea w cualquier vértice de G. Basta ver que $\chi(G-w) < \chi(G)$. La idea es ver que si sacamos v ya sabemos que el número cromático disminuye, mientras que si sacamos otro vértice podemos sacarlo de $G-v$ (que es color crítico) y agregar v con un color adicional. Si $w = v$ entonces $\chi(G-w) = \chi(G-v) < \chi(G)$. Si $w \neq v$, como $G-v$ es color crítico tenemos $\chi(G-w) \leq \chi(G-w-v) + 1 = \chi(G-v-w) + 1 < \chi(G-v) + 1$ lo que implica $\chi(G-w) \leq \chi(G-v) < \chi(G)$.

(b) Es falso. $G = C_5$ es color crítico pero $G - v = C_5 - v = P_4$ no lo es. Se cumple que $\chi(G-v) < \chi(G)$, aunque no hace falta verificarlo ya que se deduce de que G es color crítico.

NOTA: Basado en una pregunta de un alumno.

TOMADO: 2013C1R2 05-AGO-2013.

316. Sabemos que si G es cualquier grafo entonces $\chi(G) \leq \Delta(G) + 1$, donde $\Delta(G)$ es el grado máximo en G. Demostrar que si G es color crítico entonces la cota puede mejorarse a $\chi(G) \leq \delta(G) + 1$, donde $\delta(G)$ es el grado mínimo en G.

SOLUCIÓN: De acuerdo al Ejercicio 10.7.b de la Práctica, todo vértice de G cumple $\chi(G) - 1 \leq d(v)$, en particular uno de grado mínimo, es decir, $\chi(G) - 1 \leq \delta(G)$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2015C1R2 17-JUL-2015.

317. Determinar para qué valores de n, m, p y q los siguientes grafos son color críticos. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) árbol de n vértices

(e) grafo bipartito conexo de n vértices y m ejes

(f) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(g) $K_{p,q}$

SOLUCIÓN:

K_n: $n \geq 2$ (para $n = 1$ al sacar un nodo no queda un grafo).

C_n: n impar.

P_n: $n = 2$.

A n: ?

GBC n m: ?

W_n: Se cumple $W_n = C_n + K_1$ (grafo junta). De acuerdo al Ejercicio 10.8 de la Práctica es entonces $\chi(W_n) = \chi(C_n) + \chi(K_1) = \chi(C_n) + 1$. De tal modo, $\chi(W_n) = 3 + 1 = 4$ para n impar, y $\chi(W_n) = 2 + 1 = 3$ para n par. Si n es impar al eliminar el vértice universal queda el ciclo simple que es 3-coloreable; si se elimina un vértice del ciclo queda un camino simple con un vértice universal agregado, que también es 3-coloreable; en consecuencia es color crítico. Si n es par al eliminar cualquier vértice del ciclo siempre queda al menos un triángulo, por lo que el número cromático no disminuye, de modo que no es color crítico.

$K_{p,q}$: $p = q = 1$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. No se incluye ABC h (no aporta).

TOMADO: 2013C2P2 02-DIC-2013 (W_n), 2016C2P2 30-NOV-2016 (excepto W_n).

318. Sea $G = (V, E)$ un grafo. Como es usual, una partición de V es un conjunto $\{V_1, V_2, \ldots V_k\}$ tal que

- $\emptyset \neq V_i \subseteq V$ para $1 \leq i \leq k$;
- $V_i \cap V_j = \emptyset$ para $1 \leq i, j \leq k$ con $i \neq j$; y
- $\cup_{i=1}^{k} V_i = V$.

Notar que dado que una partición es un conjunto, no importa el orden en que aparecen los distintos subconjuntos dentro de la misma; lo único relevante es el contenido de cada subconjunto. Sabemos que cualquier coloreo de V que usa k colores determina una partición de V en la cual cada subconjunto V_i contiene a los vértices que recibieron un mismo color. Se dice que G es únicamente coloreable si y sólo si todo coloreo óptimo de V (es decir, que usa $\chi(G)$ colores) determina la misma partición de V. Por ejemplo, el grafo completo de n vértices K_n es únicamente coloreable, mientras que el ciclo simple de 5 vértices C_5 no lo es.

Sea $G = (V, E)$ un grafo únicamente coloreable.

(a) Demostrar que $d(v) \geq \chi(G) - 1$ para todo vértice v de G.

(b) Sea $\{V_1, V_2, \ldots V_{\chi(G)}\}$ la única partición determinada por cualquier coloreo óptimo de V. Demostrar que si $i \neq j$ entonces el subgrafo inducido por $V_i \cup V_j$ es conexo.

(c) Demostrar que si $\chi(G) \geq 2$ entonces G es conexo.

SOLUCIÓN:

(a) Sea V_i el subconjunto al cual pertenece v. Para cada $j \neq i$, el vértice v debe ser adyacente a algún vértice de V_j, ya que caso contrario v podría ser pintado con el color j, y la partición no sería única. Como eso vale para cualquier j, resulta lo pedido.

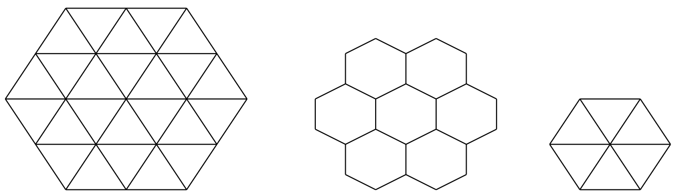
(b) Supongamos que no es conexo. Por el Ejercicio 5.6 de la Práctica el conjunto de vértices $V_i \cup V_j$ puede partirse en dos subconjuntos sin ejes entre ambas partes. Tomemos uno de esos subconjuntos, e intercambiemos los colores i y j dentro del mismo. Eso produce un coloreo válido, y una nueva partición, que entonces no era única.

(c) Sean v y w dos vértices cualesquiera, V_v y V_w sus subconjuntos. Si son distintos, el subgrafo inducido por los mismos es conexo, y por lo tanto existe un camino de v a w. Si son iguales, existe algún otro V_j, y el subgrafo inducido por $V_v = V_w$ y V_j es conexo, por lo que también existe el camino.

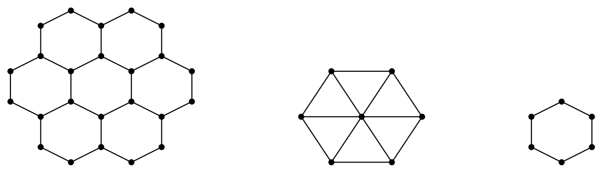
NOTA: Propuesto originalmente por Leandro Montero. Aparece en el libro de Harary.

TOMADO: 2009C1R2 26-AGO-2009.

319. (a) Determinar para cada dibujo de la figura la mínima cantidad de colores necesarios para pintar cada uno de los polígonos que lo forman con un color, de manera tal que polígonos que comparten al menos un lado reciban colores diferentes. Justificar.



(b) Determinar el número cromático de cada grafo de la figura. Justificar.



SOLUCIÓN: Los grafos modelan las adyacencias de los dibujos, de manera tal que ambos puntos son equivalentes. El primer y el tercer grafo son bipartitos con ejes, por lo que tienen número cromático 2. El segundo grafo es 3-coloreable y contiene a K_3, por lo que su número cromático es 3.

NOTA: Las figuras son las mismas que en el Ejercicio 251.

TOMADO: 2009C1P2 10-AGO-2009 (primer punto).

320. (a) Sea G_n el grafo que se obtiene eliminando de K_n cualquiera de sus ejes. Demostrar que $\chi(G_n) = n - 1$.

(b) Demostrar que K_n es el único grafo de n vértices y número cromático igual a n.

SOLUCIÓN:

(a) $\chi(G_n) \geq n - 1$ porque K_{n-1} es subgrafo de G_n. Basta entonces ver que $\chi(G_n) \leq n - 1$, para lo cual es suficiente con mostrar un coloreo que use $n - 1$ colores. Si eliminamos de G_n uno de los extremos del eje que removimos, obtenemos K_{n-1} que puede colorearse con $n - 1$ colores. Como todos los nodos reciben colores diferentes, podemos colorear el nodo que habíamos sacado con el mismo color que el otro extremo del eje. Otra forma de demostrar la cota superior es usando el teorema de Brooks, que dice que si G es conexo, no completo ni un ciclo impar, entonces $\chi(G) \leq d_{\max}(G)$; el caso $n = 2$ hay que analizarlo por separado porque no se cumplen las hipótesis del teorema; el caso $n = 1$ no interesa porque no hay ejes.

(b) Ya sabemos que $\chi(K_n) = n$. Falta ver que no hay otro grafo G de n vértices para el cual $\chi(G) = n$. Sea G de n vértices, $G \neq K_n$. Por lo tanto, G es subgrafo de K_n, y como nodos no le faltan, le faltan ejes, por lo que G es subgrafo de G_n, y utilizando el punto anterior obtenemos que $\chi(G) \leq \chi(G_n) = n - 1$.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2007C1P2 10-JUL-2007.

321. (a) Sea G un grafo no trivial. Sean v y w dos vértices de G que tienen el mismo color en algún coloreo óptimo de G. Sea G_{vw} el grafo que se obtiene eliminando v y w de G, y agregando un vértice vw adyacente a todos los vértices que eran adyacentes a v o w. Demostrar que $\chi(G_{vw}) = \chi(G)$.

(b) Demostrar que para cada $c \in \mathbb{N}$, el grafo completo K_c minimiza tanto la cantidad de vértices como la cantidad de ejes en el conjunto de grafos que tienen número cromático c.
SUGERENCIA: Usar el punto anterior.

SOLUCIÓN:

(a) El coloreo óptimo de G en el cual v y w tienen el mismo color, sirve para pintar G_{vw}, lo cual prueba $\chi(G_{vw}) \leq \chi(G)$. Cualquier coloreo óptimo de G_{vw} puede usarse para conseguir un coloreo de G en el cual v y w tienen el mismo color, lo cual prueba $\chi(G_{vw}) \geq \chi(G)$.

(b) Sea G tal que $\chi(G) = c$, y consideremos un coloreo óptimo de G. G debe tener al menos c vértices. Si tiene más, entonces tiene al menos 2 vértices del mismo color, que necesariamente no son adyacentes, y por el punto anterior los podemos colapsar obteniendo un grafo con igual número cromático pero con menos vértices y menos o igual cantidad de ejes. Esto lo podemos repetir hasta obtener un grafo con exactamente c vértices, que debe ser K_c porque si no lo fuera podría ser coloreado con menos de c colores.

NOTA: Propuesto originalmente por Pablo Heiber (segundo punto, aunque él tiene una demo de unicidad que no entendí).

TOMADO: 2010C2P2 06-DIC-2010 (primer punto), 2010C2R2 22-DIC-2010.

322. Dada una colección de intervalos sobre la recta real, se define su grafo de intervalos como 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.

    Demostrar que si G es un grafo de intervalos, entonces $\chi(G)$ coincide con la cantidad de vértices en un subgrafo completo máximo de G.

323. Sea G un grafo bipartito. Demostrar que $\chi(G^c)$ coincide con la cantidad de vértices del subgrafo completo máximo de G^c.

    SUGERENCIA: Considerar primero el caso en que G no tiene vértices aislados.

324. Sea G un grafo que tiene dos vértices $u \neq v$ tales que todo vértice adyacente a u es también adyacente a v. Demostrar que $\chi(G) = \chi(G - u)$. (Notar que u y v no son adyacentes porque si lo fueran v sería adyacente a sí mismo.)

    SOLUCIÓN: Es claro que $\chi(G - u) \leq \chi(G)$, porque $G - u$ es un subgrafo de G. Por lo tanto, para probar la igualdad, alcanza con probar $\chi(G) \leq \chi(G - u)$, para lo cual alcanza con obtener un coloreo válido de G que use a lo sumo $\chi(G-u)$ colores. Consideremos un coloreo óptimo de $G-u$. Pintemos a los nodos de G que no son u con los colores que tienen en ese coloreo, y pintemos a u con el mismo color que tiene v. Los nodos de $G-u$ no tienen conflictos en ese coloreo de G, por lo que para ver que el coloreo es válido, alcanza con ver que u no tiene conflictos. Supongamos que no es cierto, es decir, que u recibió el mismo color que algún nodo w adyacente a u. Como v tiene el mismo color que u, resulta que v y w tienen el mismo color. Sin embargo, w es adyacente a v (por ser adyacente a u), lo cual contradice que el coloreo de $G - u$ era válido.

    TOMADO: 2002C1P2 10-JUL-2002, 2013C2R2 21-DIC-2013.

325. Demostrar que un grafo G es 2-coloreable si y sólo si sus ejes pueden orientarse de manera tal que no se formen caminos dirigidos de más de un eje.

    SOLUCIÓN: Para la ida, pintar los nodos con a lo sumo dos colores, armar un conjunto V_1 con los nodos pintados con el primer color, y un conjunto V_2 con los nodos pintados del segundo color; como el coloreo es válido, sólo hay ejes entre nodos de diferente color; orientar todos los ejes hacia V_2; no se forman caminos dirigidos de más de un eje porque no hay arcos que salgan de V_2. Para la vuelta, considerar una orientación en la cual no se forman caminos dirigidos de más de un eje; sea V_1 el conjunto de nodos de los cuales salen arcos, y V_2 el conjunto de los nodos de los cuales no salen arcos; por definición, $V_1 \cup V_2 = V$, y $V_1 \cap V_2 = \emptyset$; a la vez, no hay arcos dentro de V_1 ni dentro de V_2; por lo tanto, si pintamos los nodos de V_1 con un color y los de V_2 con otro, obtenemos un coloreo válido, lo que demuestra que el grafo era 2-coloreable. Para demostrar todo esto pueden usarse algunas equivalencias para grafos bipartitos, pero hay que tener en cuenta que el grafo podría tener un solo nodo, y por lo tanto no sería bipartito.

    TOMADO: 2001C2R2 26-DIC-2001.

326. Sea G un grafo de n vértices tal que para todo vértice v se cumple que $d(v) \geq d$. Demostrar que $\chi(G) \geq n/(n-d)$.

SUGERENCIA: Sea v un vértice cualquiera de G. ¿Cuántos vértices pueden pintarse con el mismo color que v?

SOLUCIÓN: Considerar un coloreo óptimo de G. Sea V_i el conjunto de vértices pintados con el color i ($i = 1, 2, \ldots \chi(G)$). Tenemos que $n = \sum_{i=1}^{\chi(G)} |V_i| \leq \sum_{i=1}^{\chi(G)} (n-d) = \chi(G)(n-d)$.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2005C1R2 08-AGO-2005.

327. (a) Demostrar que para todo grafo $G = (V, E)$ se verifican
   i. $|V| \leq \chi(G)\alpha(G)$, y
   ii. $\chi(G) \leq |V| - \alpha(G) + 1$,

   donde $\alpha(G)$ es la cantidad de vértices de un conjunto independiente máximo de G.

   (b) Exhibir grafos G_1, G_2, G_3 y G_4 (no necesariamente distintos) tales que
   i. $|V_1| < \chi(G_1)\alpha(G_1)$,
   ii. $|V_2| = \chi(G_2)\alpha(G_2)$,
   iii. $\chi(G_3) < |V_3| - \alpha(G_3) + 1$, y
   iv. $\chi(G_4) = |V_4| - \alpha(G_4) + 1$.

   Justificar.

SOLUCIÓN:

(a) i. Considerar un coloreo óptimo de G, y sea V_i el conjunto de vértices pintados con el color i ($i = 1, 2, \ldots \chi(G)$); tenemos que $|V| = \sum_{i=1}^{\chi(G)} |V_i| \leq \sum_{i=1}^{\chi(G)} \alpha(G) = \chi(G)\alpha(G)$.
   ii. Una forma de colorear G es tomar un conjunto independiente máximo de G y pintarlo de un color, para luego pintar cada uno de los $|V| - \alpha(G)$ nodos restantes con un color diferente.

(b) Pendiente.

TOMADO: 2001C2P2 11-DIC-2001.

328. (a) Demostrar que para todo grafo $G = (V, E)$ se verifican
   i. $|E| \leq \chi'(G)\nu(G)$, y
   ii. $\chi'(G) \leq |E| - \nu(G) + 1$,

   donde $\nu(G)$ es la cantidad de ejes de una correspondencia máxima de G.

   (b) Exhibir grafos G_1, G_2, G_3 y G_4 (no necesariamente distintos) tales que
   i. $|E_1| < \chi'(G_1)\nu(G_1)$,
   ii. $|E_2| = \chi'(G_2)\nu(G_2)$,
   iii. $\chi'(G_3) < |E_3| - \nu(G_3) + 1$, y
   iv. $\chi'(G_4) = |E_4| - \nu(G_4) + 1$.

   Justificar.

SOLUCIÓN:

(a) i. Considerar un coloreo óptimo de G, y sea E_i el conjunto de ejes pintados con el color i ($i = 1, 2, \ldots \chi'(G)$); tenemos que $|E| = \sum_{i=1}^{\chi'(G)} |E_i| \leq \sum_{i=1}^{\chi'(G)} \nu(G) = \chi'(G)\nu(G)$.
   ii. Una forma de colorear G es tomar una correspondencia máxima de G y pintarla de un color, para luego pintar cada uno de los $|E| - \nu(G)$ ejes restantes con un color diferente.

(b) P_4, K_1, K_1 y K_2.

NOTA: Creo que G podría ser un multi-grafo. La primera desigualdad ahora es el Ejercicio 10.25.a de la Práctica.

TOMADO: 2003C1P2 11-JUL-2003 (primera desigualdad).

329. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia.

Sea $G = (V, E)$ un grafo que tiene un circuito hamiltoniano y es d-regular, con d impar.

(a) Demostrar que G tiene al menos dos correspondencias perfectas.

(b) Demostrar que si $d = 3$ entonces existe una partición de E en correspondencias perfectas.

(c) Demostrar que si $d = 3$ entonces $\chi'(G) = 3$.
SUGERENCIA: No confundir $\chi'(G)$ con $\chi(G)$.

SOLUCIÓN:

(a) Como $2m = \sum_{i=1}^{n} d_i = \sum_{i=1}^{n} 3 = 3n$, la cantidad de vértices resulta par. Una correspondencia está compuesta por ejes alternados del circuito hamiltoniano, la segunda por los otros ejes del circuito (que también son alternados).

(b) Existe un partición de E formada por 3 correspondencias perfectas. Dos de ellas son las del punto anterior, y la tercera está formada por los ejes que quedan. Estos últimos ejes forman una correspondencia perfecta porque son $n/2$ y no se tocan, ya que si eliminamos el circuito del grafo, todos los vértices quedan de grado 1.

(c) Basta demostrar la doble desigualdad. Por el teorema de Vizing sabemos $\chi'(G) \geq \Delta(G) = 3$. Para demostrar $\chi'(G) \leq 3$ basta exhibir un coloreo de ejes que use a lo sumo 3 colores. Tal coloreo puede lograrse pintando cada una de las correspondencias del punto anterior con un color diferente. El coloreo es válido porque cada conjunto pintado de un mismo color es una correspondencia, y todos los ejes son pintados ya que las 3 correspondencias forman una partición de E.

NOTA: Propuesto originalmente por Diego Delle Donne y Michel Mizrahi.

TOMADO: 2013C1P2 10-JUL-2013 (primer punto con una correspondencia, y último punto).

330. Demostrar que si G es un grafo bipartito, entonces $\chi'(G) = d_{\max}(G)$.

SOLUCIÓN: La demostración que se me ocurre no es del todo sencilla. Consiste en colorear los ejes con a lo sumo $d_{\max}(G) + 1$ colores, y luego recolorear de a uno los ejes que recibieron el último color. Tal vez salga por inducción en la cantidad de vértices del grafo, eliminando un vértice de grado máximo.

NOTA: Ver en Gross si está la demostración, y qué tan complicada es. Ver de hacer una variante para grafos sin circuitos (bosques) que es un caso particular de bipartito, y/o pedir un algoritmo para colorear de manera óptima. El algoritmo para árboles empieza de un nodo cualquiera, colorea los ejes incidentes con colores distintos, y repite recursivamente para los ejes no coloreados de los 'hijos'.

331. Demostrar que un grafo G tiene $\chi'(G) = 1$ si y sólo si $d_{\max}(G) = 1$.

TOMADO: 2001C1R2 06-AGO-2001.

332. 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.

Demostrar que $\chi'(G) = \chi(L(G))$.

SOLUCIÓN: Un coloreo óptimo de los ejes de G produce un coloreo válido de los nodos de $L(G)$ con la misma cantidad de colores, por lo que $\chi'(G) \geq \chi(L(G))$. Un coloreo óptimo de los nodos de $L(G)$ produce un coloreo válido de los ejes de G con la misma cantidad de colores, por lo que $\chi'(G) \leq \chi(L(G))$.

NOTA: Es parecido al Ejercicio 366.

TOMADO: 2008C1P2 11-JUL-2008.

Clique, conjunto independiente, cubrimientos, correspondencia

333. Para cada uno de los siguientes grafos determinar $\omega(G)$ = cantidad de vértices de un subgrafo completo máximo del grafo. Expresar los resultados en función de n, m, p, q y h. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) árbol de n vértices

(e) grafo bipartito conexo de n vértices y m ejes

(f) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(g) $K_{p,q}$

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

SOLUCIÓN: En la tabla siguiente aparece la solución para este y otros ejercicios similares, como los Ejercicios 334, 335, 336 y 337.

G	$\omega(G)$	$\alpha(G)$	$\tau(G)$	$\nu(G)$	$\rho(G)$
	= cardinalidad de				
	subgrafo completo máximo	conjunto independiente máximo	cubrimiento de aristas por vértices mínimo	correspon-dencia máxima	cubrimiento de vértices por aristas mínimo
K_n	n	1	$n-1$	$\lfloor n/2 \rfloor$	$\lceil n/2 \rceil$, si $n>1$
C_n	IF$(n=3;3;2)$	$\lfloor n/2 \rfloor$	$\lceil n/2 \rceil$	$\lfloor n/2 \rfloor$	$\lceil n/2 \rceil$, si $n>1$
P_n	IF$(n=1;1;2)$ $=\min\{n,2\}$	$\lceil n/2 \rceil$	$\lfloor n/2 \rfloor$	$\lfloor n/2 \rfloor$	$\lceil n/2 \rceil$, si $n>1$
A n	IF$(n=1;1;2)$ $=\min\{n,2\}$	variable	variable	variable	variable
GBC n m	2	variable	variable	variable	variable
W_n ($n+1$ vértices)	IF$(n=3;4;3)$	$\lfloor n/2 \rfloor$	$1+\lceil n/2 \rceil$	$\lfloor (n+1)/2 \rfloor$	$\lceil (n+1)/2 \rceil$
$K_{p,q}$	2	$\max\{p,q\}$	$\min\{p,q\}$	$\min\{p,q\}$	$\max\{p,q\}$
ABC h ($n=2^{h+1}-1$)	IF$(n=1;1;2)$ $=\min\{n,2\}$	h par: $(2^{h+2}-1)/3$ h impar: $(2^{h+2}-2)/3$	$n-\alpha(G)$	$n-\alpha(G)$	$\alpha(G)$, si $n>1$

Algunas de esas soluciones pueden determinarse a partir de otras utilizando las siguientes propiedades conocidas. Las propiedades que involucran a $\rho(G)$ requieren que esté definido, es decir, que no haya vértices aislados.

- $\omega(G^c) = \alpha(G)$,
- $\alpha(G) + \tau(G) = |V(G)|$,
- $\nu(G) + \rho(G) = |V(G)|$,
- $\tau(G) \geq \nu(G)$, y
- $\rho(G) \geq \alpha(G)$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Algunos casos son triviales.

TOMADO: 2019C1P2 05-JUL-2019 (excepto A n, GBC n m y ABC h).

334. Para cada uno de los siguientes grafos determinar $\alpha(G)$ = cardinalidad de un conjunto independiente máximo del grafo. Expresar los resultados en función de n, p, q y h. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(e) $K_{p,q}$

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

SOLUCIÓN: Ver Ejercicio 333.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. No se incluye A n (variable, por ejemplo $K_{1,n-1}$ versus P_n), ni GBC n m (variable, porque incluye a los árboles).

TOMADO: 2019C1R2 19-JUL-2019 (excepto K_n).

335. Para cada uno de los siguientes grafos determinar $\tau(G)$ = cardinalidad de un cubrimiento de aristas por vértices mínimo del grafo. Expresar los resultados en función de n, p, q y h. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(e) $K_{p,q}$

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

SOLUCIÓN: Ver Ejercicio 333.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. No se incluye A n (variable, por ejemplo $K_{1,n-1}$ versus P_n), ni GBC n m (variable, porque incluye a los árboles).

336. Para cada uno de los siguientes grafos determinar $\nu(G)$ = cardinalidad de una correspondencia máxima del grafo. Expresar los resultados en función de n, p, q y h. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(e) $K_{p,q}$

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

SOLUCIÓN: Ver Ejercicio 333.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. No se incluye A n (variable, por ejemplo $K_{1,n-1}$ versus P_n), ni GBC n m (variable, porque incluye a los árboles).

337. Para cada uno de los siguientes grafos determinar $\rho(G)$ = cardinalidad de un cubrimiento de vértices por aristas mínimo del grafo. Expresar los resultados en función de n, p, q y h. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(e) $K_{p,q}$

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

SOLUCIÓN: Ver Ejercicio 333.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. No se incluye A n (variable, por ejemplo $K_{1,n-1}$ versus P_n), ni GBC n m (variable, porque incluye a los árboles).

338. (a) Demostrar que para todo grafo G se verifica $\nu(G) \leq \tau(G) \leq 2\nu(G)$, donde $\nu(G)$ es la cantidad de aristas de una correspondencia máxima de G, y $\tau(G)$ es la cantidad de vértices de un cubrimiento de aristas por vértices mínimo de G.

(b) Exhibir grafos G_1, G_2, G_3 y G_4 (no necesariamente distintos) tales que

i. $\nu(G_1) < \tau(G_1)$,

ii. $\nu(G_2) = \tau(G_2)$,

iii. $\tau(G_3) < 2\nu(G_3)$, y

iv. $\tau(G_4) = 2\nu(G_4)$.

Justificar.

SOLUCIÓN:

(a) La primera desigualdad probablemente se vio en clase, y vale para cualquier cubrimiento y cualquier correspondencia, en particular para los óptimos. Cada eje de una correspondencia debe ser cubierto por un vértice distinto, ya que los ejes de una correspondencia no comparten extremos. Otros ejes que pudieran estar en el grafo también deben ser cubiertos, de modo que eventualmente son necesarios todavía más vértices.

Respecto de la segunda desigualdad, sea M una correspondencia máxima de G, y sea R el conjunto de vértices que son extremos de los ejes de M. Si G tuviera alguna arista no cubierta por R, entonces M no sería una correspondencia máxima, lo que implica que R es un cubrimiento de aristas por vértices. La cardinalidad de un cubrimiento mínimo no puede superar la de R, de modo que $\tau(G) \leq |R| = 2\nu(G)$.

(b) K_3, K_1, K_2 y K_1.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Aparentemente los grafos para los cuales $\tau(G) = 2\nu(G)$ son la unión de completos impares.

339. Dado un grafo G, se definen

- $\omega(G)$ = cantidad de vértices de un subgrafo completo máximo de G;
- $\alpha(G)$ = cantidad de vértices de un conjunto independiente máximo de G;
- $\tau(G)$ = cantidad de vértices de un cubrimiento de aristas por vértices mínimo de G;
- $\nu(G)$ = cantidad de aristas de una correspondencia máxima de G; y
- $\rho(G)$ = cantidad de aristas de un cubrimiento de vértices por aristas mínimo de G.

Sean G y H dos grafos, y sea $G + H$ su grafo junta.

(a) Demostrar que $\omega(G + H) = \omega(G) + \omega(H)$.

(b) Demostrar que $\alpha(G + H) = \max\{\alpha(G), \alpha(H)\}$.

(c) Expresar $\tau(G + H)$ en función de $\tau(G)$ y $\tau(H)$. Justificar.

(d) ¿Es cierto que $\nu(G + H) = \nu(G) + \nu(H)$, o que $\nu(G + H) = \max\{\nu(G), \nu(H)\}$? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

(e) ¿Es cierto que $\rho(G + H) = \rho(G) + \rho(H)$, o que $\rho(G + H) = \max\{\rho(G), \rho(H)\}$? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

(a) Se cumple $\omega(G + H) = \alpha((G + H)^c) = \alpha(G^c \cup H^c) = \alpha(G^c) + \alpha(H^c) = \omega(G) + \omega(H)$. Alternativamente, basta demostrar la doble desigualdad.

Sea C_G el conjunto de nodos de un subgrafo completo máximo de G, y análogamente C_H para H. Sea C la unión (disjunta) de C_G y C_H. Cada par de nodos de C es adyacente en $G + H$, ya sea porque lo es en G, porque lo es en H, o porque uno de los nodos del par está en G y el otro en H. Por lo tanto, los nodos de C inducen un subgrafo completo en $G + H$, lo que implica $\omega(G + H) \geq |C| = |C_G| + |C_H| = \omega(G) + \omega(H)$.

Sea C el conjunto de nodos de un subgrafo completo máximo de $G+H$. Sea C_G el subconjunto de C que está en G, y análogamente C_H para H. Cada para de nodos de C_G es adyacente en G, porque en $G + H$ no se agregan ejes en G. Por lo tanto, los nodos de C_G inducen un subgrafo completo en G, lo que implica $|C_G| \leq \omega(G)$. Análogamente, $|C_H| \leq \omega(H)$. En consecuencia tenemos $\omega(G + H) = |C| = |C_G| + |C_H| \leq \omega(G) + \omega(H)$.

(b) Se cumple $\alpha(G{+}H) = \omega((G{+}H)^c) = \omega(G^c \cup H^c) = \max\{\omega(G^c), \omega(H^c)\} = \max\{\alpha(G), \alpha(H)\}$. Alternativamente, basta demostrar la doble desigualdad.

Para ver que $\alpha(G{+}H) \leq \max\{\alpha(G), \alpha(H)\}$, supongamos por absurdo $\alpha(G{+}H) > \max\{\alpha(G), \alpha(H)\}$. Sea C un conjunto independiente máximo en $G + H$, por lo que $|C| = \alpha(G + H)$. Este conjunto contiene únicamente nodos de G o únicamente nodos de H, porque en $G + H$ los nodos de G y H son adyacentes entre sí. Supongamos sin pérdida de generalidad que C contiene únicamente nodos de G. Los nodos de C forman un conjunto independiente en G, porque en $G + H$ no se eliminan ejes de G. Esto implica $\alpha(G) \geq |C|$, y en consecuencia $|C| = \alpha(G + H) > \max\{\alpha(G), \alpha(H)\} \geq \alpha(G) \geq |C|$, lo cual resulta absurdo.

Un conjunto independiente máximo en G es un conjunto independiente en $G + H$, porque en $G + H$ no se agregan ejes en G. Por lo tanto $\alpha(G + H) \geq \alpha(G)$. Análogamente $\alpha(G + H) \geq \alpha(H)$, lo que implica $\alpha(G + H) \geq \max\{\alpha(G), \alpha(H)\}$.

(c) Es sabido que $\alpha(\cdot) + \tau(\cdot) = n(\cdot)$, donde $n(\cdot)$ es la cantidad de vértices. Por el punto anterior tenemos entonces $\tau(G + H) = n(G + H) - \alpha(G + H) = n(G) + n(H) - \max\{\alpha(G), \alpha(H)\} = n(G) + n(H) - \max\{n(G) - \tau(G), n(H) - \tau(H)\}$.

Podemos simplificar un poco esta expresión si tenemos en cuenta que $\max\{a, b\} = a + b - \min\{a, b\}$. En tal caso resulta $\tau(G + H) = n(G) + n(H) - [n(G) - \tau(G) + n(H) - \tau(H) - \min\{n(G) - \tau(G), n(H) - \tau(H)\}] = \tau(G) + \tau(H) + \min\{n(G) - \tau(G), n(H) - \tau(H)\}$.

En resumen, $\tau(G + H) = n(G) + n(H) - \max\{n(G) - \tau(G), n(H) - \tau(H)\} = \tau(G) + \tau(H) + \min\{n(G) - \tau(G), n(H) - \tau(H)\}$.

Intuitivamente, la última expresión nos dice que una forma óptima de cubrir todas las aristas de $G + H$ consiste en cubrir de forma óptima todas las de G, cubrir de forma óptima todas las de H, y cubrir las aristas no cubiertas que van de G a H con los vértices no elegidos hasta ahora de G o de H (el más chico de los dos).

(d) No es cierto. Por ejemplo, para $G = pK_1$ y $H = qK_1$ resulta $G + H = K_{p,q}$, y por lo tanto $\nu(G) = \nu(H) = 0$, $\nu(G) + \nu(H) = \max\{\nu(G), \nu(H)\} = 0$, $\nu(G + H) = \min\{p, q\} \geq 1$.

(e) No es cierto. Por ejemplo, para $G = pK_1$ y $H = qK_1$ resulta $G + H = K_{p,q}$, y por lo tanto $\rho(G + H) = \max\{p, q\}$, mientras que $\rho(G)$, $\rho(H)$, $\rho(G) + \rho(H)$ y $\max\{\rho(G), \rho(H)\}$ están indefinidos porque hay vértices aislados.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2018C1R2 20-JUL-2018.

340. Sea G un grafo de n vértices y m ejes. Diseñar un algoritmo de complejidad $O(n^2)$ que encuentre un

Variante 1: subgrafo completo de G que sea maximal (no necesariamente máximo).

Variante 2: conjunto independiente de G que sea maximal (no necesariamente máximo).

Variante 3: cubrimiento de aristas por vértices de G que sea minimal (no necesariamente mínimo).

Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(m + n)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

SOLUCIÓN: Para la Variante 3, por el Ejercicio 12.13.a de la Práctica, I es un conjunto independiente si y sólo si $R = V(G) - I$ es un cubrimiento de aristas por vértices. Puede demostrarse además que I es maximal si y sólo si R es minimal. Por lo tanto, basta calcular un conjunto independiente maximal y luego tomar los nodos que no fueron elegidos, lo cual tiene la complejidad requerida si el cálculo del conjunto independiente la tiene, ya que complementar los nodos es $O(n)$. Probablemente pueda resolverse también de manera directa, con ideas similares a las que aparecen a continuación para las primeras dos variantes.

Inicializar un vector de n contadores en 0 y recorrer los nodos de a uno. Para cada nodo, si cumple determinada condición, agregarlo al conjunto y sumar 1 a los contadores de todos sus nodos adyacentes. Si la condición para cada nodo puede evaluarse en $O(1)$, la complejidad es $O(m + n)$ con listas de adyacentes. La condición para elegir un nodo en la Variante 1 es que su contador coincida con la cantidad de nodos ya elegidos (eso indica que es adyacente a todos los nodos ya elegidos). La condición para elegir un nodo en la Variante 2 es que su contador sea 0 (eso indica que no es adyacente a ninguno de los nodos ya elegidos).

Alternativamente, el contador para cada nodo puede calcularse recién en el momento en que el nodo es analizado, si se mantiene un vector de n valores lógicos que indican para cada nodo si fue elegido o no. En cada paso se recorren los adyacentes del nodo actual, y se cuenta cuántos de ellos fueron elegidos. Si el nodo actual es elegido, se lo marca en el vector.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2014C2P2 01-DIC-2014 (Variante 2), 2014C2R2 19-DIC-2014 (Variante 3), 2017C1P2 07-JUL-2017 (Variante 1).

341. Sea G un grafo de m ejes sin ciclos simples de longitud 3. Sea $\alpha(G)$ la cantidad de vértices de un conjunto independiente máximo de G, y sea τ la cantidad de vértices de cualquier cubrimiento de aristas por vértices de G. Demostrar que $m \leq \alpha(G)\tau$.

    SUGERENCIA: Demostrar que para todo vértice v de G se cumple $d(v) \leq \alpha(G)$.

    SOLUCIÓN: La demostración es esencialmente la misma que la del Ejercicio 11.5 de la Práctica. Para todo nodo v sus adyacentes no son adyacentes entre sí porque en tal caso habría ciclos simples de longitud 3; eso implica que sus adyacentes forman un conjunto independiente, y por lo tanto $d(v) \leq \alpha(G)$. Sea w_i el i-ésimo nodo del cubrimiento, y sea E_i el conjunto de ejes cubiertos por ese nodo. Se cumple $m = |\cup_{i=1}^{\tau} E_i| \leq \sum_{i=1}^{\tau} |E_i| = \sum_{i=1}^{\tau} d(w_i) \leq \sum_{i=1}^{\tau} \alpha(G) = \alpha(G)\tau$.

    NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

    TOMADO: 2016C1P2 02-JUL-2016.

342. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

    Dado un grafo G, se define $\omega(G)$ como la cantidad de vértices de un subgrafo completo máximo de G.

    Sean G y H dos grafos.

    (a) Demostrar que si G es homomorfo a H entonces $\omega(G) \leq \omega(H)$.
    (b) Demostrar que $\omega(H) \geq k$ si y sólo si K_k es homomorfo a H.
    (c) Demostrar que $\omega(H)$ es el máximo $k \in \mathbb{N}$ tal que K_k es homomorfo a H.

    SOLUCIÓN:

    (a) Sea W el conjunto de nodos de un subgrafo completo máximo de G. Cada par de nodos de W es adyacente en G, por lo que sus imágenes son adyacentes en H, y por lo tanto son nodos diferentes. De modo que la imagen de W tiene la misma cantidad de nodos que W y son adyacentes de a pares. Es decir, en H hay un subgrafo completo con esa cantidad de nodos, y el máximo tiene que ser mayor o igual.
    (b) Para la vuelta, si K_k es homomorfo a H, por el punto anterior sabemos que $\omega(K_k) \leq \omega(H)$, lo que implica $\omega(H) \geq k$ ya que $\omega(K_k) = k$. Para la ida, por hipótesis hay un subgrafo completo de k nodos en H, de modo que para tener un homomorfismo de K_k a H basta asignar a cada nodo de K_k un nodo diferente de ese subgrafo completo de H.
    (c) Por el punto anterior, H contiene un subgrafo completo de al menos k nodos si y sólo si K_k es homomorfo a H. Lo que queremos probar entonces es que $\omega(H)$ es el máximo $k \in \mathbb{N}$ tal que H contiene un subgrafo completo de al menos k nodos, lo cual es cierto porque esa es la definición de $\omega(H)$.

    NOTA: El espíritu es el mismo que en el Ejercicio 308.

    TOMADO: 2012C1P2 11-JUL-2012.

343. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

    Sean $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ dos grafos, y $f : V_1 \to V_2$ una función. Demostrar que f es un homorfismo de G_1 a G_2 si y sólo si la preimagen de cada conjunto independiente en G_2 es un conjunto independiente en G_1.

SOLUCIÓN: Para la ida, sea I_2 un conjunto independiente de G_2, y sea I_1 su preimagen en G_1; si I_1 no fuera independiente, tendría al menos dos nodos adyacentes, cuyas imágenes serían adyacentes en I_2 debido a que f es homomorfismo. Para la vuelta, hay que ver que nodos adyacentes van a parar por f a nodos adyacentes; sean entonces v y w dos nodos adyacentes en G_1; supongamos que $f(v)$ y $f(w)$ no son adyacentes en G_1; eso es porque o bien $f(v) = f(w)$, o son distintos pero no adyacentes; en ambos casos $f(v)$ y $f(w)$ forman un conjunto independiente en G_2 cuya preimagen no lo es en G_1, ya que incluye a v y a w.

TOMADO: 2012C2R2 19-DIC-2012.

344. Diseñar un algoritmo eficiente que dado un grafo $G = (V, E)$, encuentre una correspondencia maximal (no necesariamente máxima) de G. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(m+n)$, donde $m = |E|$ y $n = |V|$.

    SOLUCIÓN: Inicializar un vector lógico de n elementos en Falso. Recorrer todos los ejes y por cada eje usar el vector para decidir si puede ser agregado a la correspondencia, y en tal caso marcar sus extremos en el vector.

    NOTA: Encontrado originalmente en Internet.

    TOMADO: 2009C1R2 26-AGO-2009.

345. Sea G un grafo, M una correspondencia *maximal* de G, y M^* una correspondencia *máxima* de G. Demostrar que $|M| \leq |M^*| \leq 2|M|$.

    SUGERENCIA: Para la segunda desigualdad demostrar primero que $|V(M)| \geq |V(M^*)|/2$, donde $V(X)$ es el conjunto de vértices cubiertos por la correspondencia X.

    SOLUCIÓN: La primera desigualdad es trivial por definición. Para la segunda, dado cualquier eje de M^* al menos uno de sus extremos está cubierto en M, ya que caso contrario M no sería maximal. Eso prueba la sugerencia, lo cual implica la segunda desigualdad ya que $|V(X)| = 2|X|$.

    NOTA: Encontrado originalmente en Internet.

    TOMADO: 2016C2R2 19-DIC-2016.

346. Sea G un grafo. Sea v un vértice no aislado de G. Demostrar que existe una correspondencia máxima en G tal que v es extremo de algún eje de la correspondencia.

    SOLUCIÓN: Tomar cualquier correspondencia máxima M. Si v está matcheado, listo. Si no, agregar a M cualquier eje $e = (v, w)$ incidente a v, y sacar de M el eje que había incidente a w (que debe existir, porque de lo contrario e estaría en M).

    TOMADO: 2003C1R2 04-AGO-2003.

347. Se dispone de un número arbitrariamente grande de cajas de capacidad 1, en las cuales se desea almacenar p items, donde el i-ésimo item tiene tamaño a_i ($i = 1, 2, \ldots p$). Cada caja puede almacenar cualquier número de items, con la sola condición de que la suma de los tamaños de los items almacenados en ella no exceda la capacidad de la caja. El objetivo es almacenar los items usando el menor número posible de cajas. Se dispone de un algoritmo que dado un grafo, encuentra una correspondencia máxima en el mismo. Si para todo item su tamaño cumple que $1/3 < a_i \leq 1$, explicar cómo podría resolverse el problema de almacenar los items utilizando el algoritmo mencionado. Justificar.

    SOLUCIÓN: Poner un vértice por cada item; poner un eje entre cada par de items si y sólo si los items pueden ponerse en la misma caja. Siendo $a_i > 1/3$, cada caja va a contener uno o dos items. El objetivo es entonces aparear los items en cajas.

    TOMADO: 2001C2R2 26-DIC-2001.

348. Considerar el siguiente problema de planificación de tareas. Existen p tareas y q recursos. Cada tarea puede ser ejecutada indistintamente en una de dos máquinas, requiriendo la utilización transitoria de algunos de los recursos. Llamamos R_{ij} al número de unidades del recurso j-ésimo que la tarea i-ésima necesita para su ejecución ($R_{ij} \in \mathbb{R}$; $i = 1, 2, \ldots p$; $j = 1, 2, \ldots q$). Asimismo, existe una cota superior al número de unidades de cada recurso que pueden utilizarse simultáneamente. Llamamos r_j a la cota superior correspondiente al recurso j-ésimo ($r_j \in \mathbb{R}$; $j = 1, 2, \ldots q$). Todas las tareas demoran el mismo tiempo. Si ignoráramos la utilización de recursos, las p tareas podrían

ejecutarse en $\lceil p/2 \rceil$ unidades de tiempo, utilizando las dos máquinas la mayor parte del tiempo posible. Sin embargo, los valores r_j imponen restricciones sobre las formas permitidas de ejecutar las tareas. El objetivo es encontrar una forma de ejecutar las tareas que demore el mínimo tiempo posible, sin utilizar ningún recurso más allá de la cantidad permitida. Se dispone de un algoritmo que dado un grafo, encuentra una correspondencia máxima en el mismo. Explicar cómo podría resolverse el problema de planificar la ejecución de las tareas utilizando el algoritmo mencionado. Justificar.

SOLUCIÓN: Poner un vértice por cada tarea; poner un eje entre cada par de tareas si y sólo si las tareas pueden ejecutarse simultáneamente, es decir, si para todo j se cumple que $R_{aj} + R_{bj} \leq j$, donde a y b son las tareas en cuestión.

TOMADO: 2002C1P2 10-JUL-2002.

349. Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia.

Determinar para qué valores de n, p, q y h los siguientes grafos tienen alguna correspondencia perfecta. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(e) $K_{p,q}$

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

SOLUCIÓN: Para que haya correspondencia perfecta la cantidad de nodos debe ser par.

K_n: Alcanza con que n sea par. Aparear los nodos arbitrariamente, ya que cualquier eje existe. Alternativamente, usar alguno de los puntos siguientes, ya que esos grafos son subgrafos del completo par.

C_n: Alcanza con que n sea par. Elegir ejes alternados del ciclo.

P_n: Alcanza con que n sea par. Elegir ejes alternados del camino, empezando por uno de los extremos.

$K_{p,q}$: Debe ser $p = q$ ya que todo eje tiene sus extremos en partes diferentes y los ejes de la correspondencia no pueden tocarse. Alcanza con eso, ya que en tal caso se pueden aparear los nodos de partes diferentes arbitrariamente, debido a que cualquier eje entre ellos existe.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. No se incluye A n (variable), ni GBC n m (variable).

TOMADO: 2013C1R2 05-AGO-2013 (excepto W_n y ABC h).

350. Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia.

Variante 1: Exhibir una familia infinita de grafos tal que el n-ésimo grafo tenga

Variante 2: Sea $n \in \mathbb{N}$. Construir un grafo con

$2n$ vértices, n^2 ejes, y una única correspondencia perfecta. Justificar.

SOLUCIÓN: Para $n = 1$ es trivial. Para $n > 1$, tomar la solución para $n - 1$, agregarle un vértice adyacente a todos los otros, y otro vértice adyacente sólo al otro que se agregó. El eje entre los últimos dos vértices agregados debe pertenecer a cualquier correspondencia perfecta.

NOTA: Basado en 2.35 de Manber. La Variante 1 es más fácil porque sugiere la solución inductiva.

TOMADO: 2005C1P2 15-JUL-2005 (Variante 1).

351. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia.

Sea $G = (V, E)$ un grafo bipartito d-regular, con $d \geq 1$. Sea (V_1, V_2) cualquier partición de V tal que tanto V_1 como V_2 son conjuntos independientes.

(a) Demostrar que todo subconjunto de r vértices de V_1 es adyacente a r o más vértices de V_2. Deducir que G tiene una correspondencia completa de V_1 en V_2 (según el Ejercicio 11.9.b de la Práctica).

(b) Demostrar que $|V_1| = |V_2|$. Deducir que G tiene una correspondencia perfecta. Justificar.

(c) Demostrar que E puede particionarse en d correspondencias perfectas. Deducir que G tiene al menos d correspondencias perfectas. Justificar.

SOLUCIÓN:

(a) Los adyacentes de los r nodos de V_1 reciben al menos rd ejes, y para eso se necesitan al menos r nodos de V_2. Por el ejercicio mencionado en el enunciado se deduce que G tiene una correspondencia completa de V_1 en V_2.

(b) Por el punto anterior $|V_1| \leq |V_2|$, pero como los papeles de V_1 y V_2 pueden intercambiarse también es cierto que $|V_2| \leq |V_1|$, de modo que resultan iguales. Otra forma de demostrarlo es $d|V_1| = m = d|V_2|$. La correspondencia completa del punto anterior toca a todos los nodos de V_2, por lo que resulta perfecta.

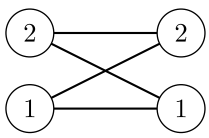
(c) Inducción en d. Para $d = 1$ es el punto anterior. Para $d > 1$, sea C la correspondencia perfecta del punto anterior. El grafo $G-C$ es $(d-1)$-regular, con $d-1 \geq 1$, de modo que por hipótesis inductiva sus ejes pueden particionarse en $d-1$ correspondencias perfectas, las que junto con C forman una partición de E en correspondencias perfectas. Como hay una partición de E en d correspondencias perfectas, G tiene al menos esas correspondencias perfectas. Se deduce que $K_{d,d}$ tiene $d!$ correspondencias perfectas, aunque por otro lado eso es obvio. Talvez vale para todo G dentro de las hipótesis del enunciado (al principio lo creí, pero ahora no sé).

NOTA: Propuesto originalmente por Michel Mizrahi (hasta el segundo punto).

TOMADO: 2012C1R2 08-AGO-2012.

352. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Un grafo es d-numerable ($d \in \mathbb{N}$) si y sólo si es d-regular y es posible asignar un número entero entre 1 y d a cada vértice de manera tal que para todo vértice sus vecinos tengan números diferentes entre sí. Una d-numeración es la mencionada asignación. Un grafo es d-numerado si y sólo si tiene asociada una d-numeración. Por ejemplo, $K_{2,2}$ es 2-numerable y una posible 2-numeración se muestra en el siguiente grafo 2-numerado.



Sea G un grafo de n vértices.

(a) Demostrar que si G es d-numerado entonces para cada entero entre 1 y d existen al menos dos vértices adyacentes que tienen asignado ese entero. Deducir que si G es d-numerable entonces $n \geq 2d$.

(b) Demostrar que si G es d-numerado entonces para cada entero entre 1 y d hay exactamente n/d vértices que tienen asignado ese entero. Deducir que si G es d-numerable entonces n es múltiplo de d.

(c) Demostrar que si G es d-numerable entonces n es múltiplo de 2.

(d) Demostrar que G es d-numerable si y sólo si cada una de sus componentes conexas lo es.

(e) Demostrar que G es 1-numerable si y sólo si cada componente conexa de G es K_2.

(f) Demostrar que G es 2-numerable si y sólo si cada componente conexa de G es un ciclo simple formado por una cantidad de vértices que es múltiplo de 4.

(g) Exhibir todos los grafos no isomorfos 3-numerables de 6 vértices. Justificar.

(h) Para cada $d \in \mathbb{N}$ exhibir un grafo d-numerable que tenga la mínima cantidad posible de vértices. Justificar.

(i) Para cada $d \in \mathbb{N}$ exhibir un grafo d-regular que no sea d-numerable, o demostrar que tal grafo no existe. Justificar.

SOLUCIÓN: Notemos que en cualquier grafo d-numerado, para cada vértice v y cada entero entre 1 y d, el vértice v tiene exactamente un vecino que tiene asignado ese entero. Esto es así ya que por definición no puede haber más de uno, mientras que si no hubiera ninguno, existirían al menos dos vecinos con el mismo entero asignado, porque habría menos de d enteros para asignar a d vecinos.

(a) Sea k entre 1 y d. Sea v cualquier nodo. Por lo dicho al principio existe v_k adyacente a v que tiene asignado el valor k. Por lo dicho al principio existe w_k adyacente a v_k que también tiene asignado el valor k.

(b) Sea k entre 1 y d. Como cada nodo tiene d adyacentes, si sumamos sobre todos los nodos la cantidad de nodos adyacentes que tienen asignado el valor k, vamos a sumar d veces cada nodo que tiene asignado el valor k. Por lo tanto, la cantidad total de nodos que tienen asignado el valor k es $1/d$ veces dicha suma. Como para cada nodo la cantidad de nodos adyacentes que tienen asignado el valor k es exactamente 1 según dijimos, mientras que la cantidad de sumandos es n, resulta que la cantidad buscada es n/d.

Alternativamente, basta ver que para cada entero entre 1 y d la cantidad de vértices que tienen asignado ese entero es la misma. Como hay n vértices y d enteros, esa cantidad debe ser n/d. Sean $i \neq j$ dos enteros entre 1 y d. Por lo dicho al principio cada vértice que tiene asignado el valor i tiene exactamente un vecino que tiene asignado el valor j y viceversa. Eso significa que hay una biyección entre los vértices que tienen asignado el valor i y los que tienen asignado el valor j, de modo que las cantidades son iguales.

(c) Si d es par, como por el punto anterior n es múltiplo de d, entonces n es par. Si d es impar, por sumatoria de grados tenemos que $2m = nd$, de modo que nd es par, y al ser d impar debe ser n par.

(d) Basta ver que una d-numeración de G induce una d-numeración de cada componente conexa, y viceversa.

(e) Para la ida, si G es 1-numerable, en particular es 1-regular, y por el Ejercicio 49 cada una de sus componentes conexas es K_2. Para la vuelta, por el punto anterior basta ver que K_2 es 1-numerable, lo cual es inmediato.

(f) Para la ida, si G es 2-numerable, en particular es 2-regular, y por el Ejercicio 49 cada una de sus componentes conexas es un ciclo simple. Consideremos una 2-numeración de G y cualquier componente conexa. Por el primer punto y otro punto anterior, esa componente conexa tiene dos nodos adyacentes v_1 y w_1 que tienen asignado el valor 1. El nodo w_1 tiene un nodo adyacente $v_2 \neq v_1$ que tiene asignado el valor 2. El nodo v_2 tiene un nodo adyacente $w_2 \neq w_1$ que tiene asignado el valor 2. Es así que tenemos un camino v_1, w_1, v_2, w_2 dentro del ciclo. Por lo dicho al principio el mismo patrón de asignaciones debe repetirse una y otra vez, hasta que el último nodo es adyacente a v_1, lo cual produce una cantidad de vértices que es múltiplo de 4.

Para la vuelta, por un punto anterior basta ver que un ciclo simple formado por una cantidad de vértices que es múltiplo de 4 es 2-numerable. Es fácil ver que una 2-numeración puede lograrse repitiendo cierta cantidad de veces el patrón mencionado en el párrafo anterior.

(g) Por el primer punto en cualquier 3-numeración hay (al menos) dos vértices adyacentes v_1 y w_1 que tienen asignado el valor 1. Por lo dicho al principio v_1 tiene otros dos adyacentes w_2 y w_3 que tienen asignados respectivamente los valores 2 y 3, sin más adyacentes para v_1. Por lo dicho al principio w_1 tiene otros dos adyacentes v_2 y v_3 que tienen asignados respectivamente los valores 2 y 3, sin más adyacentes para w_1. Notar que $v_2 \neq w_2$ y $v_3 \neq w_3$, ya que caso contrario w_2 o w_3 tendrían dos adyacentes con el valor 1 asignado (v_1 y w_1). Como ya tenemos los 6 nodos del grafo, por lo dicho al principio v_2 y w_2 deben ser adyacentes, lo mismo que v_3 y w_3. Para que el grafo termine de cumplir la definición, falta que v_2 y w_2 sean adyacentes a un nodo con el valor 3 asignado, y que v_3 y w_3 sean adyacentes a un nodo con el valor 2 asignado. Si v_2 es adyacente a w_3, debe ser w_2 adyacentes a v_3, y el grafo resulta $K_{3,3}$. La otra posibilidad es que v_2 sea adyacente a v_3, en cuyo caso w_2 debe ser adyacentes a w_3, y el grafo resulta un prisma de base triangular.

Alternativamente (y mejor) podemos notar que el complemento es 2-regular, es decir, colección de ciclos. Las únicas posibilidades para el complemento son $2C_3$ y C_6. Los grafos correspondientes sin complementar son $K_{3,3}$ y un prisma de base triangular. Para cada uno de ellos es fácil encontrar la 3-numeración descripta.

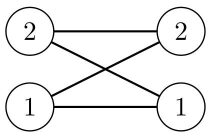
(h) Por el primer punto basta encontrar para cada d un grafo d-numerable que tenga $2d$ nodos. Es fácil ver que $K_{d,d}$ es d-numerable y tiene $2d$ nodos. Una posible d-numeración es asignar en cada parte los d valores posibles.

(i) Para $d = 1$ no es posible, ya que por un punto anterior y el Ejercicio 49 son equivalentes G es 1-numerable, G es 1-regular, y cada componente conexa de G es K_2. Para $d \geq 2$, K_{d+1} es d-regular pero no es d-numerable, ya que tiene $n = d + 1$ nodos y si fuera d-numerable por el primer punto debería ser $d + 1 \geq 2d$, lo cual implica $d \leq 1$.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez (n/d). Comparte algunos puntos con el Ejercicio 353.

TOMADO: 2017C2P1 04-OCT-2017 (tres primeros puntos), 2018C2R1 12-DIC-2018 (primer punto y dos últimos puntos).

353. Un grafo es d-numerable ($d \in \mathbb{N}$) si y sólo si es d-regular y es posible asignar un número entero entre 1 y d a cada vértice de manera tal que para todo vértice sus vecinos tengan números diferentes entre sí. Una d-numeración es la mencionada asignación. Un grafo es d-numerado si y sólo si tiene asociada una d-numeración. Por ejemplo, $K_{2,2}$ es 2-numerable y una posible 2-numeración se muestra en el siguiente grafo 2-numerado.



Sea G un grafo de n vértices d-numerable.

(a) Demostrar que si G es d-numerado entonces para cada entero entre 1 y d existen al menos dos vértices adyacentes que tienen asignado ese entero, y hay exactamente n/d vértices que tienen asignado ese entero.

(b) Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia.

Demostrar que G tiene una correspondencia perfecta.

SUGERENCIA: Considerar una d-numeración de G y elegir las aristas tales que sus dos extremos tengan asignado el mismo entero.

(c) Demostrar que n es múltiplo de $2d$.

SOLUCIÓN:

(a) Ver Ejercicio 352.

(b) Consideremos una d-numeración de G y sea M el conjunto de aristas de G tales que sus dos extremos tienen asignado un mismo entero en esa d-numeración. Digo que M es una correspondencia perfecta. Por absurdo supongamos que M no es una correspondencia, o si lo es, no es perfecta.

Si M no es una correspondencia, tiene al menos dos ejes (u, v) y (u, w) que inciden sobre el mismo vértice u. Por definición de M eso implica que en la d-numeración considerada, los vértices u, v y w tienen asignado el mismo entero, lo cual no puede ocurrir porque tanto v como w son adyacentes a u.

Si M es una correspondencia pero no es perfecta, existe al menos un vértice u que no es extremo de ningún eje de M. Por definición de d-numeración, existe al menos un vértice v adyacente a u que tiene asignado el mismo entero que u. Eso implica que el eje (u, v) está en M, lo cual contradice la definición de u.

(c) Consideremos una d-numeración de G. Sea k entre 1 y d. Por el primer punto hay n/d nodos que tienen asignado el valor k. Por cómo es la correspondencia del punto anterior, esa cantidad n/d es par, es decir, $n/d = 2z$ para algún z entero, o equivalentemente $n = 2dz$ que es múltiplo de $2d$.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez (n/d y correspondencia). Comparte algunos puntos con el Ejercicio 352.

TOMADO: 2017C2R2 18-DIC-2017.

354. Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia.

Sea G un grafo, y sea M una correspondencia perfecta de G.

(a) Demostrar que M es una correspondencia máxima.

(b) Demostrar que M es un cubrimiento mínimo de vértices por aristas.

SOLUCIÓN:

(a) Basta ver que si M' es cualquier correspondencia, entonces $|M'| \leq |M|$. Cada eje de cualquier correspondencia tiene dos extremos que no comparte con otros ejes de la correspondencia. El conjunto de todos los extremos son a lo sumo todos los vértices del grafo. En el caso de M, los extremos son efectivamente todos los vértices del grafo. Es así que $2|M'| \leq n = 2|M|$, lo que implica $|M'| \leq |M|$.

(b) Por definición de correspondencia perfecta, M es un cubrimiento de vértices por aristas. Basta ver que si R' es cualquier cubrimiento de vértices por aristas, entonces $|R'| \geq |M|$. Por el Ejercicio 11.3.b de la Práctica se cumple $|R'| \geq n/2$, mientras que de acuerdo a lo dicho en el primer punto tenemos $n = 2|M|$. Es así que $|R'| \geq n/2 = |M|$, lo que implica $|R'| \geq |M|$.

Alternativamente se puede argumentar que si M' es cualquier correspondencia y R' es cualquier cubrimiento de vértices por aristas, entonces $|M'| \leq n/2 \leq |R'|$ (esto es lo que permite afirmar que es verdadero lo que dice Ejercicio 11.2.e de la Práctica). Por lo tanto, si $|M'| = n/2$ se cumple que $|M'|$ es máxima, y si $|R'| = n/2$ se cumple que $|R'|$ es mínimo. Como M es algún M' y algún R', y además cumple $|M| = n/2$, se puede deducir lo que pide el ejercicio.

Alternativamente, podemos usar que $\nu(G) + \rho(G) = n$ para demostrar el primer punto a partir del segundo o viceversa, donde $\nu(\cdot)$ es la cantidad de ejes de una correspondencia máxima, y $\rho(\cdot)$ es la cantidad de ejes de un cubrimiento mínimo de vértices por aristas. Por ejemplo, si M es una correspondencia máxima entonces $\nu(G) = |M| = n/2$, de modo que $\rho(G) = n - \nu(G) = n - n/2 = n/2$. Como M es un cubrimiento de vértices por aristas de cardinalidad $\rho(G)$, es necesariamente mínimo.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2018C2P2 03-DIC-2018.

355. (a) Sea G un grafo de n vértices. Demostrar que $\alpha(G) \geq \lceil n/(1 + \Delta(G)) \rceil$, donde

- $\alpha(G)$ = cantidad de vértices de un conjunto independiente máximo de G; y
- $\Delta(G)$ = grado máximo de los vértices de G.

SUGERENCIA: Demostrar que existe un conjunto independiente de al menos $\lceil n/(1+\Delta(G)) \rceil$ elementos.

(b) Sean $n, \Delta \in \mathbb{Z}$ tales que $0 \leq \Delta \leq n - 1$. Exhibir un grafo de n vértices, todos ellos de grado a lo sumo Δ, cuyos conjuntos independientes máximos tengan exactamente $\lceil n/(1+\Delta) \rceil$ elementos. Justificar.

SUGERENCIA: Analizar primero casos particulares, como $n = 6$ y $\Delta = 0$, $n = 6$ y $\Delta = 1$, $n = 6$ y $\Delta = 2$, $n = 8$ y $\Delta = 3$, $n = 11$ y $\Delta = 3$.

SOLUCIÓN:

(a) Se puede demostrar que el algoritmo del Ejercicio 11.7 de la Práctica produce un conjunto independiente de al menos $\lceil n/(1+\Delta(G)) \rceil$ elementos. Sea k la cantidad de nodos del conjunto independiente, la cual coincide con la cantidad de iteraciones del algoritmo. En cada iteración se eliminan del grafo a lo sumo $1 + \Delta(G)$ nodos, de modo que si $n(i)$ es la cantidad de nodos eliminados en la i-ésima iteración tenemos $n = \sum_{i=1}^{k} n(i) \leq \sum_{i=1}^{k} 1 + \Delta(G) = (1 + \Delta(G))k$, es decir, $n \leq (1 + \Delta(G))k$. Por lo tanto $k \geq n/(1 + \Delta(G))$, lo que equivale a $k \geq \lceil n/(1 + \Delta(G)) \rceil$ ya que k es entero.

Otra forma de demostrarlo es usando el Ejercicio 327. Según ese ejercicio $n \leq \chi(G)\alpha(G)$, y como por el algoritmo secuencial de coloreo $\chi(G) \leq 1 + \Delta(G)$, resulta $n \leq (1 + \Delta(G))\alpha(G)$, lo que equivale a $\alpha(G) \geq n/(1 + \Delta(G))$, que a su vez equivale a $\alpha(G) \geq \lceil n/(1 + \Delta(G)) \rceil$ ya que $\alpha(G)$ es entero.

(b) Dividir los nodos en grupos de $1 + \Delta$ elementos (salvo el último grupo que podría tener menos), y formar un completo con cada grupo. El grado máximo es Δ, la cantidad de grupos es $\lceil n/(1 + \Delta) \rceil$, y de cada grupo exactamente un nodo puede formar parte de un conjunto independiente.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2008C1R2 25-JUL-2008.

356. Un grafo se dice d-regular si y sólo si todos sus vértices tienen grado d.

Sea G un grafo conexo de n vértices y d-regular, y sea $\alpha(G)$ la cantidad de vértices de un conjunto independiente máximo de G.

(a) Demostrar que si $3 \leq d \leq n - 2$ entonces $n \leq d \times \alpha(G)$.

(b) Demostrar que si $3 \leq d \leq n - 2$ entonces $\alpha(G) \geq \lceil n/d \rceil$.

(c) Mostrar con contraejemplos que la propiedad del primer punto no es cierta para $d = 0$, $d = 1$, $d = 2$ ni $d = n - 1$. Justificar.

SOLUCIÓN:

(a) Es un paso intermedio en el punto siguiente.

(b) La demostración es prácticamente igual a la segunda demostración del Ejercicio 355a. La diferencia es que es posible usar el teorema de Brooks para acotar $\chi(G) \leq \Delta(G) = d$, ya que G es conexo, y no es completo ni un ciclo impar para los valores previstos de d.

(c) Para $d = 0$ el único grafo que cumple las hipótesis es K_1, el cual sirve como contraejemplo ya que la izquierda vale 1 y la derecha 0. Para $d = 1$ el único grafo que cumple las hipótesis es K_2, el cual sirve como contraejemplo ya que la izquierda vale 2 y la derecha 1. Para $d = 2$ los grafos que cumplen las hipótesis son C_n con $n \geq 3$; cualquiera de ellos con n impar sirve como contraejemplo ya que la izquierda vale n y la derecha $n - 1$. Para $d = n - 1$ los grafos que cumplen las hipótesis son K_n; cualquiera de ellos sirve como contraejemplo ya que la izquierda vale n y la derecha $n - 1$.

TOMADO: 2014C2R2 19-DIC-2014 (parecido).

357. Dado un grafo G, se definen

- $\Delta(G)$ = grado máximo de los vértices de G; y
- $\nu(G)$ = cantidad de ejes de una correspondencia máxima de G.

Sea G un grafo de m ejes.

(a) El siguiente algoritmo toma como entrada al grafo G y produce como salida un conjunto M de ejes.

```
M = ∅
mientras G tenga ejes
        elegir un eje e de G
        M ← M ∪ {e}
        eliminar de G a e y a todos sus ejes incidentes
fin mientras
```

Mostrar que M es una correspondencia de G. Demostrar que si $m \geq 1$ entonces $|M| \geq \lceil m/(2\Delta(G) - 1) \rceil$, y deducir que $\nu(G) \geq \lceil m/(2\Delta(G) - 1) \rceil$.

(b) Demostrar que $\nu(G) \geq \lceil m/(1+\Delta(G)) \rceil$.

SOLUCIÓN:

(a) Es claro que M es una correspondencia porque en cada iteración se eliminan los ejes en conflicto con el elegido. El resto es similar a la primera demostración del Ejercicio 355a. La cantidad de iteraciones del algoritmo es $|M|$. En cada iteración se eliminan del grafo a lo sumo $2\Delta(G)-1$ ejes, de modo que si $m(i)$ es la cantidad de ejes eliminados en la i-ésima iteración tenemos $m = \sum_{i=1}^{|M|} m(i) \leq \sum_{i=1}^{|M|} 2\Delta(G)-1 = (2\Delta(G)-1)|M|$, es decir, $m \leq (2\Delta(G)-1)|M|$. Si $m \geq 1$ entonces $2\Delta(G)-1 > 0$, y por lo tanto $|M| \geq m/(2\Delta(G)-1)$, lo que equivale a $|M| \geq \lceil m/(2\Delta(G)-1) \rceil$ ya que $|M|$ es entero. Como el máximo es mayor o igual que uno particular, resulta $\nu(G) \geq \lceil m/(2\Delta(G)-1) \rceil$

(b) Es similar a la segunda demostración del Ejercicio 355a. Según el Ejercicio 328, o el Ejercicio 10.25.a de la Práctica, se cumple $m \leq \chi'(G)\nu(G)$. Por el teorema de Vizing $\chi'(G) \leq 1+\Delta(G)$, de donde resulta $m \leq (1+\Delta(G))\nu(G)$, lo que equivale a $\nu(G) \geq m/(1+\Delta(G))$, que a su vez equivale a $\nu(G) \geq \lceil m/(1+\Delta(G)) \rceil$ ya que $\nu(G)$ es entero.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2014C1P2 07-JUL-2014.

358. (a) Exhibir una familia infinita de grafos tal que el n-ésimo grafo tenga n vértices y $\Omega(3^{n/3})$ conjuntos independientes maximales. Justificar.

SUGERENCIA: Dividir los nodos en grupos de 3, y agregar ejes de tal modo que en cada grupo haya 3 opciones diferentes.

(b) Sea G un grafo de n vértices. Demostrar que G tiene $O(3^{n/3})$ conjuntos independientes maximales.

SOLUCIÓN: Armar $n/3$ triángulos.

NOTA: La solución es para el primer punto.

TOMADO: 2004C1P2 14-JUL-2004 (primer punto).

359. (a) Sea $G = (V, E)$ un grafo no trivial. Demostrar que G es bipartito si y sólo si existe $I \subseteq V$ tal que I es un conjunto independiente y también un cubrimiento de ejes por vértices.

(b) Diseñar un algoritmo eficiente que dado un grafo $G = (V, E)$, encuentre un mínimo conjunto independiente de G que sea también un cubrimiento de ejes por vértices; si tal conjunto de vértices no existe, el algoritmo debe informarlo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(m+n)$, donde $m = |E|$ y $n = |V|$.

SOLUCIÓN: La solución sería como sigue. Se eliminan de G los vértices aislados, ya que no cubren nada. Supongamos que lo que queda es conexo (caso contrario, repetimos para cada componente conexa). Analizamos si lo que queda es bipartito, y simultáneamente calculamos la partición. Si no es bipartito, por el primer punto no existe solución. Si es bipartito, el conjunto buscado es el subconjunto (de la partición) de cardinalidad menor. ¿Por qué? Tomamos un vértice cualquiera, el cual puede estar o no en la solución. Si está, sus vértices adyacentes no están en la solución (por ser un conjunto independiente). Sin embargo, los adyacentes a estos sí deben estar (para cubrir los ejes que los unen a ellos). Nuevamente los adyacentes a estos no pueden estar, y los adyacentes a estos últimos deben estar, y así sucesivamente. Al final, todos los vértices del lado del primer vértice deben estar en la solución. Si el vértice original no estaba en la solución, de manera análoga todos los vértices del otro subconjunto deben estar en la solución. Resulta entonces que hay sólo dos conjuntos independientes que cubren todos los ejes, concretamente cada uno de los dos subconjuntos de la partición, y la solución es el mínimo de ambos.

NOTA: El ejercicio original era el segundo punto. Yo agregué el primero porque me parece que con eso sale la solución. Habría que verificarlo.

TOMADO: 2002C1R2 07-AGO-2002, 2014C1R2 23-JUL-2014.

360. Para cada uno de los siguientes grafos determinar cuántos subgrafos completos maximales y cuántos subgrafos completos cualesquiera tiene. Expresar los resultados en función de n, m, p y q. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) árbol de n vértices

(e) grafo bipartito conexo de n vértices y m ejes

(f) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(g) $K_{p,q}$

SOLUCIÓN:

K_n: 1 maximal y $2^n - 1$ cualesquiera (la suma de los combinatorios $C(n, i)$ para $i = 1, 2, \ldots n$) ya que cualquier subconjunto no vacío de vértices induce un subgrafo completo.

C_n: $m = n$ maximales (cada eje) y $m + n = 2m = 2n$ cualesquiera (cada eje y cada vértice), salvo para $n = 3$ en cuyo caso hay 1 maximal y 7 cualesquiera.

P_n: Es un caso particular del punto siguiente.

A n: Es un caso particular del punto siguiente, salvo para $n = 1$ en cuyo caso hay 1 de cada uno.

GBC n m: m maximales (cada eje) y $m + n$ cualesquiera (cada eje y cada vértice).

NOTA: Propuesto originalmente por Pablo Heiber. No se incluye ABC h (no aporta).

TOMADO: 2010C2R2 22-DIC-2010 (excepto W_n y $K_{p,q}$).

361. Una clique de máxima frontera de un grafo G, es un subgrafo completo S de G que maximiza la frontera, es decir, la cantidad de ejes que tienen exactamente un extremo en S.

Para cada uno de los siguientes grafos determinar la frontera en una clique de máxima frontera. Expresar los resultados en función de n, p, q y h. Justificar.

(a) K_n

(b) C_n (ciclo simple de $n \geq 3$ vértices)

(c) P_n (camino simple de n vértices)

(d) W_n (ciclo simple de $n \geq 3$ vértices con un vértice universal agregado)

(e) $K_{p,q}$

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

SOLUCIÓN:

K_n: Cualquier subconjunto no vacío de nodos es candidato porque induce un completo. Si el subconjunto tiene $k \geq 1$ nodos su frontera tiene $k(n-k)$ ejes. Basta maximizar eso en función de $k \in \mathbb{N}$. El máximo ocurre en los enteros positivos más cercanos al vértice $x = n/2$ de la parábola. Se puede elegir $k = \lceil n/2 \rceil$, o también $k = \lfloor n/2 \rfloor$ si $n \geq 2$. Como sea, la frontera resulta $\lceil n/2 \rceil \lfloor n/2 \rfloor$.

C_n: Los subgrafos completos son de 1 o 2 nodos, o si $n = 3$ también de 3 nodos. El de 3 nodos tiene frontera 0 que no es máxima, así que lo podemos descartar. Tanto el de 1 como el de 2 nodos tienen la misma frontera, de 2 ejes.

P_n: Los subgrafos completos son de 1 nodo, o si $n \geq 2$ también de 2 nodos. Para $n = 1$ hay que elegir un nodo y la frontera es 0. Para $n = 2$ conviene elegir un nodo y la frontera es 1. Para $n = 3$ conviene elegir el nodo central y la frontera es 2. Para $n \geq 4$ conviene elegir 1 o 2 nodos centrales y la frontera es 2.

$K_{p,q}$: Los subgrafos completos son de 1 nodo o de 2 nodos en partes diferentes. Si se fuera a elegir 1 nodo, conviene que sea de la parte que tiene menos, ya que la frontera es la cantidad de nodos en la otra parte (la que tiene más), es decir, $\max\{p, q\}$. Si se fuera a elegir un nodo de cada parte, cada nodo aporta a la frontera la cantidad de nodos de la otra parte menos 1 por el nodo de la otra parte que también se eligió, es decir, $p + q - 2$. La frontera máxima en definitiva es $\max\{p, q, p + q - 2\}$, que para $p = 1$ o $q = 1$ es $\max\{p, q\} = p + q - 1$ y caso contrario es $p + q - 2$.

NOTA: Basado en el problema de 2013C2L3. No se incluye A n (variable), ni GBC n m (variable).

TOMADO: 2014C2R2 19-DIC-2014 (excepto W_n y ABC h).

362. Una empresa de desarrollo de software acaba de implementar un novedoso sistema con el cual espera aumentar la productividad de sus programadores. Cada programador debe cumplir por día cierta cantidad de horas de trabajo en la empresa, pero es libre de hacerlo en el horario que desee. La cantidad de horas puede ser diferente para cada programador, y no puede fraccionarse en más de un intervalo de tiempo a lo largo del día. El único inconveniente que subsiste con el nuevo sistema es que cuando hay muchos programadores trabajando, ocasionalmente la máquina expendedora de café se queda sin suministros. Para evitar esto es necesario determinar cuál es la mayor cantidad de programadores que están simultáneamente dentro de la empresa. A tal fin se dispone de dos listas, extraidas del sistema de control de ingresos y egresos de la empresa. Una de las listas contiene para cada programador su número de legajo y horario de ingreso a la empresa; en la otra lista los horarios son de egreso. Tanto la lista de ingresos como la de egresos se encuentran ordenadas por horario, ya que esa es la manera en que el sistema de control registra los eventos. Se considera que cada programador está dentro de la empresa desde su horario de ingreso hasta su horario de egreso, incluyendo ambos extremos.

   (a) Modelar el problema de determinar cuál es la mayor cantidad de programadores que están simultáneamente dentro de la empresa como el problema de calcular la cantidad de vértices en un subgrafo completo máximo de cierto grafo. Justificar.

   (b) Diseñar un algoritmo eficiente que resuelva el problema. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad $O(n)$, donde n es la cantidad de programadores de la empresa.

   SUGERENCIA: Usar las listas directamente, sin armar el grafo.

   SOLUCIÓN:

   (a) Armar un grafo con un nodo por cada programador, y un eje por cada solapamiento de los intervalos de permanencia en la empresa. Como dados dos puntos de un intervalo, todo lo del medio también pertenece a él, puede demostrarse que el solapamiento de a pares implica que todos coinciden en algún punto.

   (b) Hacer un merge de las dos listas ordenadas en $O(n)$. Luego recorrer la lista resultante sumando/restando 1 por cada apertura/cierre de intervalo, también en $O(n)$, registrando el máximo valor de ese contador.

   NOTA: Propuesto originalmente por Francisco Soulignac.

   TOMADO: 2007C1P2 10-JUL-2007.

363. Decimos que un camino C es inducido si y sólo si C es un camino simple y el subgrafo inducido por los vértices de C es el propio C. Sea G un grafo no trivial. Sean v y w dos vértices de G tales que todo camino inducido entre ellos tiene una cantidad par de ejes. Sea G_{vw} el grafo que se obtiene eliminando v y w de G, y agregando un vértice vw adyacente a todos los vértices que eran adyacentes a v o w.

   (a) Demostrar que v y w no son adyacentes entre sí.

   (b) Demostrar que $\omega(G) = \omega(G_{vw})$, donde $\omega(\cdot)$ es la cantidad de vértices de un subgrafo completo máximo.

   SOLUCIÓN:

   (a) Si v y w fueran adyacentes, el eje que los une sería un camino inducido de longitud impar.

   (b) Basta ver $\omega(G) \leq \omega(G_{vw})$ y $\omega(G_{vw}) \leq \omega(G)$. Sea H un subgrafo completo máximo de G. Como v y w no son adyacentes, no están ambos en H, de modo que H también es subgrafo de G_{vw}, por lo que $\omega(G) \leq \omega(G_{vw})$. Ahora sea H un subgrafo completo máximo de G_{vw}. Si H no contiene a vw, H es subgrafo de G, por lo que $\omega(G_{vw}) \leq \omega(G)$. Si H contiene a vw, digo que en G todos los otros vértices de H o bien son adyacentes a v, o bien son adyacentes a w. Si fuera cierto, H sería subgrafo de G, por lo que $\omega(G_{vw}) \leq \omega(G)$. Por absurdo, supongamos que en H existen dos vértices x e y (adyacentes entre sí y adyacentes a vw), tales que en G ocurre que x es adyacente sólo a v, mientras que y es adyacente sólo a w. Pero en tal caso, en G habría un camino inducido $v - x - y - w$ de longitud 3 (impar).

NOTA: Propuesto originalmente por Francisco Soulignac. Los vértices v y w se llaman even pair. El grafo G_{vw} se llama contracción. Para volver más difícil el ejercicio se puede suprimir el primer punto.

TOMADO: 2007C1R2 08-AGO-2007.

364. Un grafo se dice perfecto si y sólo si para todo subgrafo inducido H del mismo se cumple $\chi(H) = \omega(H)$, donde $\omega(H)$ es la cantidad de vértices de un subgrafo completo máximo de H. Sea G un grafo.

   (a) Demostrar que si G es perfecto entonces todo subgrafo inducido de G es perfecto.

   (b) Sea v un vértice de G tal que los vértices adyacentes a v forman un subgrafo completo. Demostrar que G es perfecto si y sólo si $G - v$ es perfecto.

   SOLUCIÓN:

   (a) Sea G_1 cualquier subgrafo inducido de G. Sea G_2 cualquier subgrafo inducido de G_1. El grafo G_2 es un subgrafo inducido de G, y como G es perfecto se cumple $\chi(G_2) = \omega(G_2)$, por lo que G_1 es perfecto.

   (b) Para la ida, notar que $G - v$ es un subgrafo inducido de G, y debido al punto anterior es perfecto. Para la vuelta, sea H cualquier subgrafo inducido de G; queremos ver que $\chi(H) = \omega(H)$; sabemos que para cualquier grafo se cumple $\chi(H) \geq \omega(H)$, por lo que para demostrar la igualdad alcanza con mostrar que $\chi(H) \leq \omega(H)$; si H no contiene a v, entonces H es subgrafo inducido de $G-v$, que es perfecto, y por lo tanto $\chi(H) = \omega(H)$; si H contiene a v, consideremos el grafo $H - v$ que es subgrafo inducido de $G - v$, y por lo tanto $\chi(H - v) = \omega(H - v)$; si en H existe una clique máxima que no contiene a v, entonces $\omega(H) = \omega(H - v)$; consideremos un coloreo óptimo de $H - v$; si bien v y sus adyacentes forman una clique en H, esa clique tiene a lo sumo $\omega(H - v) = \chi(H - v)$ nodos, por lo que los nodos adyacentes a v en $H - v$ usan menos de $\chi(H - v)$ colores, y v puede pintarse con un color libre de los usados para pintar $H - v$; de tal modo, $\chi(H) \leq \chi(H - v) = \omega(H - v) = \omega(H)$; por el contrario, si en H toda clique máxima contiene a v, al sacarlo disminuye en una unidad la cantidad de nodos de la clique máxima, es decir, $\omega(H) = \omega(H - v) + 1$, y por lo tanto $\chi(H) \leq \chi(H - v) + 1 = \omega(H - v) + 1 = \omega(H)$.

   NOTA: Propuesto originalmente por Flavia Bonomo.

   TOMADO: 2003C1R2 04-AGO-2003, 2014C2P2 01-DIC-2014.

365. Dado un grafo G, se definen

   - $\omega(G)$ = cantidad de vértices de un subgrafo completo máximo de G;
   - $\chi(G)$ = cantidad mínima de colores con los que es posible pintar los nodos de G de tal manera que nodos adyacentes reciban colores diferentes (es decir, el número cromático de G);
   - $\alpha(G)$ = cantidad de vértices de un conjunto independiente máximo de G; y
   - $k(G)$ = cantidad mínima de subgrafos completos que considerados en conjunto contienen a todos los nodos de G.

   Sea G un grafo.

   (a) Demostrar que $\omega(G) = \alpha(G^c)$.

   (b) Demostrar que $\chi(G) = k(G^c)$.

   (c) Un grafo se dice perfecto si y sólo si para todo subgrafo inducido H del mismo se cumple $\omega(H) = \chi(H)$ y $\alpha(H) = k(H)$. Demostrar que G es perfecto si y sólo si G^c lo es.

   (d) Demostrar que $\omega(G) \leq \chi(G)$.

   (e) Demostrar que $\alpha(G) \leq k(G)$.

   SOLUCIÓN:

   (a) Demostrar la doble desigualdad.

   (b) Demostrar la doble desigualdad.

(c) Usar los dos primeros puntos y que si H es subgrafo inducido de G entonces H^c es subgrafo inducido de G^c. Eventualmente puede probarse que si G es perfecto entonces G^c lo es, y luego aplicar la propiedad a G^c.

(d) Cada nodo de una clique debe recibir un color diferente.

(e) Cada nodo de un conjunto independiente debe ser cubierto por una clique diferente.

NOTA: Basado en lo visto en la materia PGTC.

TOMADO: 2005C2P2 06-DIC-2005 (tres primeros puntos).

366. Dado un grafo G, se definen

- $\nu(G)$ = cantidad de ejes de una correspondencia máxima de G; y
- $\alpha(G)$ = cantidad de vértices de un conjunto independiente máximo de G.

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.

Sea G un grafo con ejes.

(a) Demostrar que $\nu(G) = \alpha(L(G))$.

(b) Es sabido que actualmente no se conocen algoritmos polinomiales para el problema CONJUNTO INDEPENDIENTE MÁXIMO. Teniendo en cuenta el resultado del punto anterior, ¿puede deducirse que no se conocen algoritmos polinomiales para el problema CORRESPONDENCIA MÁXIMA? Justificar.

NOTA: Es parecido al Ejercicio 332.

367. Supongamos que se dispone de un algoritmo que dado un grafo y un entero k, informa en tiempo polinomial si el grafo tiene un

Variante 1: subgrafo completo de k o más

Variante 2: conjunto independiente de k o más

Variante 3: cubrimiento de aristas por vértices de k o menos

vértices. En base a ese algoritmo, diseñar un nuevo algoritmo que dado un grafo G de n vértices, encuentre en tiempo polinomial un

Variante 1: subgrafo completo máximo

Variante 2: conjunto independiente máximo

Variante 3: cubrimiento de aristas por vértices mínimo

de G. El algoritmo propuesto debe invocar al algoritmo original $O(n)$ veces. Mostrar que el algoritmo propuesto es correcto y determinar la cantidad de veces que invoca al algoritmo original. Justificar. La cantidad de veces que el mejor algoritmo que conocemos invoca al algoritmo original es $n + \log_2 n + O(1)$, lo cual es necesario para obtener puntaje máximo en este ejercicio.

SOLUCIÓN: Para la Variante 3, por el Ejercicio 12.13.a de la Práctica, I es un conjunto independiente si y sólo si $R = V(G) - I$ es un cubrimiento de aristas por vértices. Puede demostrarse además que I es máximo si y sólo si R es mínimo. Por lo tanto, basta calcular un conjunto independiente máximo y luego tomar los nodos que no fueron elegidos. Es posible que pueda resolverse también de manera directa, con ideas similares a las que aparecen a continuación para las primeras dos variantes.

Usar búsqueda binaria para encontrar el valor máximo de k. Luego, para cada nodo v de G, si A responde sí en $G - v$, sacar v de G. Al final se obtiene un grafo que sólo tiene al conjunto de nodos buscado (porque A respondió sí para la última versión del grafo). El grafo no puede contener otros nodos ya que A debería haber respondido sí al intentar sacarlos. No se eliminaron/agregaron ejes entre los nodos subsistentes, de modo que cumplen lo pedido en G.

NOTA: Es parecido al Ejercicio 260.

TOMADO: 2015C1P2 04-JUL-2015 (Variante 2), 2018C2R2 19-DIC-2018 (Variante 3).

Flujo en redes

368. Sea S el conjunto de números reales no negativos de la forma $a\,\pi + b$, con $a, b \in \mathbb{Z}$. Sea G una red de flujo. Demostrar que si para todo eje su capacidad pertenece a S, entonces...

    (a) para todo corte su capacidad pertenece a S.
    (b) para todo flujo máximo su valor pertenece a S.
    (c) existe un flujo máximo que asigna a cada eje un valor que pertenece a S.

    ¿Valen las recíprocas? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

    SOLUCIÓN:

    (a) La capacidad de un corte es la suma de las capacidades de ciertos arcos. Esta suma pertenece a S ya que S es cerrado bajo suma y los sumandos pertenecen a S.
    (b) El valor de un flujo máximo coincide con la capacidad de cualquier corte mínimo, que por el punto anterior pertenece a S.
    (c) Se puede demostrar que si se corre el algoritmo de FFEK, en cada paso el flujo en cada arco pertenece a S y la red residual tiene arcos con capacidades que pertenecen a S (por ejemplo, por inducción en el número de paso del algoritmo). Como el algoritmo termina y encuentra un flujo máximo, al finalizar se obtiene un flujo como el requerido.

    Las recíprocas son todas falsas:

    (a) Si la red es C_4 con todas las capacidades 0.5, $d_{\text{in}}(s) = d_{\text{out}}(t) = 0$ y $d_{\text{out}}(s) = d_{\text{in}}(t) = 2$, se cumple que todo corte tiene capacidad $1 \in S$, pero $0.5 \notin S$ (ya que en tal caso $\pi \in \mathbb{Q}$).
    (b) Si la red es $K_{1,2}$ con $d_{\text{in}}(s) = 0$, $d_{\text{out}}(s) = 2$, el arco (s,t) con capacidad 1 y el otro arco con capacidad 0.5, se cumple que el valor de cualquier flujo máximo es $1 \in S$, pero hay un eje de capacidad $0.5 \notin S$.
    (c) Si la instancia es la misma que en el punto anterior, se cumple que existe un flujo máximo que asigna 1 al arco (s,t), y 0 al otro arco, ambos valores en S, pero hay un eje de capacidad $0.5 \notin S$.

    NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma. Para volver más fácil el ejercicio se puede definir $S = \{a\,\pi\}$, $S = \mathbb{Q}$, $S = \mathbb{Z}$, etc. Aparentemente las idas se cumplen siempre que S sea un grupo.

    TOMADO: 2017C1P2 07-JUL-2017.

369. Sea G una red donde cada arco e tiene asociada una capacidad $c(e) > 0$. Demostrar que el valor del flujo máximo en G es 0 si y sólo si G no tiene caminos dirigidos de la fuente al sumidero.

    SOLUCIÓN: Para la ida, por dualidad existe un corte mínimo S de capacidad 0. Como las capacidades son todas positivas, significa que no hay arcos de S a $V - S$. Eso implica que no hay caminos dirigidos de la fuente al sumidero, porque tales caminos deberían pasar de S a $V - S$ ya que la fuente está en S y el sumidero en $V - S$. Alternativamente, podemos demostrar la contrarecíproca. Supongamos que G tiene un camino dirigido de la fuente al sumidero. Si corremos el algoritmo de FF o FFEK con un flujo inicial nulo, en la red residual ese camino existe porque las capacidades son todas positivas. Por lo tanto el algoritmo va a encontrar al menos un camino de aumento, y va a aumentar el valor del flujo a un valor positivo. Como el flujo máximo tiene un valor mayor o igual que el de cualquier flujo, el valor del flujo máximo también es positivo.

    Para la vuelta, sea S el conjunto de todos los nodos alcanzables desde la fuente. Por definición, la fuente está en S, mientras que el sumidero está en $V - S$ ya que no hay caminos dirigidos de la fuente al sumidero. De tal modo, S es un corte. Además, no hay arcos de S a $V - S$, porque en tal caso habría en $V - S$ nodos alcanzables desde la fuente, contradiciendo la definición de S. Eso implica que la capacidad de S es 0, y por lo tanto es un corte de capacidad mínima. Por dualidad, el valor del flujo máximo también es 0. Alternativamente, si corremos el algoritmo de FF o FFEK con un flujo inicial nulo, en la red residual no hay ningún camino de aumento, por lo que el algoritmo termina de inmediato con un flujo máximo de valor 0.

    NOTA: Propuesto originalmente por Guido Tagliavini Ponce.

    TOMADO: 2016C1R2 15-JUL-2016.

370. Sea $F \in \mathbb{N}_0$, y sea G una red donde cada arco e tiene asociada una capacidad $c(e) \in \mathbb{N}_0$. Diseñar un algoritmo eficiente que permita encontrar en G un flujo de valor exactamente F si la red lo admite, o que informe que tal flujo no existe en caso contrario. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

 SOLUCIÓN: Se puede demostrar que hay un flujo de valor F si y sólo si F es menor o igual que el valor del flujo máximo. Hay al menos tres formar de encontrar un flujo de valor F. Una es correr FFEK, y en cuanto el valor del flujo quedaría mayor o igual que F, subirlo sólo hasta F. Otra es correr el algoritmo completo y al final multiplicar en flujo en todos los arcos por F y dividir por el valor del flujo máximo. La tercera es agregar un arco de capacidad F desde una nueva fuente hacia la original, o desde el sumidero original hacia uno nuevo, y correr FFEK; hay flujo de valor F si y sólo si el arco agregado está saturado, y en tal caso el flujo en el resto de los arcos es el buscado.

 NOTA: Basado en una pregunta de un alumno.

 TOMADO: 2010C2R2 22-DIC-2010, 2019C1R2 19-JUL-2019.

371.

Variante 1: Sea G una red donde cada arco tiene una capacidad y un flujo válido asignados. Diseñar un algoritmo eficiente que decida si el flujo es máximo.

Variante 2: Sea G un grafo bipartito, y sea M una correspondencia de G. Diseñar un algoritmo eficiente que decida si M es máxima.

 Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. El mejor algoritmo que conocemos tiene complejidad coincidente con la de BFS, lo cual es necesario para obtener puntaje máximo en este ejercicio.

 SOLUCIÓN: Para la Variante 2 armar una red agregando una fuente, un sumidero, con todos los arcos de capacidad 1, y un flujo de 1 en cada arco de la correspondencia. En ambos casos hacer una iteración del algoritmo para calcular flujo máximo para decidir si hay algún camino de aumento. Era máximo si y sólo si no hay caminos de aumento.

 NOTA: Encontrado originalmente en Internet.

 TOMADO: 2008C1R2 25-JUL-2008 (Variante 1), 2009C1P2 10-AGO-2009 (Variante 2), 2015C1R2 17-JUL-2015 (Variante 2).

372. Supongamos que luego de resolver un problema de flujo máximo para una red determinada, se descubre que la capacidad de uno de los arcos de la red era incorrecta. ¿Es necesario recomenzar todo el trabajo desde el principio, o existe alguna forma de aprovechar los resultados obtenidos? Justificar.

 SOLUCIÓN: Si la capacidad aumentó, se puede aplicar FFEK, usando como flujo inicial el flujo máximo que se tenía antes de cambiar la capacidad; sin embargo, si el eje no estaba saturado, no pertenecía al corte mínimo, por lo que al cambiar su capacidad el corte mínimo no cambia, y por lo tanto tampoco el flujo máximo. Si la capacidad disminuyó, pero el flujo que había en el eje es menor o igual que la nueva capacidad, tampoco es necesario hacer nada; si en cambio el flujo pasa a ser mayor, una posibilidad es multiplicar los flujos de todos los ejes por determinada constante menor que 1, de modo tal de obtener un flujo factible, y usarlo como solución inicial en el algoritmo mencionado; también podrían buscarse caminos de disminución desde los extremos del eje que cambió de capacidad, pero en el peor caso habría que buscar muchos, y parece más engorroso que multiplicar los flujos de todos los ejes.

 TOMADO: 2001C2R2 26-DIC-2001.

373. Sea $G = (V, E)$ una red donde cada arco $e \in E$ tiene asociadas dos capacidades $c_1(e)$ y $c_2(e)$. Para $i \in \{1, 2\}$, sea $\alpha_i \in \mathbb{R}_{\geq 0}$ y sea f_i un flujo válido para la capacidad c_i. Para cada arco $e \in E$ definimos $f(e) = \alpha_1 f_1(e) + \alpha_2 f_2(e)$ y $c(e) = \alpha_1 c_1(e) + \alpha_2 c_2(e)$.

 (a) Demostrar que c es una capacidad válida y que f es un flujo válido para ella.

 (b) ¿Es cierto que si f_1 y f_2 son flujos máximos entonces f también lo es? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

 SOLUCIÓN:

(a) Hay que demostrar que para cada arco e se cumple $0 \leq f(e) \leq c(e)$, y que f cumple la ley de conservación en todos los nodos salvo la fuente y el sumidero. Todo eso se puede probar usando que $\alpha_i \geq 0$ y que f_i es válido.

(b) No es cierto. Un caso donde puede ocurrir es cuando f_1 satura un arco pero no otro, f_2 a la inversa, y el flujo combinado no satura ninguno. Por ejemplo si la red es de la forma $s \to v \to t$, con $c_1(s,v) = 1$, $c_1(v,t) = 2$, $\alpha_1 = 1$, $f_1(s,v) = 1$, $f_1(v,t) = 1$, $c_2(s,v) = 2$, $c_2(v,t) = 1$, $\alpha_2 = 1$, $f_2(s,v) = 1$, $f_2(v,t) = 1$. Resulta $c(s,v) = 3$, $c(v,t) = 3$, $f(s,v) = 2$, $f(v,t) = 2$. Los flujos f_1 y f_2 son máximos, pero f no lo es.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2014C2R2 19-DIC-2014.

374. Sea $G = (V,E)$ una red con capacidades en los arcos. Sean f_1 y f_2 dos flujos válidos en G. Sea $\alpha \in [0,1]$. Definimos la función $f : E \to \mathbb{R}$ como $f(e) = \alpha f_1(e) + (1-\alpha) f_2(e)$.

(a) Demostrar que f es un flujo válido en G.

(b) ¿Es cierto que si f_1 y f_2 son flujos máximos entonces f también lo es? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

SOLUCIÓN:

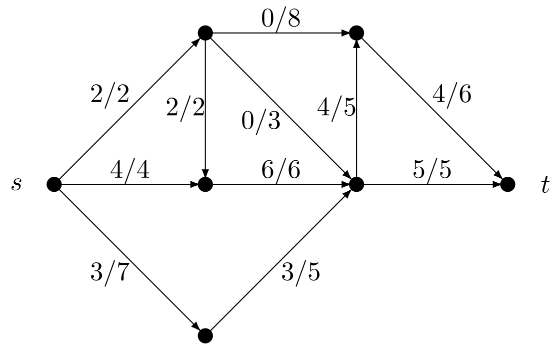
(a) Hay que demostrar que f toma para cada arco un valor entre 0 y la capacidad del arco, y que es conservativo en todos los nodos salvo la fuente y el sumidero. Todo eso se puede probar usando que $\alpha \in [0,1]$ y que vale para f_1 y f_2.

(b) Es cierto. Usando la definición de valor de un flujo y que $f = \alpha f_1 + (1-\alpha) f_2$, se puede demostrar que el valor de f es la misma combinación lineal de los valores de f_1 y f_2. Como esos valores son máximos y por lo tanto iguales, el valor de f resulta igual al valor de un flujo máximo.

NOTA: Propuesto originalmente por Esteban Feuerstein. Es un caso particular del Ejercicio 373, tomando $\alpha_1 = \alpha$, $\alpha_2 = 1 - \alpha$ y $c_1 = c_2$.

TOMADO: 2003C1P2 11-JUL-2003 (primer punto), 2011C2R2 21-DIC-2011 (primer punto), 2017C2P2 01-DIC-2017.

375. Sea la siguiente red donde cada arco tiene un flujo y una capacidad asignados.



(a) Encontrar un camino de aumento que produzca el mayor aumento posible en el flujo. Justificar.

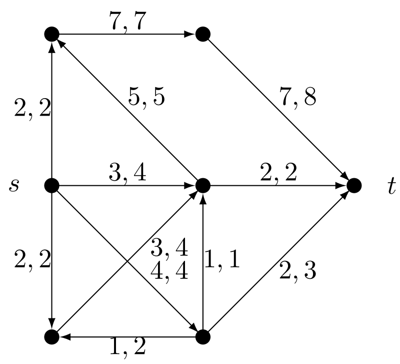
(b) Utilizar el algoritmo de Ford-Fulkerson para obtener un flujo máximo a partir del flujo inicial. Justificar.

(c) Indicar un corte de capacidad mínima. Justificar.

SOLUCIÓN: Puede hallarse un camino que aumenta el flujo en 2, el cual produce el mayor aumento posible ya que satura los arcos que entran a t.

TOMADO: 2004C1R2 09-AGO-2004 (primer punto), 2007C1R2 08-AGO-2007 (últimos dos puntos), 2014C1R2 23-JUL-2014 (primer punto).

376. Sea la siguiente red donde cada arco tiene un flujo y una capacidad asignados.



(a) Encontrar un camino de aumento que produzca el mayor aumento posible en el flujo. Justificar.

(b) Utilizar el algoritmo de Ford-Fulkerson para obtener un flujo máximo a partir del flujo inicial. Justificar.

(c) Indicar un corte de capacidad mínima. Justificar.

TOMADO: 2005C1P2 15-JUL-2005 (últimos dos puntos), 2018C2R2 19-DIC-2018 (últimos dos puntos).

377. Sean $p = (p_1, p_2, \dots p_r) \in \mathbb{N}^r$ y $q = (q_1, q_2, \dots q_s) \in \mathbb{N}^s$. Diseñar un algoritmo eficiente basado en grafos que decida si existe $M \in \{0,1\}^{r \times s}$ tal que p_i es la suma de la i-ésima fila de M y q_j es la suma de la j-ésima columna de M. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Armar una red con una fuente, un nodo por cada fila, un nodo por cada columna, y un sumidero. Poner un arco de capacidad p_i desde la fuente a la i-ésima fila, un arco de capacidad 1 desde cada fila a cada columna, un arco de capacidad q_j desde la j-ésima columna al sumidero. Buscar el flujo máximo en esta red y determinar si tal flujo resulta ser la sumatoria de los p_i y también la sumatoria de los q_j. Los arcos centrales que tengan flujo no nulo indican las posiciones no nulas de M. El valor del flujo máximo está acotado por rs ya que hay un corte con esa capacidad. Por lo tanto, ya sea que se use FF o FFEK la cantidad de iteraciones es $O(rs)$, de modo que el costo de encontrar el flujo máximo es $O(mrs) = O(r^2s^2)$ si la red está representada con listas de sucesores. El costo de armar la red y la verificación final no superan a eso.

TOMADO: 2001C1P2 10-JUL-2001, 2015C2P2 30-NOV-2015.

378. Sean d_{out} y d_{in} dos vectores, cada uno de ellos formado por n enteros no negativos. Diseñar un algoritmo eficiente basado en flujo que encuentre un digrafo simple tal que el i-ésimo vértice tenga grado de salida $d_{\text{out}}(i)$ y grado de entrada $d_{\text{in}}(i)$ para $i = 1, 2, \dots n$. El digrafo buscado no debe tener arcos de un vértice a sí mismo, ni más de un arco con igual dirección entre el mismo par de vértices (aunque sí puede tener dos arcos con direcciones opuestas). Si un digrafo como el descripto no existe, el algoritmo debe informarlo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SUGERENCIA: Definir una red con una fuente, un sumidero y $2n$ vértices adicionales.

SOLUCIÓN: Podemos representar al digrafo con una matriz que tiene un 1 en la posición i, j si y sólo si hay un arco del i-ésimo vértice al j-ésimo. Lo que se pregunta entonces es si existe una matriz binaria, con diagonal nula, cuyas filas suman d_{out} y cuyas columnas suman d_{in}. Visto de esa forma el ejercicio es un caso particular del Ejercicio 377, definiendo $p = d_{\text{out}}$ y $q = d_{\text{in}}$, con la restricción adicional de la diagonal nula. Se resuelve casi como ese ejercicio, poniendo un nodo sucesor de la fuente por cada $d_{\text{out}}(i)$ conectado por un arco de capacidad igual a ese valor, un nodo sucesor de cada uno de esos por cada $d_{\text{in}}(i)$ conectado por un arco de capacidad 1, y un sumidero sucesor de cada uno de esos conectado por un arco de capacidad igual al valor del predecesor. A fin de evitar self-loops, se suprimen los arcos de $d_{\text{out}}(i)$ a $d_{\text{in}}(i)$ para un mismo i. Luego se calcula flujo máximo, cuyo valor debe ser igual tanto a la suma de d_{out} como a la suma de d_{in} (que por lo tanto deben ser iguales entre sí). Los arcos centrales con flujo no nulo (y por lo tanto igual a 1) indican las adyacencias en el digrafo. En el otro ejercicio los enteros son positivos y acá pueden ser nulos. Eso probablemente es porque en el otro ejercicio un valor nulo permite eliminar una fila o columna de la matriz, lo que reduce el problema. En este ejercicio un valor nulo no permite eliminar un nodo del digrafo (aunque sí se puede eliminar un nodo de la red ya que no va a pasar flujo por él, lo que equivale a eliminar una fila o columna de la matriz que representa al digrafo).

NOTA: Propuesto originalmente por Michel Mizrahi.

TOMADO: 2012C1P2 11-JUL-2012, 2018C2P2 03-DIC-2018.

379. Sea $A \in \mathbb{R}^{n\times n}$. Diseñar un algoritmo eficiente basado en grafos que encuentre $B \in \mathbb{R}^{n\times n}$ sin ceros en su diagonal, obtenida de permutar las filas de A; si tal matriz B no existe, el algoritmo debe informarlo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Armar un grafo bipartito con un nodo por cada fila de A y uno por cada fila de B. El nodo correspondiente a la fila i de A es adyacente al nodo correspondiente a la fila j de B si y sólo si la fila i de A no tiene un cero en la columna j (es decir, puede ser la fila j de B). Lo que queremos es ver si una correspondencia máxima (que va a ser perfecta) en este grafo tiene cardinalidad n. Para eso es suficiente con armar la red de flujo habitual para correspondencia máxima en un grafo bipartito, y calcular el valor del flujo máximo. La correspondencia está definida por los arcos entre filas que tienen flujo no nulo, y eso indica qué fila de A hay que ubicar en cada fila de B.

TOMADO: 2001C1R2 06-AGO-2001, 2016C1P2 02-JUL-2016.

380. Cada persona tiene $O(1)$ nombres, cada uno de ellos de largo $O(1)$. Aunque una persona tenga varios nombres, normalmente se elige un solo nombre para referirse a la persona. Si dos personas comparten nombres, es posible elegir el mismo nombre para ambas, lo cual produciría ambigüedad.

(a) Dado un grupo de p personas junto con la lista de nombres de cada una, diseñar un algoritmo eficiente basado en grafos que elija un nombre de cada persona de manera tal que no haya dos personas para las cuales se elija el mismo nombre. Si tal elección no es posible, el algoritmo debe informarlo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

(b) Aplicar el algoritmo al siguiente grupo de $p = 4$ personas: Armando Esteban Quico, Quico Ken Tucky, Ken, Clark Ken.

SOLUCIÓN: Es un problema de correspondencia máxima en un grafo bipartito, que se puede resolver con flujo. Armar un digrafo con una fuente, un nodo para cada persona, un nodo para cada nombre distinto, y un sumidero. Poner un arco de la fuente a cada persona, de cada nombre al sumidero, y de cada persona a cada uno de sus nombres. Todos los arcos tienen capacidad 1. Buscar un flujo máximo en este grafo. Si es p es posible la asignación, y los ejes entre personas y nombres que tengan flujo no nulo indican una tal asignación.

Como cada persona tiene $O(1)$ nombres hay $O(p)$ nombres distintos, de modo que la red tiene $O(p)$ nodos y ejes. Como cada nombre tiene largo $O(1)$, es posible asignar identificadores a los nombres con complejidad $O(p)$ si se utiliza un diccionario implementado sobre un trie. Otros diccionarios menos eficientes también son viables. Si la red se representa con listas de sucesores es posible armarla con complejidad $O(p)$. La red resulta conexa y con capacidades enteras, de modo que la cantidad de iteraciones de FF o su caso particular FFEK está acotada por el valor del flujo máximo, que es a lo sumo p ya que hay un corte de esa capacidad. El costo de cada iteración es $O(m) = O(p)$, por lo que obtener el flujo máximo cuesta $O(p^2)$. Determinar si el flujo máximo vale p se hace en $O(1)$, y encontrar la asignación de nombres demora $O(m) = O(p)$. La complejidad total es $O(p^2)$.

TOMADO: 2008C1P2 11-JUL-2008 (con otros nombres), 2018C1P2 06-JUL-2018.

381. El Magnate de los Celulares tiene un negocio de venta de (teléfonos) celulares y accesorios en el microcentro porteño. Entre los accesorios disponibles se encuentran las fundas para celulares. Cada celular y cada funda tienen tamaños específicos. Para que un celular pueda ser guardado dentro de una funda, la funda debe ser más grande que el celular, pero no excesivamente grande a fin de que el celular no se sacuda dentro de la funda. Concretamente, un celular de tamaño c puede ser guardado dentro de una funda de tamaño f si y sólo si $c < f < 1.1c$. Dado que un celular enfundado se vende más caro que el celular y la funda por separado, El Magnate quiere saber cuál es la máxima cantidad de celulares que puede enfundar. Diseñar un algoritmo eficiente basado en grafos que determine esa cantidad. La entrada del algoritmo es

Variante 1: la cantidad de celulares, la cantidad de fundas, y los tamaños de cada uno de esos elementos.

Variante 2: la cantidad de modelos de celulares, la cantidad de modelos de fundas, y para cada modelo su tamaño junto con la cantidad de unidades en stock.

Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN:

Variante 1: Lo que se busca es la cardinalidad de una correspondencia máxima en un grafo bipartito en el que cada celular y cada funda es un vértice, y hay un eje entre un celular y una funda si y sólo si la funda es apropiada para el celular. Dicha cardinalidad puede calcularse como el valor del flujo máximo en la red asociada al grafo bipartito. La red se obtiene agregando una fuente, un sumidero, arcos de la fuente a cada vértice de V_1, arcos de cada vértice de V_2 al sumidero, y dirigiendo los ejes que hay entre V_1 y V_2 desde V_1 hacia V_2, con todos los arcos de capacidad 1. Es indistinto si V_1 contiene a los celulares y V_2 a las fundas, o a la inversa. Si la red se representa con listas de sucesores, armarla tiene complejidad $O(pq)$, donde p es la cantidad de celulares y q es la cantidad de fundas. El valor del flujo máximo puede calcularse aplicando FFEK, cuya complejidad es la cantidad de iteraciones multiplicada por el costo de cada iteración. Como el flujo es entero en cada paso del algoritmo, la cantidad de iteraciones está acotada por el valor del flujo máximo, que es $O(\min\{p,q\})$, mientras que el costo de cada iteración es $O(|E|+|V|) = O(pq+p+q) = O(pq)$ para la representación indicada. La complejidad total es entonces $O(\min\{p,q\}pq)$.

Variante 2: Armar una red similar con un nodo por cada modelo, capacidades infinitas entre fundas y celulares, y capacidad coincidente con el stock del modelo para cada arco incidente sobre el modelo y la fuente o el sumidero. El costo de armar la red es el mismo que antes, el costo de cada iteración también, y la cantidad de iteraciones es $O(|E| \times |V|) = O(pq(p+q))$. La complejidad total es entonces $O(p^2q^2(p+q))$.

NOTA: Propuesto originalmente por Michel Mizrahi.

TOMADO: 2013C1P2 10-JUL-2013 (Variante 1), 2013C1R2 05-AGO-2013 (Variante 2).

382. Gracias al ejercicio del primer parcial, El Magnate de la Bondiola pudo instalar p puestos en la franja de la Costanera que está al sur de Ciudad Universitaria. En el i-ésimo puesto están trabajando a_i empleados ($i = 1, 2, \dots p$). Pero los beneficios no son los esperados, y el empresario lo atribuye a que los empleados no saben preparar la bondiola como se debe. Para solucionar esto, El Magnate (de la Bondiola) ha decidido que los empleados asistan a q cursos de especialización en distintos países europeos. El j-ésimo curso tiene cupo para b_j nuevos asistentes ($j = 1, 2, \dots q$). Todos los cursos ocurren simultáneamente, de modo que no es posible que un mismo empleado asista a más de un curso. Cuando terminen los cursos, cada empleado que haya asistido transmitirá sus nuevos conocimientos a sus compañeros de puesto, de modo que no puede haber dos empleados del mismo puesto que asistan al mismo curso. Diseñar un algoritmo eficiente basado en grafos que determine el mayor número de empleados que pueden asistir a los cursos. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. Aplicar el algoritmo al caso particular $p = 7$, $q = 5$, $a_i = 3$, $b = (6, 4, 5, 4, 3)$.

SOLUCIÓN: Ver Ejercicio 383, salvo que alcanza con calcular el flujo máximo.

NOTA: Los Ejercicios 382, 383, 384, 385 y 386 son similares o iguales.

TOMADO: 2010C2P2 06-DIC-2010, 2017C2R2 18-DIC-2017.

383.

Variante 1: Hoy se casan Fausto y Filomena.

Variante 2: Fausto y Filomena se casaron el 2 de diciembre de 2013. Luego se divorciaron, y hoy se casan nuevamente.

A la fiesta de casamiento concurren p familias. Sea a_i el número de miembros de la i-ésima familia ($i = 1, 2, \dots p$). Por otra parte, se dispone de q mesas donde ubicar a los concurrentes. Sea b_j la cantidad de personas que caben en la j-ésima mesa ($j = 1, 2, \dots q$). Fausto y Filomena desean que en cada mesa haya a lo sumo

Variante 1: un miembro

Variante 2: dos miembros

de cada familia. Diseñar un algoritmo eficiente basado en grafos que ubique a los concurrentes en las mesas de manera tal que se cumpla esa regla. Si no es posible ubicar a los concurrentes, el algoritmo debe informarlo. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Ver Ejercicio 384, renombrando a a b y b a c, y asignando capacidad 1 o 2 a los arcos intermedios, dependiendo de la variante.

NOTA: Propuesto originalmente por Esteban Feuerstein (Variante 1). Los Ejercicios 382, 383, 384, 385 y 386 son similares o iguales.

TOMADO: 2004C1P2 14-JUL-2004 (Variante 1), 2013C2P2 02-DIC-2013 (Variante 1), 2019C1P2 05-JUL-2019 (Variante 2).

384. Es necesario transportar p productos utilizando q camiones a través de una ruta insegura. Hay b_i bolsas del i-ésimo producto y el j-ésimo camión puede transportar c_j bolsas ($i = 1, 2, \ldots p$; $j = 1, 2, \ldots q$). A fin de aumentar las posibilidades de que alguna bolsa de cada producto llegue a destino, ningún camión debe llevar más de una bolsa de cada producto. Diseñar un algoritmo eficiente basado en grafos que decida si es posible realizar el transporte requerido. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar. Aplicar el algoritmo al caso particular $p = 7$, $q = 5$, $b_i = 3$, $c = (6, 4, 5, 4, 3)$.

SOLUCIÓN: Armar un digrafo con una fuente, un nodo por cada tipo de producto, un nodo por cada camión, y un sumidero. Poner un arco de capacidad b_i desde la fuente al i-ésimo producto, un arco de capacidad 1 desde cada producto a cada camión, un arco de capacidad c_j desde el j-ésimo camión al sumidero. Buscar el flujo máximo en este grafo y determinar si tal flujo resulta ser la sumatoria de los b_i. Los arcos centrales que tengan flujo no nulo indican qué bolsas llevar en cada camión. El valor del flujo máximo está acotado por pq ya que hay un corte con esa capacidad. Por lo tanto, ya sea que se use FF o FFEK la cantidad de iteraciones es $O(pq)$, de modo que el costo de encontrar el flujo máximo es $O(mpq) = O(p^2q^2)$.

NOTA: Los Ejercicios 382, 383, 384, 385 y 386 son similares o iguales.

TOMADO: 2001C2P2 11-DIC-2001, 2016C2R2 19-DIC-2016.

385.

Variante 1: Usted es el jefe del comité de programa de una importante conferencia internacional sobre ciencias de la computación. Diferentes autores han enviado un total de t trabajos de investigación que desean presentar en la conferencia. Para seleccionar los mejores trabajos, usted dispone de la ayuda de r revisores, cada uno de los cuales ha elegido hasta 20 trabajos que estaría dispuesto a examinar. Cada trabajo puede ser examinado por hasta 5 revisores distintos, y cada revisor puede examinar hasta 10 trabajos distintos. Sin embargo, un revisor sólo puede examinar trabajos que él haya indicado que estaría dispuesto a examinar.

Diseñar un algoritmo eficiente basado en grafos que determine la máxima cantidad total de opiniones que es posible obtener por parte de los revisores. La entrada del algoritmo es la cantidad t de trabajos, la cantidad r de revisores, y para cada revisor los trabajos que estaría dispuesto a examinar.

Variante 2: Se tienen m materiales cuyas propiedades es necesario analizar. Para llevar a cabo esta tarea, se han seleccionado b laboratorios especializados en este tipo de pruebas. Cada material se identifica por un entero distinto entre 1 y m. Cada laboratorio se identifica por un entero distinto entre 1 y b, y está capacitado para analizar sólo algunos de los materiales existentes. A fin de limitar el efecto que pudiera tener algún laboratorio incompetente, está previsto que un mismo material sea analizado por hasta 5 laboratorios distintos, y que cada laboratorio analice hasta 10 materiales distintos.

Diseñar un algoritmo eficiente basado en grafos que determine la máxima cantidad total de informes sobre los materiales que es posible obtener por parte de los laboratorios. La entrada del algoritmo es la cantidad m de materiales, la cantidad b de laboratorios, y para cada laboratorio la lista de materiales que el laboratorio es capaz de analizar.

Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Lo que sigue es en los términos de la Variante 2, pero lo mismo vale para la Variante 1. Armar un digrafo con una fuente, un nodo por cada material, un nodo por cada laboratorio, y un sumidero. Poner un arco de capacidad 5 desde la fuente a cada material, un arco de capacidad 1 desde cada material a cada laboratorio que puede analizarlo, un arco de capacidad 10 desde cada laboratorio al sumidero. Buscar el flujo máximo. Los arcos con flujo positivo definen las pruebas a realizar.

NOTA: Propuesto originalmente por Esteban Feuerstein (Variante 1). Los Ejercicios 382, 383, 384, 385 y 386 son similares o iguales.

TOMADO: 2003C1R2 04-AGO-2003 (Variante 2), 2005C2P2 06-DIC-2005 (Variante 1), 2015C2R2 18-DIC-2015 (Variante 1), 2018C1R2 20-JUL-2018 (Variante 2).

386. Peligra la corrección del examen de hoy. Algunos docentes no corrigen, otros pueden corregir más de un ejercicio, y cada docente que corrige sólo está dispuesto a corregir algunos de los ejercicios que le parecen lindos. Son tantas restricciones que no sabemos cómo organizarlo. Para poder corregir el examen, necesitamos que alguien nos revele cómo repartir los ejercicios entre los docentes.

 Hay d docentes que corrigen, y e ejercicios para corregir. Cada docente se identifica por un entero distinto entre 1 y d. Cada ejercicio se identifica por un entero distinto entre 1 y e.

 Diseñar un algoritmo eficiente basado en grafos que decida si es posible corregir todos los ejercicios respetando las restricciones mencionadas, y en tal caso indique una posible distribución de los ejercicios entre los docentes. La entrada del algoritmo es la cantidad d de docentes, la cantidad e de ejercicios, y para cada docente la lista de ejercicios que le parecen lindos y la cantidad máxima de ejercicios que puede corregir. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

 SOLUCIÓN: La solución es muy parecida a la del Ejercicio 385. Los ejercicios son los materiales, y los docentes son los laboratorios. Cada ejercicio es analizado por un único docente (en vez de a lo sumo 5), y cada docente tiene una cantidad máxima variable de ejercicios que puede corregir (en vez de 10). El objetivo es maximizar la cantidad de ejercicios corregidos, que en este caso deben ser todos para que la corrección sea posible.

 NOTA: Propuesto originalmente por Brian Bokser. Los Ejercicios 382, 383, 384, 385 y 386 son similares o iguales.

 TOMADO: 2017C1R2 21-JUL-2017.

387. En un colegio cada materia está a cargo de un único profesor, cada profesor tiene una o más materias a cargo, y cada alumno cursa una o más materias. Los resultados a fin de año no son muy buenos, ya que cada materia tiene muchos alumnos que por las notas que tienen deberían desaprobarla. Para no dar una mala imagen y quedarse sin alumnos, las autoridades del colegio impusieron ciertas restricciones que deben cumplirse. El i-ésimo profesor no puede desaprobar a más de $x_i > 0$ alumnos en total entre todas las materias a su cargo; caso contrario es tildado de "profesor excesivamente severo" y no recibe su bono navideño. La i-ésima materia no puede tener más de $y_i > 0$ alumnos desaprobados; caso contrario es tildada de "materia excesivamente difícil" y sacada del plan de estudios. Por último, el i-ésimo alumno no puede desaprobar más de $z_i > 0$ materias; caso contrario queda libre. Dada la lista de alumnos que deberían desaprobar las distintas materias, las autoridades del colegio quieren regalar nota para que todos los profesores reciban su bono navideño, ninguna materia sea sacada del plan de estudios, y ningún alumno quede libre. Esto siempre puede lograrse regalando nota a todos los alumnos que deberían desaprobar cada materia. Sin embargo, para que el colegio no parezca poco serio, las autoridades quieren que la cantidad total de regalos de nota sea la menor posible. Diseñar un algoritmo eficiente basado en grafos que determine a cuáles alumnos de cada materia regalarles nota a fin de que se cumpla el objetivo de las autoridades del colegio. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

 SUGERENCIA: Minimizar la cantidad de regalos de nota equivale a maximizar la cantidad de desaprobados.

 SOLUCIÓN: Armar un digrafo con una fuente, un nodo por cada profesor, un nodo por cada materia, un nodo por cada alumno que debería desaprobar alguna materia, y un sumidero. Poner un arco desde la fuente al i-ésimo profesor con capacidad x_i, un arco desde el profesor que tiene a cargo la i-ésima materia hacia la misma con capacidad y_i, un arco desde cada materia hacia cada alumno que debería desaprobarla con capacidad 1, un arco desde el i-ésimo alumno hacia el sumidero con capacidad z_i. Las capacidades indican la cantidad máxima de desaprobados de acuerdo a las restricciones (las capacidades de los arcos de materias a alumnos indican que cada alumno puede desaprobar a lo sumo 1 vez cada materia). Calcular el flujo máximo en la red, es decir, la cantidad máxima de desaprobados en total. Los arcos de materias a alumnos que resulten con flujo positivo indican los alumnos que deben desaprobar, mientras que los que resulten con flujo nulo indican los regalos de nota. La complejidad es la de armar el grafo (que depende del formato de entrada), más la del algoritmo de flujo (por ejemplo FFEK).

NOTA: Propuesto originalmente por Marina Groshaus.

TOMADO: 2012C2P2 05-DIC-2012.

388. Según surge indubitablemente de un reciente mail anónimo, los alumnos que no aprueben este examen piensan sabotear el acceso a Internet del Departamento de Computación (DC). El objetivo de este ejercicio es favorecer la aprobación de los alumnos que tienen el conocimiento suficiente como para llevar a cabo ese plan, y de esa manera evitar que se realice.

La red del DC está formada por s computadoras y está contenida en una red más grande formada por n computadoras ($0 < s < n$). Cada computadora se identifica por un entero distinto entre 1 y n, siendo $1, 2, \ldots s$ las computadoras del DC y n el servidor que permite el acceso a Internet de la red general. Esta red contiene también m cables que transmiten información. Cada cable conecta en forma directa determinado par de computadoras. Una computadora dada p tiene acceso a Internet si y sólo si existe una sucesión de computadoras $p_1, p_2, \ldots p_k$ tal que $p_1 = p$, $p_k = n$ y hay un cable entre p_i y p_{i+1} para $1, 2, \ldots k-1$. Los saboteadores planean cortar cables, no necesariemente dentro de la red del DC, de manera tal que ninguna computadora del DC tenga acceso a Internet. Como localizar cada cable es dificultoso, quieren cortar la mínima cantidad posible.

Diseñar un algoritmo eficiente basado en grafos que determine la mínima cantidad de cables tal que al ser cortados producen que ninguna computadora del DC tenga acceso a Internet. La entrada del algoritmo es la cantidad s de computadoras en la red del DC, la cantidad n de computadoras en la red general, la cantidad m de cables en la red general, y para cada cable las computadoras que conecta. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Se puede argumentar que nunca es necesario cortar cables entre computadores del DC. Armar un grafo con un nodo por computadora y un eje por cada cable, contraer todas las computadoras del DC en un único nodo (puede quedar un multigrafo), asignar capacidad 1 a cada eje, y calcular un corte mínimo (o flujo máximo) desde el nodo DC hacia el nodo n. Habría que ver que lo de conectividad se aplica a multigrafos, o talvez agregar un nodo intermedio en cada eje múltiple, lo cual no cambia el corte mínimo (porque si se agrega un nodo x en el medio de un eje (v, w), alcanza con cortar (v, x) o (x, w) para desconectar v de w).

NOTA: Propuesto originalmente por Michel Mizrahi.

TOMADO: 2013C2R2 21-DIC-2013.

389. En nuestro deporte favorito dos equipos juegan en una cancha desplazando una pelota dentro de la misma. Cada equipo tiene un arco y un jugador especial llamado arquero que permanece cerca de su arco. Nuestro equipo está formado por n jugadores, identificados por enteros distintos entre 1 y n; el número 1 identifica a nuestro arquero. Una jugada de gol es una sucesión de movimientos de la pelota dentro de la cancha, donde el primer movimiento (llamado saque de arco) lo realiza el arquero de un equipo, y el último movimiento (llamado gol) produce que la pelota entre en el arco del equipo contrario. Los movimientos de la pelota que no son el gol mismo son pases de la pelota entre jugadores de un mismo equipo. Los jugadores ocupan posiciones relativamente fijas dentro de la cancha, de modo que cada jugador puede pasarle la pelota sólo a ciertos jugadores de su equipo, y no necesariamente todos los jugadores pueden tirar la pelota al arco contrario. Decimos que dos jugadas de gol son independientes si y sólo si no contienen a un mismo movimiento de la pelota.

Diseñar un algoritmo eficiente basado en grafos que determine la máxima cantidad de jugadas de gol independientes de a pares que puede realizar nuestro equipo. La entrada del algoritmo es la cantidad n de jugadores de nuestro equipo, y para cada jugador la lista de jugadores a los cuales puede pasarles la pelota y la indicación de si puede tirar la pelota al arco contrario. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Armar un grafo dirigido con un nodo por cada jugador de nuestro equipo y un eje dirigido de un jugador a otro si el primero puede pasarle la pelota al segundo. Agregar un nodo t que representa al arco contrario, con un eje dirigido hacia él por cada jugador que puede tirar al arco. Lo que pide el enunciado es la cantidad de caminos disjuntos en aristas que van del nodo 1 (arquero) al nodo t (arco contrario). Como es sabido, esa cantidad puede calcularse asignado capacidad 1 a todos los arcos y calculando el flujo máximo de 1 a t.

NOTA: Propuesto originalmente por Michel Mizrahi.

TOMADO: 2014C1P2 07-JUL-2014.

390. Astro Boy ha crecido, y ha cambiado su nombre a Astro Void. Desde el cuatrimestre pasado se dedica a la compraventa de asteroides para ganarse la vida. Sea $p \in \mathbb{N}^n$ tal que p_i es el precio de un asteroide el i-ésimo día en una secuencia de n días ($i = 1, 2, \dots n$). Astro Void quiere comprar y vender asteroides durante esos n días de manera tal que todas las ventas que haga "valgan la pena". Para que la venta el día j de un asteroide comprado el día i ($j > i$) valga la pena, debe cumplirse que $p_j - p_i > j - i$, es decir, se debe ganar más dinero que los días transcurridos entre la compra y la venta. Debido a las dificultades que existen en el almacenaje y transporte de asteroides, Astro Void comienza y termina sin asteroides, puede comprar a lo sumo un asteroide cada día, y puede vender a lo sumo un asteroide cada día. Diseñar un algoritmo que determine la máxima cantidad de ventas que puede realizar Astro Void respetando las restricciones indicadas. Por ejemplo, para los vectores $p = (1, 1, 1, 5, 5, 5)$, $p = (1, 1, 3, 3)$ y $p = (2, 7, 1, 8, 2, 8, 1, 8)$, los resultados deben ser respectivamente 3, 1 y 4. El algoritmo debe tener complejidad $O(n^3)$ y estar basado en grafos. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

SOLUCIÓN: Armar un grafo con un nodo por cada posible día de compra, y un nodo por cada posible día de venta; la cantidad de nodos es $O(n)$. Agregar un eje entre el i-ésimo nodo compra y el j-ésimo nodo venta ($j > i$) si y sólo si la venta vale la pena; la cantidad de ejes es $O(n^2)$. La máxima cantidad de ventas que valen la pena es la cardinalidad de una correspondencia máxima en el grafo descripto. Dicha cardinalidad puede calcularse mediante flujo máximo de la manera usual. El valor del flujo máximo está acotado por n ya que hay un corte con esa capacidad. Por lo tanto, ya sea que se use FF o FFEK la cantidad de iteraciones es $O(n)$, de modo que el costo de encontrar el flujo máximo es $O(mn) = O(n^3)$ si la red está representada con listas de sucesores. El costo de armar la red no supera a eso.

NOTA: Propuesto originalmente por Guido Tagliavini Ponce.

TOMADO: 2016C2P2 30-NOV-2016.

Reducciones polinomiales

391. Sean Π_1 y Π_2 los siguientes problemas.

Π_1: NOMBRE1
Entrada: xxx.
Pregunta: ¿xxx?

Π_2: NOMBRE2
Entrada: xxx.
Pregunta: ¿xxx?

Variante 1: Encontrar una reducción polinomial de Π_1 a Π_2 y otra de Π_2 a Π_1. Demostrar que efectivamente son reducciones polinomiales.

Variante 2: (a) Demostrar que si $\Pi_1 \in$ P entonces $\Pi_2 \in$ P.
(b) Demostrar que si $\Pi_1 \in$ NP-completo entonces $\Pi_2 \in$ NP-completo.

Variante 3: Demostrar que $\Pi_2 \in$ NP-completo.
SUGERENCIA: Usar que $\Pi_1 \in$ NP-completo.

Variante 4: Demostrar que $\Pi_2 \in$ NP-completo mediante una reducción polinomial y usando que $\Pi_1 \in$ NP-completo.

Variante 5: (a) Encontrar una reducción polinomial de Π_1 a Π_2. Demostrar que efectivamente es una reducción polinomial.
(b) ¿Qué se puede decir de Π_2 sabiendo que $\Pi_1 \in$ NP-completo? Justificar.

Variante 6: Bajo la hipótesis de que uno de los siguientes problemas está en P, demostrar que el otro también está en P mediante una reducción polinomial. Indicar claramente cuál problema se supone que está en P y cuál se intenta demostrar que está en P.

Variante 7: Bajo la hipótesis de que uno de los siguientes problemas es NP-completo, demostrar que el otro también lo es mediante una reducción polinomial. Indicar claramente cuál problema se supone que es NP-completo y cuál se intenta demostrar que lo es.

SOLUCIÓN:

Variante 1: Hay que encontrar las dos transformaciones, demostrar que son polinomiales, que transforman instancias adecuadamente, y que la respuesta a un problema es SÍ si y sólo si la respuesta al otro problema es SÍ.

Variante 2: Para el primer punto alcanza con encontrar una reducción polinomial de Π_2 a Π_1. Alternativamente, si se sabe que $\Pi_1 \in$ NP-completo, al suponer que está en P resulta P = NP, por lo que demostrar $\Pi_2 \in$ P equivale a demostrar $\Pi_2 \in$ NP.

Para el segundo punto alcanza con encontrar una reducción polinomial de Π_1 a Π_2 y demostrar que $\Pi_2 \in$ NP.

Variante 3: Ídem segundo punto de la Variante 2.

Variante 4: Ídem Variante 3, pero fuerza la reducción de Π_1 a Π_2.

Variante 5: Para el primer punto hay que encontrar las transformación, demostrar que es polinomial, que transforma instancias adecuadamente, y que la respuesta a un problema es SÍ si y sólo si la respuesta al otro problema es SI.

Respecto del segundo punto, $\Pi_2 \in$ NP-hard. Podría ser NP-completo, pero para deducirlo habría que demostrar que está en NP.

Variante 6: Ídem primer punto de la Variante 2, pero hay que elegir el sentido de la reducción.

Variante 7: Ídem segundo punto de la Variante 2, pero hay que elegir el sentido de la reducción.

NOTA: Este es en realidad un meta-ejercicio que puede instanciarse con las reducciones polinomiales de los ejercicios siguientes. Para la Variante 1 y la Variante 2 se necesitan reducciones polinomiales en ambos sentidos, mientras que para las otras variantes se necesita sólo una reducción.

392. Reducciones en ambos sentidos.

Π_1: PARTICIÓN
Entrada: multi-conjunto A de enteros positivos.
Pregunta: ¿existe $A' \subseteq A$ tal que suma(A') = suma$(A - A')$?

Π_2: PARTICIÓN DE ENTEROS
Entrada: multi-conjunto B de enteros.
Pregunta: ¿existe $B' \subseteq B$ tal que suma(B') = suma$(B - B')$?

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_2$: La transformación es la función identidad, es decir, $B = f(A) = A$.

$\Pi_2 \leq_p \Pi_1$: Calcular el valor absoluto de los enteros, y eliminar los de valor 0. Los que son negativos en Π_2, deben cambiar de conjunto en Π_1, y viceversa.

TOMADO: 2004C1P2 14-JUL-2004 (ambos sentidos).

393. Reducciones en ambos sentidos.

Π_1: PARTICIÓN
Entrada: multi-conjunto A de enteros positivos.
Pregunta: ¿existe $A' \subseteq A$ tal que suma(A') = suma$(A - A')$?

Π_2: SUMA DEL SUBCONJUNTO
Entrada: multi-conjunto C de enteros positivos; $k \in \mathbb{N}$.
Pregunta: ¿existe $C' \subseteq C$ tal que suma$(C') = k$?

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_2$: Sea $S = \text{suma}(A)$. Definir $(C, k) = f(A) = \text{IF}(S \bmod 2 = 0; (A, S/2); \text{instancia negativa}) = \text{IF}(S/2 \in \mathbb{Z}; (A, S/2); \text{instancia negativa})$. Para poder repartirlo en partes iguales, la suma debe ser par, y en tal caso cada parte debe sumar la mitad.

$\Pi_2 \leq_p \Pi_1$: Sea $S = \text{suma}(C)$. Si $S < 2k$ agregar $2k - S$, si $S = 2k$ no agregar nada, y si $S > 2k$ agregar $S - 2k$. Por ejemplo, definir $A = f(C, k) = C \cup \text{IF}(S = 2k; \emptyset; \{|S - 2k|\})$. La motivación es la siguiente. Supongamos que una parte suma k, y por lo tanto la otra $S - k$. Si pensamos en agregar un elemento x en la parte que suma k para que sume lo mismo que la otra parte, esa parte va a pasar a sumar $k + x$ que debe coincidir con $S - k$. Despejando x de $k + x = S - k$ resulta $x = S - 2k$. Sin embargo, si k es bastante grande resulta $x = S - 2k < 0$, lo cual no está permitido. En tal caso es equivalente agregar $-x = 2k - S$ en la parte que suma $S - k$. Podemos definir la reducción comparando S y $2k$, o compararlos implícitamente tomando el valor absoluto de la resta, como figura arriba, aunque de todos modos hay que evitar agregar un 0 inútil cuando $S = 2k$, ya que tampoco está permitido.

Alternativamente, definir $A = f(C, k) = C \cup \{S, S + |S - 2k|\}$. La motivación es la misma que en la primera transformación, pero en vez de agregar $|S - 2k|$ lo agrandamos sumando S, por lo que agregamos otro elemento de valor S para compenzar. En general, podríamos agregar cualquier par de elementos j y $j+|S-2k|$, siempre que j sea lo suficientemente grande como para asegurar (al demostrar la vuelta) que no pueden estar ambos elementos del mismo lado de la partición. Este tipo de transformaciones tienen la ventaja de que no es necesario considerar por separado el caso $S = 2k$.

Alternativamente, definir $A = f(C, k) = C \cup \{S, 2k\}$. La motivación es la siguiente. Supongamos que una parte suma k, y por lo tanto la otra $S - k$. Si pensamos en agregar un elemento en cada parte para que sumen lo mismo, podrámos agregar $S - k$ en la parte que suma k, y k en la parte que suma $S-k$. Sin embargo, como no sabemos si C' existe realmente, $S-k$ podría ser negativo, lo cual podemos solucionar sumando k a los dos elementos que mencionamos, obteniendo $S-k+k = S$ y $k+k = 2k$. Notar que cuando $S \leq 2k$ esta transformación coincide con la anterior.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

394. Reducción de Π_1 a Π_2.

Π_1: SUMA DEL SUBCONJUNTO
Entrada: multi-conjunto C de enteros positivos; $k \in \mathbb{N}$.
Pregunta: ¿existe $C' \subseteq C$ tal que $\text{suma}(C') = k$?

Π_2: SUMA CERO
Entrada: multi-conjunto D de enteros.
Pregunta: ¿existe $D' \subseteq D$ tal que $D' \neq \emptyset$ y $\text{suma}(D') = 0$?

SOLUCIÓN: Agregar $-k$, es decir, definir $D = f(C, k) = C \cup \{-k\}$.

NOTA: Propuesto originalmente por Pablo Heiber.

TOMADO: 2010C2R2 22-DIC-2010.

395. Reducción de Π_1 a Π_2.

Π_1: PARTICIÓN
Entrada: multi-conjunto A de enteros positivos.
Pregunta: ¿existe $A' \subseteq A$ tal que $\text{suma}(A') = \text{suma}(A - A')$?

Π_2: KRR
Entrada: familia $\mathbb{B}$ de rectángulos; rectángulo C. (Todos estos rectángulos están en el plano y tienen lados paralelos a los ejes del sistema de coordenadas.)
Pregunta: ¿es posible desplazar en el plano los rectángulos de $\mathbb{B}$ (sin rotarlos) de manera tal que queden ubicados dentro de C sin que se superpongan entre sí (salvo sus bordes)?

SOLUCIÓN: Definir para cada $a \in A$ un rectángulo en $\mathbb{B}$ de lados $1 \times a$; definir C de lados $2 \times \text{suma}(A)/2$; todos estos elementos están ubicados en cualquier parte del plano. La primera fila de la organización resultante luego de desplazar los rectángulos determina qué debe contener A'.

NOTA: Propuesto originalmente por Esteban Feuerstein.

TOMADO: 2004C1R2 09-AGO-2004.

396. Reducciones de Π_0 a los otros. OJO que no es claro si $\Pi_3 \in$ NP.

Π_0: CUBRIMIENTO DE EJES POR VÉRTICES (VERTEX COVER)
Entrada: grafo $G = (V, E)$; $k \in \mathbb{N}_0$.
Pregunta: ¿existe $V' \subseteq V$ tal que $|V'| \leq k$ y todo elemento de E es incidente a algún elemento de V'?

Π_1: CUBRIMIENTO DE CONJUNTOS (SET COVER)
Entrada: conjuntos finitos $S_1, S_2, \ldots S_p$; $k \in \mathbb{N}_0$.
Pregunta: ¿existe $I' \subseteq I = \{1, 2, \ldots p\}$ tal que $|I'| \leq k$ y $\cup_{i \in I'} S_i = \cup_{i \in I} S_i$?

Π_2: HITTING SET
Entrada: conjuntos finitos $S_1, S_2, \ldots S_p$; $k \in \mathbb{N}_0$.
Pregunta: ¿existe un conjunto S tal que $|S| \leq k$ y para todo $i = 1, 2, \ldots p$ se cumple $S_i \cap S \neq \emptyset$?

Π_3: CUBRIMIENTO DE CICLOS POR VÉRTICES
Entrada: grafo $H = (W, F)$; $\ell \in \mathbb{N}_0$.
Pregunta: ¿existe $W' \subseteq W$ tal que $|W'| \leq \ell$ y todo ciclo simple de H tiene algún vértice en W'?

SOLUCIÓN:

$\Pi_0 \leq_p \Pi_1$: Para cada nodo v definir S_v como el conjunto de ejes incidentes a v. Notar que la unión de todos es E.

Si Π_0 responde SI, sea V' que cubre a todos los ejes; sea $I' = V'$; sea $e \in E$ cualquiera; sea $v \in V'$ que lo cubre. Se deduce que $v \in I'$, y como $e \in S_v$ entonces $e \in \cup_{i \in I'} S_i$.

Si Π_1 responde SI, sea I' que cubre a $E = \cup_{i \in I} S_i$; sea $V' = I'$; sea $e \in E$ cualquiera; sea S_v con $v \in I'$ que lo contiene. Se deduce que v cubre a e, y entre todos los de V' cubren todo E.

$\Pi_0 \leq_p \Pi_2$: Para cada eje $e = (v, w)$ definir $S_e = \{v, w\}$ (los extremos de e).

$\Pi_0 \leq_p \Pi_3$: Revisar esto.

Agregar un vértice por cada eje, adyacente a los dos extremos del eje. Definir además $\ell = k$.

Si Π_0 responde SI, el mismo conjunto $W' = V'$ en Π_3 cubre todos los ciclos, ya que los ciclos que ya estaban en G tienen cubiertos cada uno de sus ejes, y los ciclos nuevos de H (que son triángulos) tienen cubiertos los ejes que ya estaban en G.

Si Π_3 responde SI, existe un conjunto de la misma o menor cardinalidad que sólo usa los vértices que ya estaban en G (si se usara un vértice nuevo, lo sacamos y ponemos uno de los originales, o ninguno si alguno de los originales también está). Ese conjunto cubre los ejes en Π_0, ya que cada nuevo triángulo de Π_3 tiene uno de sus nodos originales en el conjunto.

NOTA: Propuesto originalmente por Pablo Heiber ($\Pi_0 \leq_p \Pi_2$). Encontrado originalmente en Internet ($\Pi_0 \leq_p \Pi_3$).

TOMADO: 2007C1R2 08-AGO-2007 ($\Pi_0 \leq_p \Pi_3$), 2010C2P2 06-DIC-2010 ($\Pi_0 \leq_p \Pi_2$), 2014C1R2 23-JUL-2014 ($\Pi_0 \leq_p \Pi_1$).

397. Reducciones de Π_0 a los otros.

Π_0: CONJUNTO DOMINANTE DE VÉRTICES
Entrada: grafo $G = (V, E)$; $k \in \mathbb{N}$.
Pregunta: ¿existe $V' \subseteq V$ tal que $|V'| \leq k$ y todo elemento de $V - V'$ es adyacente a algún elemento de V'?

Π_1: CUBRIMIENTO DE CONJUNTOS (SET COVER)
Entrada: conjuntos finitos $S_1, S_2, \ldots S_p$; $k \in \mathbb{N}_0$.
Pregunta: ¿existe $I' \subseteq I = \{1, 2, \ldots p\}$ tal que $|I'| \leq k$ y $\cup_{i \in I'} S_i = \cup_{i \in I} S_i$?

Π_2: HITTING SET
Entrada: conjuntos finitos $S_1, S_2, \ldots S_p$; $k \in \mathbb{N}_0$.
Pregunta: ¿existe un conjunto S tal que $|S| \leq k$ y para todo $i = 1, 2, \ldots p$ se cumple $S_i \cap S \neq \emptyset$?

SOLUCIÓN: La reducción en ambos casos es la misma. Para cada nodo v definir $S_v = \{v\} \cup N(v)$ (el nodo y sus vecinos).

$\Pi_0 \leq_p \Pi_1$: Si Π_0 responde SI, sea V' un conjunto dominante de a lo sumo k elementos. La unión de los conjuntos definidos a partir de los nodos de V' forma todo V.

Si Π_1 responde SI, sea I' un cubrimiento de a lo sumo k elementos. Para cada nodo de V existe un $i \in I'$ tal que S_i lo contiene, es decir, v es dominado por i, de modo que I' es un conjunto dominante.

$\Pi_0 \leq_p \Pi_2$: Si Π_0 responde SI, sea V' un conjunto dominante de a lo sumo k elementos. Para cada S_i algún elemento lo toca, ya que contiene los nodos que dominan al nodo i, y todos los nodos son dominados.

Si Π_2 responde SI, sea S un conjunto de a lo sumo k elementos con intersección no vacía con cada S_i. Eso implica que cada nodo es dominado por algún elemento de S.

TOMADO: 2012C1P2 11-JUL-2012 ($\Pi_0 \leq_p \Pi_1$), 2012C1R2 08-AGO-2012 ($\Pi_0 \leq_p \Pi_2$).

398. Reducciones en ambos sentidos.

Π_1: EMPAQUETAMIENTO DE CONJUNTOS (SET PACKING)
Entrada: conjuntos finitos $S_1, S_2, \ldots S_p$; $k \in \mathbb{N}$.
Pregunta: ¿k o más conjuntos son disjuntos de a pares?

Π_2: CONJUNTO INDEPENDIENTE MÁXIMO
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿tiene G un conjunto independiente de k o más vértices?

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_2$: Poner un nodo por cada conjunto y hacer dos nodos adyacentes si y sólo si los conjuntos correspondientes tienen intersección no vacía.

$\Pi_2 \leq_p \Pi_1$: Para cada nodo v definir S_v como el conjunto de ejes incidentes a v.

TOMADO: 2001C2R2 26-DIC-2001 ($\Pi_2 \leq_p \Pi_1$), 2005C1R2 08-AGO-2005 (ambos sentidos).

399. Reducciones en ambos sentidos.

Π_1: TSP
Entrada: grafo completo G en el cual cada eje tiene asociada una longitud entera no negativa; $T \in \mathbb{N}_0$.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices, y de longitud total a lo sumo T?

Π_2: k-TSP
Entrada: grafo completo H en el cual cada eje tiene asociada una longitud entera no negativa; $T \in \mathbb{N}_0$; vértice v de H.
Pregunta: ¿es posible cubrir los vértices de H con k o menos circuitos simples que coincidan sólo en v, y tales que la suma de sus longitudes totales sea a lo sumo T?
(Notar que k no forma parte de la entrada sino que es cualquier entero positivo fijo.)

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_2$: Elegir un nodo cualquiera para que sea v, y armar H agregando $k-1$ nodos a G, conectados a v por ejes de longitud 0, y a los otros nodos por ejes de longitud infinito. Si Π_1 responde SI, ese mismo circuito, junto con un circuito de ida y vuelta por cada nodo agregado, cumplen lo pedido en Π_2. Si Π_2 responde SI, los nodos agregados sólo pueden formar circuito con v (caso contrario, se superaría T), de modo que los otros nodos forman un circuito como el pedido en Π_1.

$\Pi_2 \leq_p \Pi_1$: Armar G remplazando v por k copias de sí mismo, conectadas entre sí por ejes de longitud 0, y a los otros nodos originales de H por ejes de igual longitud que los originales. Si Π_1 responde SI, cada subcircuito de G entre dos copias de v se corresponde con un circuito (eventualmente de menos de 3 nodos) en H, con la misma longitud. Si Π_2 responde SI, recorriendo los circuitos de H, y pasando de una copia de v a otra cada vez que termina un circuito y comienza otro, se consigue en G un circuito como el pedido en Π_1.

NOTA: La versión de optimización de 2-TSP fue el problema de 2007C2L3. A unos alumnos se les ocurrió la reducción a TSP, lo que les permitió traducir soluciones factibles de TSP a solucioes factibles de 2-TSP, manteniendo el costo.

TOMADO: 2008C1P2 11-JUL-2008 ($\Pi_1 \leq_p \Pi_2$), 2008C1R2 25-JUL-2008 ($\Pi_2 \leq_p \Pi_1$).

400. Reducción de Π_1 a Π_2.

Π_1: VECINOS DISTINGUIBLES
Entrada: grafo H; $\ell \in \mathbb{N}$.
Pregunta: ¿es posible asignar un número entero entre 1 y ℓ a cada vértice de H, de manera tal que para todo vértice sus vecinos tengan números diferentes entre sí?

Π_2: COLOREO DE VÉRTICES
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible colorear los vértices de G con k o menos colores, de manera tal que vértices adyacentes reciban colores diferentes?

SOLUCIÓN: Considerar los mismos vértices que están en el grafo original (sin los ejes), y para cada uno de ellos agregar un eje entre cada par de sus vecinos. Definir además $k = \ell$.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez, quien en su momento argumentó que $\Pi_1 \in$ NP-completo.

TOMADO: 2017C1R2 21-JUL-2017.

401. Reducción de Π_1 a Π_2.

Π_1: COLOREO DE VÉRTICES
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible colorear los vértices de G con k o menos colores, de manera tal que vértices adyacentes reciban colores diferentes?
(Suponer que k es menor o igual que la cantidad de vértices de G.)

Π_2: CONJUNTO INDEPENDIENTE MÁXIMO
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿tiene G un conjunto independiente de k o más vértices?

SUGERENCIA: Definir $f(G, k) = (G_k, n)$, donde n es la cantidad de vértices de G y G_k es el grafo formado por k copias de G, con ejes adicionales de manera tal que las k copias de cada vértice formen K_k.

SOLUCIÓN: Si Π_1 responde SI, consideremos un coloreo de G que usa $j \leq k$ colores, y sea V_i el conjunto de nodos pintados con el i-ésimo color para $i = 1, 2, \ldots j$. Notar que cada V_i es un conjunto independiente, y que forman una partición de V. Si ahora elegimos en la i-ésima copia de G a los nodos de V_i, forman un conjunto independiente de n elementos. Son n porque la unión de los V_i es V. Son independientes porque cada V_i es independiente en G (y por lo tanto en una de sus copias), y los nodos de dos V_i distintos no son adyacentes en Π_2 porque corresponden a nodos diferentes de G.

Si Π_2 responde SI, sea V_i el conunto de nodos independientes que están en la i-ésima copia de G. Como los nodos independientes no pueden contener más de una copia de cada nodo de G porque serían adyacentes, pero a la vez hay n nodos independients, alguna copia de cada nodo de G pertenece al conjunto independiente, y por lo tanto a exactamente un V_i. Para cada V_i, pintamos a los nodos originales de G con el i-ésimo color. Por lo dicho cada nodo de G recibe un color entre 1 y la cantidad de copias (k), y nodos del mismo color forman un conjunto independiente en su copia, y por lo tanto en G.

NOTA: Propuesto originalmente por Diego Delle Donne.

TOMADO: 2015C1R2 17-JUL-2015.

402. Reducciones de cada uno a Π_0.

Π_1: COLOREO DE VÉRTICES
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible colorear los vértices de G con k o menos colores, de manera tal que vértices adyacentes reciban colores diferentes?
(Suponer que k es menor o igual que la cantidad de vértices de G.)

Π_2: CUBRIMIENTO DE VÉRTICES POR SUBGRAFOS COMPLETOS DISJUNTOS
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible cubrir los vértices de G con k o menos subgrafos completos disjuntos de a pares?
(Suponer que k es menor o igual que la cantidad de vértices de G.)

Π_0: M-PARTICIÓN
Entrada: grafo H; matriz simétrica $M \in \{0, 1, *\}^{\ell \times \ell}$.
Pregunta: ¿es posible repartir los vértices de H en ℓ conjuntos disjuntos $V_1, V_2, \dots V_\ell$, eventualmente vacíos, de tal modo que entre V_i y V_j no exista ningún eje si $M[i, j] = 0$, existan todos los ejes posibles si $M[i, j] = 1$, y ocurra cualquier cosa si $M[i, j] = *$?

SOLUCIÓN: En ambos casos definir $H = G$, $\ell = k$, $M[i, j] = *$ para $i \neq j$.

$\Pi_1 \leq_p \Pi_0$: Definir además $M[i, i] = 0$.

$\Pi_2 \leq_p \Pi_0$: Definir además $M[i, i] = 1$.

TOMADO: 2005C1P2 15-JUL-2005 ($\Pi_1 \leq_p \Pi_0$), 2007C1P2 10-JUL-2007 ($\Pi_2 \leq_p \Pi_0$).

403. Reducción de Π_1 a Π_2.

Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, un homomorfismo de G_1 a G_2 es una función $f : V_1 \to V_2$ tal que $(v, w) \in E_1 \Rightarrow (f(v), f(w)) \in E_2$. Decimos que G_1 es homomorfo a G_2 si y sólo si existe un homomorfismo de G_1 a G_2.

Π_1: COLOREO DE VÉRTICES
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible colorear los vértices de G con k o menos colores, de manera tal que vértices adyacentes reciban colores diferentes?
(Suponer que k es menor o igual que la cantidad de vértices de G.)

Π_2: HOMOMORFISMO DE GRAFOS
Entrada: dos grafos G y H.
Pregunta: ¿es G homomorfo a H?

SOLUCIÓN: Definir $f(G, k) = (G, K_k)$. La demostración de que eso anda surge del Ejercicio 308.

NOTA: Propuesto originalmente por Federico Lebrón.

TOMADO: 2011C2R2 21-DIC-2011.

404. Reducción de Π_p a Π_q para $2 < p < q$. Se cumple $\Pi_k \in$ NP-completo para $k \geq 3$.

Π_k: k-COLOREO DE VÉRTICES
Entrada: grafo G.
Pregunta: ¿es posible colorear los vértices de G con k o menos colores, de manera tal que vértices adyacentes reciban colores diferentes?
(Notar que k no forma parte de la entrada sino que es cualquier entero positivo fijo.)

SOLUCIÓN: Definir $f(G) = G + K_{q-p}$ (grafo junta). Por el Ejercicio 10.8 de la Práctica se cumple que $\chi(G + K_{q-p}) = \chi(G) + \chi(K_{q-p}) = \chi(G) + q - p$, es decir $\chi(G + K_{q-p}) = \chi(G) + q - p$ o equivalentemente $\chi(G) = \chi(G + K_{q-p}) - q + p$. Si Π_p responde SI, significa que $\chi(G) \leq p$ y por lo tanto $\chi(G + K_{q-p}) = \chi(G) + q - p \leq p + q - p = q$. Si Π_q responde SI, significa que $\chi(G + K_{q-p}) \leq q$ y por lo tanto $\chi(G) = \chi(G + K_{q-p}) - q + p \leq q - q + p = p$.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez ($p = 3$ y $q = 5$).

TOMADO: 2016C1P2 02-JUL-2016 ($p = 314$ y $q = 315$).

405. Reducciones en ambos sentidos.

Π_1: CAMINO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un camino que pasa exactamente una vez por cada uno de sus vértices?

Π_2: ASIGNACIÓN SEPARADA
Entrada: grafo H de n vértices.
Pregunta: ¿es posible asignar a cada vértice de H un entero distinto entre 1 y n de manera tal que vértices adyacentes reciban enteros no consecutivos?

SUGERENCIA: Asignar equivale a establecer un orden.

SOLUCIÓN: La reducción en cualquiera de los dos sentidos es complementar el grafo.

$\Pi_1 \leq_p \Pi_2$: Si Π_1 responde SI, podemos numerar los nodos del camino siguiendo el orden en que aparecen en el mismo. Eso asigna un entero distinto entre 1 y n a cada nodo del complemento, ya que el camino original es hamiltoniano. Además, si dos vértices son adyacentes en el complemento no lo eran en el grafo original, y por lo tanto no reciben enteros consecutivos.

Si Π_2 responde SI, podemos ordenar los nodos de acuerdo al número que reciben. Cada par de nodos consecutivos no es adyacente en el complemento, de modo que sí lo es en el grafo original, y por lo tanto forman un camino. El camino es hamiltoniano porque todos los nodos reciben un número.

$\Pi_2 \leq_p \Pi_1$: Análogo.

NOTA: Propuesto originalmente por Mariano Pérez Rodríguez.

TOMADO: 2014C1P2 07-JUL-2014 ($\Pi_1 \leq_p \Pi_2$).

406. Reducciones en ambos sentidos.

Π_1: CAMINO MÁXIMO
Entrada: grafo G; $k \in \mathbb{N}_0$.
Pregunta: ¿tiene G un camino simple de k o más ejes?

Π_2: CAMINO MÁXIMO ENTRE DOS VÉRTICES
Entrada: grafo H; dos vértices $v \neq w$ de H; $\ell \in \mathbb{N}$.
Pregunta: ¿tiene H un camino simple entre v y w de ℓ o más ejes?

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_2$: Para obtener H agregar a G dos vértices v y w adyacentes a todos los vértices de G. Definir además $\ell = k + 2$.

Al camino en G le podemos agregar v y w para obtener el camino en H, y hacer la operación inversa para volver a G.

$\Pi_2 \leq_p \Pi_1$: Para obtener G conectar a v un camino simple de $p = n(H)$ ejes y hacer lo mismo con w. Definir además $k = \ell + 2p$.

Es claro que si H tiene un camino entre v y w de ℓ o más ejes, entonces G tiene un camino de k o más ejes.

Si G tiene un camino de k o más ejes, ese camino debe tener parte en los dos caminos agregados a H, y por lo tanto pasa por v y por w. Como el largo máximo dentro de los agregados es $2p$, la parte que queda dentro de H conecta v con w y tiene ℓ o más ejes.

NOTA: Los caminos agregados a H podrían tener menos ejes. Creo que el mínimo es $p - \ell$.

TOMADO: 2002C1P2 10-JUL-2002 (ambos sentidos), 2009C1R2 26-AGO-2009 ($\Pi_1 \leq_p \Pi_2$).

407. Reducción de Π_1 a Π_2.

Π_1: CAMINO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un camino que pasa exactamente una vez por cada uno de sus vértices?

Π_2: CAMINO MÁXIMO
Entrada: grafo G; $k \in \mathbb{N}_0$.
Pregunta: ¿tiene G un camino simple de k o más ejes?

SOLUCIÓN: Definir $f(G) = (G, n(G) - 1)$.

TOMADO: 2015C1P2 04-JUL-2015.

408. Reducciones de Π_1 a Π_2 y de Π_3 a Π_4.

Π_1: CAMINO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un camino que pasa exactamente una vez por cada uno de sus vértices?

Π_2: CAMINO SEMIHAMILTONIANO
Entrada: grafo H.
Pregunta: ¿tiene H un camino simple que pasa por al menos la mitad de sus vértices?

Π_3: CIRCUITO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices?

Π_4: CIRCUITO SEMIHAMILTONIANO
Entrada: grafo H.
Pregunta: ¿tiene H un circuito simple que pasa por al menos la mitad de sus vértices?

SOLUCIÓN: Agregar tantos vértices aislados como vértices tiene G, o más en general, cualquier grafo de esa cantidad de vértices que no tenga camino o circuito hamiltoniano. Otra posibilidad es agregar una copia de G.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2016C2R2 19-DIC-2016 ($\Pi_3 \leq_p \Pi_4$).

409. Reducciones de Π_0 a los otros.

Π_0: CAMINO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un camino que pasa exactamente una vez por cada uno de sus vértices?

Π_1: ÁRBOL GENERADOR DE GRADO ACOTADO
Entrada: grafo G; $k \in \mathbb{N}_0$.
Pregunta: ¿tiene G un árbol generador en el cual todos los vértices tienen grado a lo sumo k?

Π_2: ÁRBOL GENERADOR CON HOJAS MÍNIMAS
Entrada: grafo G; $k \in \mathbb{N}_0$.
Pregunta: ¿tiene G un árbol generador con k o menos hojas?

Π_3: ÁRBOL GENERADOR ISOMORFO
Entrada: grafo G; árbol T.
Pregunta: ¿tiene G un árbol generador isomorfo a T?

SOLUCIÓN:

$\Pi_0 \leq_p \Pi_1$: Definir $f(G) = (G, 2)$. Si Π_1 responde SI, el camino hamiltoniano en G es un árbol generador de grado a lo sumo 2. Si Π_2 responde SI, el árbol de grado a lo sumo 2 es un camino (hay que demostrarlo), que al ser generador contiene a todos los vértices del grafo, y por lo tanto es hamiltoniano.

$\Pi_0 \leq_p \Pi_2$: Definir $f(G) = (G, 2)$.

$\Pi_0 \leq_p \Pi_3$: Definir $f(G) = (G, P_{n(G)})$.

TOMADO: 2015C2P2 30-NOV-2015 ($\Pi_0 \leq_p \Pi_2$), 2017C2P2 01-DIC-2017 ($\Pi_0 \leq_p \Pi_3$), 2018C1R2 20-JUL-2018 ($\Pi_0 \leq_p \Pi_1$).

410. Reducciones en ambos sentidos.

Π_1: CAMINO HAMILTONIANO MÍNIMO
Entrada: grafo G en el cual cada eje tiene asociada una longitud entera no negativa; $T \in \mathbb{N}_0$.
Pregunta: ¿tiene G un camino que pasa exactamente una vez por cada uno de sus vértices, y de longitud total a lo sumo T?

Π_2: CIRCUITO HAMILTONIANO MÍNIMO
Entrada: grafo G en el cual cada eje tiene asociada una longitud entera no negativa; $T \in \mathbb{N}_0$.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices, y de longitud total a lo sumo T?

SOLUCIÓN: Revisar esto.

$\Pi_1 \leq_p \Pi_2$: Agregar a G un vértice al cual están conectados todos los vértices originales del grafo, por medio de ejes de longitud 0.

$\Pi_2 \leq_p \Pi_1$: Elegir un vértice cualquiera del grafo y reemplazarlo por dos vértices conectados entre sí por un eje de longitud infinito. Ambos vértices son adyacentes a todos los vértices a los cuales era adyacente el vértice eliminado.

TOMADO: 2001C2P2 11-DIC-2001 (ambos sentidos).

411. Reducción de Π_1 a Π_2.

Π_1: CAMINO HAMILTONIANO ENTRE DOS VÉRTICES
Entrada: grafo H; dos vértices $v \neq w$ de H.
Pregunta: ¿tiene H un camino entre v y w que pasa exactamente una vez por cada uno de sus vértices?

Π_2: CIRCUITO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices?

SOLUCIÓN: Agregar un vértice adyacente sólo a v y w.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez.

TOMADO: 2016C2P2 30-NOV-2016.

412. Reducciones en ambos sentidos.

Π_1: CIRCUITO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices?

Π_2: CIRCUITO HAMILTONIANO DIRIGIDO
Entrada: digrafo H.
Pregunta: ¿tiene H un circuito dirigido que pasa exactamente una vez por cada uno de sus vértices?

SUGERENCIA: Para $\Pi_2 \leq_p \Pi_1$, por cada vértice v de H generar 3 vértices v_{in}, v y v_{out} en G.

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_2$: Por cada eje de G, poner en H dos arcos de direcciones opuestas. Si Π_1 responde SI, el circuito de G existe en H, tanto si se lo dirige en un sentido como en el otro. Si Π_2 responde SI, al circuito de H se le puede remover la dirección para obtener un circuito en G.

$\Pi_2 \leq_p \Pi_1$: Generar los vértices como dice la sugerencia, haciendo que cada arco $v \to w$ de H sea un eje $(v_{\text{out}}, w_{\text{in}})$ de G. Por cada vértice v en H agregar además ejes (v_{in}, v) y (v, v_{out}) en G.

Si Π_1 responde SI, como cada vértice v sin subíndice tiene grado 2 en G, la única manera de que forme parte de un circuito hamiltoniano es $v_{\text{in}} - v - v_{\text{out}}$. Como eso vale para todos los vértices, el circuito de G es una permutación de fragmentos como ese. Las conexiones entre los fragmentos definen el circuito de H.

Si Π_2 responde SI, cada porción de circuito $v \to w$ en H puede convertirse en $v - v_{\text{out}} - w_{\text{in}} - w$ en G, y el circuito formado por todas esas porciones resulta hamiltoniano.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2009C1P2 10-AGO-2009 ($\Pi_1 \leq_p \Pi_2$), 2012C2R2 19-DIC-2012 ($\Pi_2 \leq_p \Pi_1$).

413. Reducciones de Π_1 a los otros y de Π_2 a los otros.

Π_1: CIRCUITO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices?

Π_2: SUBGRAFO INDUCIDO HAMILTONIANO
Entrada: grafo $G = (V, E)$; conjunto W tal que $\emptyset \neq W \subseteq V$.
Pregunta: ¿tiene G un circuito simple que pasa por todos los vértices de W pero no por otros vértices?

Π_3: SUPERGRAFO INDUCIDO HAMILTONIANO
Entrada: grafo $G = (V, E)$; conjunto W tal que $\emptyset \neq W \subseteq V$.
Pregunta: ¿tiene G un circuito simple que pasa por todos los vértices de W y talvez por otros vértices?

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_2$: Definir $f(G) = (G, V(G))$.

$\Pi_1 \leq_p \Pi_3$: Definir $f(G) = (G, V(G))$.

$\Pi_2 \leq_p \Pi_1$: Definir $f(G, W) = G[W]$ (subgrafo inducido).

$\Pi_2 \leq_p \Pi_3$: Definir $f(G, W) = (G[W], W)$.

NOTA: Propuesto originalmente por Leopoldo Taravilse ($\Pi_1 \leq_p \Pi_3$).

TOMADO: 2016C1R2 15-JUL-2016 ($\Pi_1 \leq_p \Pi_3$), 2017C2R2 18-DIC-2017 ($\Pi_2 \leq_p \Pi_1$), 2018C2R2 19-DIC-2018 ($\Pi_2 \leq_p \Pi_3$).

414. Reducciones de Π_1 a Π_2 y de Π_3 a Π_4.

Π_1: CLIQUE MÁXIMA
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿tiene G un subgrafo completo de k o más vértices?

Π_2: CLIQUE MÁS PESADA
Entrada: grafo G en el cual cada vértice tiene asociado un peso entero positivo; $T \in \mathbb{N}$.
Pregunta: ¿tiene G un subgrafo completo de peso total T o mayor?

Π_3: CONJUNTO INDEPENDIENTE MÁXIMO
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿tiene G un conjunto independiente de k o más vértices?

Π_4: CONJUNTO INDEPENDIENTE MÁS PESADO
Entrada: grafo G en el cual cada vértice tiene asociado un peso entero positivo; $T \in \mathbb{N}$.
Pregunta: ¿tiene G un conjunto independiente de peso total T o mayor?

SOLUCIÓN: La reducción en ambos casos es asignar peso 1 a todos los nodos y definir $T = k$.

TOMADO: 2011C2P2 05-DIC-2011 ($\Pi_3 \leq_p \Pi_4$), 2013C2R2 21-DIC-2013 ($\Pi_1 \leq_p \Pi_2$).

415. Reducción de Π_1 a Π_2.

Π_1: CLIQUE MÁXIMA
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿tiene G un subgrafo completo de k o más vértices?

Π_2: CLIQUE DE MÁXIMA FRONTERA
Entrada: grafo H; $\ell \in \mathbb{N}_0$.
Pregunta: ¿tiene H un subgrafo completo S tal que ℓ o más ejes de H tienen exactamente un extremo en S?

SUGERENCIA: Definir $(H, \ell) = f(G, k) = (G + K_p^c, p \times k)$, donde $p = |V| + |E| + 1$ y $H = G + K_p^c$ es el grafo junta entre G y p vértices aislados.

SOLUCIÓN: Si Π_1 responde SI, en Π_2 basta elegir como S el subgrafo completo de k o más vértices de G.

Si Π_2 responde SI, sea S un subgrafo de H que cumple lo pedido. Si S tiene k o más vértices de G, listo, ya que esos vértices forman un subgrafo completo de G. Por absurdo supongamos entonces que S tiene frontera al menos $\ell = pk$ pero $k' \leq k - 1$ vértices de G. Notemos que S puede tener a lo sumo un vértice de K_p^c ya que caso contrario no sería un subgrafo completo. La frontera de S se puede acotar como sigue: de los k' vértices de S que están en G salen a lo sumo $|E|$ ejes a $G - S$ y a lo sumo pk' ejes a $K_p^c - S$, mientras que del (posible) único vértice de S que está en

K_p^c salen a lo sumo $|V|$ ejes a $G - S$ y 0 ejes a $K_p^c - S$. La frontera de S es entonces a lo sumo $|E| + pk' + |V| < p + pk' \leq p + p(k-1) = pk = \ell$.

NOTA: Propuesto originalmente por Pablo Haramburu.

TOMADO: 2013C2P2 02-DIC-2013.

416. Reducción de Π_1 a Π_2.

Π_1: CONJUNTO DOMINANTE DE EJES
Entrada: grafo $H = (W, F)$; $\ell \in \mathbb{N}$.
Pregunta: ¿existe $F' \subseteq F$ tal que $|F'| \leq \ell$ y todo elemento de $F - F'$ es adyacente a algún elemento de F'?

Π_2: CONJUNTO DOMINANTE DE VÉRTICES
Entrada: grafo $G = (V, E)$; $k \in \mathbb{N}$.
Pregunta: ¿existe $V' \subseteq V$ tal que $|V'| \leq k$ y todo elemento de $V - V'$ es adyacente a algún elemento de V'?

SOLUCIÓN: Definir $(G, k) = f(H, \ell) = \text{IF}(F = \emptyset; \text{instancia positiva}; (L(H), \ell))$, donde $L(H)$ es el grafo de líneas de H.

TOMADO: 2005C2P2 06-DIC-2005.

417. Reducciones de cada uno a Π_0. OJO que no se sabe si $\Pi_4 \in$ NP-completo.

Π_1: CAMINO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un camino que pasa exactamente una vez por cada uno de sus vértices?

Π_2: CIRCUITO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices?

Π_3: CLIQUE MÁXIMA
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿tiene G un subgrafo completo de k o más vértices?
(Suponer que k es menor o igual que la cantidad de vértices de G.)

Π_4: ISOMORFISMO DE GRAFOS
Entrada: dos grafos G y H.
Pregunta: ¿son isomorfos G y H?

Π_0: SUBGRAFO ISOMORFO
Entrada: dos grafos G y H.
Pregunta: ¿tiene G un subgrafo isomorfo a H? (El subgrafo puede no ser inducido.)

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_0$: Definir $f(G) = (G, P_{n(G)})$.

$\Pi_2 \leq_p \Pi_0$: Definir $f(G) = (G, C_{n(G)})$.

$\Pi_3 \leq_p \Pi_0$: Definir $f(G, k) = (G, K_k)$.

$\Pi_4 \leq_p \Pi_0$: Definir $f(G, H) = \text{IF}(n(G) = n(H) \wedge m(G) = m(H); (G, H); \text{instancia negativa})$. Hay otras posibilidades.

TOMADO: 2002C1R2 07-AGO-2002 ($\Pi_3 \leq_p \Pi_0$), 2012C2P2 05-DIC-2012 ($\Pi_2 \leq_p \Pi_0$), 2018C1P2 06-JUL-2018 ($\Pi_3 \leq_p \Pi_0$).

418. Reducciones de cada uno a Π_0. OJO que no se sabe si $\Pi_4 \in$ NP-completo.

Π_1: CAMINO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un camino que pasa exactamente una vez por cada uno de sus vértices?

Π_2: CIRCUITO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices?

Π_3: CLIQUE MÁXIMA
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿tiene G un subgrafo completo de k o más vértices?
(Suponer que k es menor o igual que la cantidad de vértices de G.)

Π_4: ISOMORFISMO DE GRAFOS
Entrada: dos grafos G y H.
Pregunta: ¿son isomorfos G y H?

Π_0: SUBGRAFO COMÚN MÁXIMO
Entrada: grafo $G_1 = (V_1, E_1)$; grafo $G_2 = (V_2, E_2)$; $\ell \in \mathbb{N}_0$.
Pregunta: ¿tiene G_1 un subgrafo de ℓ o más ejes que sea isomorfo a un subgrafo de G_2? Es decir, ¿existen V_1', V_2', E_1' y E_2' tales que $V_1' \subseteq V_1$, $V_2' \subseteq V_2$, $E_1' \subseteq E_1$, $E_2' \subseteq E_2$, $|E_1'| \geq \ell$, con $G_1' = (V_1', E_1')$ y $G_2' = (V_2', E_2')$ grafos isomorfos? (Los subgrafos pueden no ser inducidos.)

SOLUCIÓN:

$\Pi_1 \leq_p \Pi_0$: Definir $f(G) = (G, P_{n(G)}, n(G) - 1)$.

$\Pi_2 \leq_p \Pi_0$: Definir $f(G) = (G, C_{n(G)}, n(G))$.

$\Pi_3 \leq_p \Pi_0$: Definir $f(G, k) = (G, K_k, k(k-1)/2)$.

$\Pi_4 \leq_p \Pi_0$: Definir $f(G, H) = \mathrm{IF}(n(G) = n(H) \wedge m(G) = m(H); (G, H, m(G)); \text{instancia negativa})$. Hay otras posibilidades.

NOTA: En G&J la pregunta para Π_0 es distinta.

TOMADO: 2003C1R2 04-AGO-2003 ($\Pi_3 \leq_p \Pi_0$), 2014C2R2 19-DIC-2014 ($\Pi_1 \leq_p \Pi_0$), 2015C2R2 18-DIC-2015 ($\Pi_4 \leq_p \Pi_0$).

419. Reducción de Π_1 a Π_2.

Π_1: SUBGRAFO ISOMORFO
Entrada: dos grafos G y H.
Pregunta: ¿tiene G un subgrafo isomorfo a H? (El subgrafo puede no ser inducido.)

Π_2: SUBGRAFO COMÚN MÁXIMO
Entrada: grafo $G_1 = (V_1, E_1)$; grafo $G_2 = (V_2, E_2)$; $\ell \in \mathbb{N}_0$.
Pregunta: ¿tiene G_1 un subgrafo de ℓ o más ejes que sea isomorfo a un subgrafo de G_2? Es decir, ¿existen V_1', V_2', E_1' y E_2' tales que $V_1' \subseteq V_1$, $V_2' \subseteq V_2$, $E_1' \subseteq E_1$, $E_2' \subseteq E_2$, $|E_1'| \geq \ell$, con $G_1' = (V_1', E_1')$ y $G_2' = (V_2', E_2')$ grafos isomorfos? (Los subgrafos pueden no ser inducidos.)

SOLUCIÓN: Definir $(G_1, G_2, \ell) = f(G, H) = (G, H, m(H))$.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

420. Reducciones en ambos sentidos.

Π_1: CORTE MÁXIMO
Entrada: grafo $G = (V, E)$; $k \in \mathbb{N}$.
Pregunta: ¿existe $V' \subseteq V$ tal que k o más ejes tienen un extremo en V' y otro en $V - V'$?

Π_2: SUBGRAFO BIPARTITO MÁXIMO
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿tiene G un subgrafo bipartito de k o más ejes? (El subgrafo puede no ser inducido.)

SOLUCIÓN: Los dos problemas son escencialmente el mismo, y la reducción en cualquiera de los dos sentidos es la identidad.

$\Pi_1 \leq_p \Pi_2$: Tomar como subgrafo bipartito a V' por un lado, $V - V'$ por el otro, y los ejes que tienen un extremo en cada lado.

$\Pi_2 \leq_p \Pi_1$: Tomar como V' a cualquiera de los lados de la partición en el subgrafo bipartito.

TOMADO: 2014C2P2 01-DIC-2014 (sentido a elección).

421. Reducción de Π_1 a Π_2.

Un grafo cluster es un grafo en el que cada componente conexa es un subgrafo completo.

Π_1: SUBGRAFO CLUSTER MÁXIMO
Entrada: grafo G; $k \in \mathbb{N}_0$.
Pregunta: ¿tiene G un subgrafo H tal que H es un grafo cluster de k o más ejes?

Π_2: COLOREO DE MÁXIMO IMPACTO
Entrada: dos grafos G_1 y G_2 sobre el mismo conjunto de vértices; $k \in \mathbb{N}_0$.
Pregunta: ¿existe un coloreo de vértices válido para G_1, tal que k o más ejes de G_2 tienen sus extremos del mismo color?

SUGERENCIA: Definir $(G_1, G_2, k) = f(G, k) = (G^c, G, k)$.

SOLUCIÓN: Si Π_1 responde SI, se puede elegir un color diferente para cada componente conexa de H, y pintar todos sus nodos con ese color en G^c, pintando el resto de los nodos con cualquier color que no produzca conflictos. El coloreo es válido porque cada componente conexa de H forma un conjunto independiente en G^c, y el impacto es al menos k ya que al ser H subgrafo de G cada eje de H aporta al impacto en G.

Si Π_2 responde SI, sea V_i el conjunto de nodos pintados con el color i. Cada V_i es un conjunto independiente en G^c e induce un completo en G. El impacto del coloreo de G^c en G proviene de cada V_i, y es al menos k por hipótesis. Por lo tanto, los ejes de todos los completos en G son al menos k.

NOTA: Propuesto originalmente por Min Chih Lin.

TOMADO: 2013C1P2 10-JUL-2013.

Otros ejercicios de clases de complejidad

422. (a) Reducción de Π_1 a Π_2.

Π_1: PARTICIÓN
Entrada: multi-conjunto A de enteros positivos.
Pregunta: ¿existe $A' \subseteq A$ tal que suma(A') = suma$(A - A')$?

Π_2: BIN PACKING
Entrada: multi-conjunto E de enteros positivos; $b, k \in \mathbb{N}$.
Pregunta: ¿es posible repartir los elementos de E en k conjuntos disjuntos $E_1, E_2, \ldots E_k$, eventualmente vacíos, de tal modo que la suma de los elementos en cada conjunto sea a lo sumo b?

(b) Considerar el siguiente algoritmo para Π_2.

```
función BP(E,b,k)
        ordenar E de manera decreciente
        j ← 1
        E_j ← ∅
        para cada e ∈ E
             si suma(E_j) + e > b
                si (j = k) o (e > b)
                   devolver FALSO
                caso contrario
                   j ← j + 1
                   E_j ← ∅
                fin si
             fin si
             ubicar e en E_j
        fin para
        devolver VERDADERO
fin función
```

i. Describir una implementación eficiente del algoritmo y determinar su complejidad.
ii. ¿Es cierto que el algoritmo nunca usa más de k conjuntos para repartir los elementos de E? Justificar.
iii. ¿Es cierto que la suma de los elementos asignados por el algoritmo a cada conjunto es a lo sumo b? Justificar.

iv. ¿Es cierto que el algoritmo resuelve Π_2? Justificar.

SOLUCIÓN:

(a) Sea $S = \text{suma}(A)$. Definir $(E, b, k) = f(A) = \text{IF}(S \bmod 2 = 0; (A, S/2, 2); \text{instancia negativa}) = \text{IF}(S/2 \in \mathbb{Z}; (A, S/2, 2); \text{instancia negativa})$.

(b) i. El costo está dominado por el ordenamiento. Se puede implementar en $O(|E| \log |E|)$.
  ii. Sí.
  iii. Sí.
  iv. No. El algoritmo puede devolver `FALSO` cuando es posible repartir los elementos como se pide (por ejemplo para $E = \{0.6, 0.5, 0.5, 0.4\}$, $b = 1$ y $k = 2$).

NOTA: Propuesto originalmente por Esteban Feuerstein.

TOMADO: 2003C1P2 11-JUL-2003.

423. (a) Reducciones de Π_0 a los otros y viceversa.

Π_0: COLOREO DE VÉRTICES
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible colorear los vértices de G con k o menos colores, de manera tal que vértices adyacentes reciban colores diferentes?

Π_1: CUBRIMIENTO DE VÉRTICES POR SUBGRAFOS COMPLETOS
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible cubrir los vértices de G con k o menos subgrafos completos?

Π_2: CUBRIMIENTO DE VÉRTICES POR SUBGRAFOS COMPLETOS MAXIMALES
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible cubrir los vértices de G con k o menos subgrafos completos maximales?

Π_3: CUBRIMIENTO DE VÉRTICES POR SUBGRAFOS COMPLETOS DISJUNTOS
Entrada: grafo G; $k \in \mathbb{N}$.
Pregunta: ¿es posible cubrir los vértices de G con k o menos subgrafos completos disjuntos de a pares?

(b) Considerar la siguiente estrategia golosa para Π_3.

```
función Cubrimiento(G,k)
        para j = 1,2,... k
             encontrar un subgrafo completo máximo C de G
             eliminar de G todos los vértices que forman parte de C
             si G no tiene vértices
                devolver VERDADERO
             fin si
        fin para
        devolver FALSO
fin función
```

i. ¿Es cierto que cuando la estrategia devuelve `VERDADERO`, la respuesta es correcta? Justificar.
ii. ¿Es cierto que cuando la estrategia devuelve `FALSO`, la respuesta es correcta? Justificar.

SOLUCIÓN:

(a) En todos los casos definir $f(G, k) = (G^c, k)$.

(b) La estrategia golosa es equivalente a aplicar sobre G^c la estrategia del Ejercicio 11.8 de la Práctica.
  i. Sí.
  ii. No. Para encontrar un contraejemplo basta complementar un contraejemplo para la estrategia del Ejercicio 11.8 de la Práctica.

NOTA: La versión más fácil parece ser la de subgrafos disjuntos (Π_3), y la más difícil la de subgrafos maximales (Π_2).

La estrategia golosa también es aplicable a Π_1, pero no a Π_2.

TOMADO: 2001C1R2 06-AGO-2001 ($\Pi_0 \leq_p \Pi_2$), 2005C2R2 22-DIC-2005 ($\Pi_0 \leq_p \Pi_3$ y segundo punto), 2013C1R2 05-AGO-2013 ($\Pi_0 \leq_p \Pi_2$).

424. (a) Sea G un grafo. A partir de G se define el digrafo H de la siguiente manera. Por cada vértice v de G el digrafo H tiene los vértices v_{in} y v_{out} con un eje dirigido $(v_{\text{in}}, v_{\text{out}})$. Por cada eje $e = (v, w)$ de G el digrafo H tiene los ejes dirigidos $(v_{\text{out}}, w_{\text{in}})$ y $(w_{\text{out}}, v_{\text{in}})$.

    i. Demostrar que H es bipartito.

    ii. Demostrar que G es hamiltoniano si y sólo si H lo es (suponiendo que K_2 es hamiltoniano, pero K_1 no).

    (b) Demostrar que $\Pi \in$ NP-hard. ¿Qué faltaría demostrar para saber que $\Pi \in$ NP-completo? Justificar.

    Π: DIGRAFO BIPARTITO HAMILTONIANO
    Entrada: digrafo bipartito H.
    Pregunta: ¿tiene H un circuito dirigido que pasa exactamente una vez por cada uno de sus vértices?

SOLUCIÓN:

(a) i. Sea $V_{\text{in}} = \{v_{\text{in}}$ tal que $v \in V(G)\}$. Sea $V_{\text{out}} = \{v_{\text{out}}$ tal que $v \in V(G)\}$. Por definición de H no existen ejes entre vértices de V_{in} ni de V_{out}, y juntos forman una partición de $V(H)$.

ii. Los casos en que G contiene uno o dos nodos pueden analizarse por separado.
Para la ida, sea $v_1, v_2, \ldots v_n$ un circuito hamiltoniano de G. Por definición de H los n arcos $(v_{1,\text{out}}, v_{2,\text{in}}), (v_{2,\text{out}}, v_{3,\text{in}}), \ldots (v_{n,\text{out}}, v_{1,\text{in}})$ existen en ese digrafo, así como los n arcos $(v_{i,\text{in}}, v_{i,\text{out}})$ para $i = 1, 2, \ldots n$. Ambos arcos en conjunto forman un circuito hamiltoniano en H, ya que aparecen todos los vértices de H sin repetición.
Consideremos un circuito hamiltoniano C de H. Por definición de H el único arco que llega a cada v_{out} es $(v_{\text{in}}, v_{\text{out}})$, de modo que todos esos arcos aparecen en C, el cual contiene una permutación de todos ellos. Las conexiones en C entre esos arcos definen el circuito hamiltoniano de G.

(b) Basta demostrar que existe $\Pi' \in$ NP-completo tal que $\Pi' \leq_p \Pi$. A tal fin se puede utilizar el siguiente problema.

Π': CIRCUITO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices?

La transformación es la del punto anterior. Faltaría demostrar $\Pi \in$ NP.

TOMADO: 2019C1P2 05-JUL-2019.

425. Demostrar que alguno de los siguientes problemas es NP-completo suponiendo que el otro lo es y utilizando una reducción polinomial entre ambos. Indicar claramente cuál problema se intenta demostrar que es NP-completo y cuál se supone que lo es.

Π_1: CAMINO EULERIANO
Entrada: grafo G.
Pregunta: ¿tiene G un camino que pasa exactamente una vez por cada uno de sus ejes?

Π_2:

Variante 1: PRIMO IMPAR
Entrada: número primo $p \geq 3$.
Pregunta: ¿es p impar?

Variante 2: PRIMO PAR
Entrada: número primo $p \geq 3$.
Pregunta: ¿es p par?

SOLUCIÓN: Todas las instancias del problema de la Variante 1 tienen respuesta afirmativa, y todas las instancias del problema de la Variante 2 tienen respuesta negativa. Como Π_1 tiene ambos tipos de instancias, la reducción es necesariamente $\Pi_2 \leq_p \Pi_1$. Para la Variante 1 podemos definir $G = f(p) = K_2$, mientras que para la Variante 2 se puede usar $G = f(p) = 2K_2$. En general basta elegir una instancia de Π_1 que tenga respuesta afirmativa o negativa, respectivamente.

Notar que la implicación $\Pi_2 \in$ NP-completo $\Rightarrow \Pi_1 \in$ NP-completo es trivialmente cierta porque el antecedente es falso. El enunciado no admite esa demostración trivial porque exige utilizar una reducción polinomial entre los dos problemas.

Otros enunciados habituales son posibles. Además, Π_1 puede remplazarse por cualquier problema $\Pi_1' \in$ NP que tenga respuestas afirmativas y negativas, incluso un problema trivial.

El objetivo de este ejercicio es determinar si los alumnos entienden los conceptos más allá de la "trampa".

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2018C2P2 03-DIC-2018 (Variante 2).

426. Sean los siguientes problemas.

Π_1: CIRCUITO EULERIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus ejes?

Π_2: CIRCUITO HAMILTONIANO
Entrada: grafo G.
Pregunta: ¿tiene G un circuito que pasa exactamente una vez por cada uno de sus vértices?

Considerar una función f que recibe como entrada un grafo y determina en tiempo polinomial si tal grafo tiene un circuito que pasa exactamente una vez por cada eje; en caso afirmativo la función devuelve K_3, y caso contrario devuelve K_3^c. Decidir si f es una reducción polinomial de Π_1 a Π_2. En caso afirmativo demostrar; en caso negativo justificar.

SOLUCIÓN: Es una reducción polinomial porque cumple la definición de tal (verificarlo).

El problema Π_1 puede remplazarse por cualquier problema $\Pi_1' \in$ P. El problema Π_2 puede remplazarse por cualquier problema $\Pi_2' \in$ NP que tenga respuestas afirmativas y negativas, incluso un problema trivial.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2017C1P2 07-JUL-2017.

427. Sean Π_1', Π_1 y Π_2 problemas de decisión tales que $\Pi_1 \leq_p \Pi_2$, y Π_1' es igual a Π_1 salvo que cambia la respuesta en un conjunto finito de instancias. Además, los tres problemas tienen al menos una instancia cuya respuesta es afirmativa, y al menos una instancia cuya respuesta es negativa.

Variante 1: Demostrar las siguientes afirmaciones.

(a) $\Pi_1' \leq_p \Pi_2$
(b) $\Pi_1' \leq_p \Pi_1$
(c) $\Pi_1 \leq_p \Pi_1'$

Variante 2: ¿Es cierto que...

(a) $\Pi_1' \leq_p \Pi_2$?
(b) $\Pi_1' \leq_p \Pi_1$?
(c) $\Pi_1 \leq_p \Pi_1'$?

En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

TOMADO: 2019C1R2 19-JUL-2019 (Variante 2).

428. Demostrar que si $\Pi \in$ NP-completo entonces hay instancias de Π cuya respuesta es afirmativa y hay instancias de Π cuya respuesta es negativa.

SOLUCIÓN: Sea $\Pi' \in$ NP que tiene ambos tipos de instancias (una posibilidad es tomar SAT, para el cual Cook probó que es NP-completo y por lo tanto está en NP). Por absurdo supongamos que todas las instancias de Π tienen respuesta afirmativa, o todas tienen respuesta negativa. Como $\Pi' \in$ NP y $\Pi \in$ NP-completo, existe una reducción polinomial f de Π' a Π. Pero f no puede generar ambos tipos de instancias, de modo que no preserva respuestas, lo cual es absurdo.

NOTA: Propuesto originalmente por Alejandro Strejilevich de Loma.

429. Demostrar las siguientes afirmaciones.

(a) P = co-P

(b) NP-completo $\subseteq$ NP-hard

430. Sean X e Y dos conjuntos cualesquiera de problemas de decisión.

(a) Demostrar que co-(X $\cup$ Y) = co-X $\cup$ co-Y.

(b) Demostrar que co-(X $\cap$ Y) = co-X $\cap$ co-Y.

SOLUCIÓN: Para cada punto demostramos la doble inclusión.

(a)

$$\begin{aligned}
\Pi \in \text{co-}(X \cup Y) &\Leftrightarrow \Pi^c \in (X \cup Y)\\
&\Leftrightarrow (\Pi^c \in X) \vee (\Pi^c \in Y)\\
&\Leftrightarrow (\Pi \in \text{co-}X) \vee (\Pi \in \text{co-}Y)\\
&\Leftrightarrow \Pi \in (\text{co-}X \cup \text{co-}Y)
\end{aligned}$$

(b)

$$\begin{aligned}
\Pi \in \text{co-}(X \cap Y) &\Leftrightarrow \Pi^c \in (X \cap Y)\\
&\Leftrightarrow (\Pi^c \in X) \wedge (\Pi^c \in Y)\\
&\Leftrightarrow (\Pi \in \text{co-}X) \wedge (\Pi \in \text{co-}Y)\\
&\Leftrightarrow \Pi \in (\text{co-}X \cap \text{co-}Y)
\end{aligned}$$

NOTA: Basado en una pregunta de un alumno.

431. (a) Sean Π_1 y Π_2 dos problemas de decisión. Demostrar que $\Pi_1 \leq_p \Pi_2 \Leftrightarrow \Pi_1^c \leq_p \Pi_2^c$.

(b) Sea X cualquier conjunto de problemas de decisión. Demostrar que (co-X)-completo = co-(X-completo).

SOLUCIÓN:

(a) Es suficiente probar la ida ya que $(\Pi^c)^c = \Pi$. Para probarla, usar la misma reducción polinomial.

(b) Basta probar la doble inclusión. Si usamos el primer punto y las definiciones de co- y -completo, tenemos que

$$\begin{aligned}
\Pi_2 \in (\text{co-}X)\text{-completo} &\Leftrightarrow \Pi_2 \in \text{co-}X \wedge (\forall \Pi_1 \in \text{co-}X)\ \Pi_1 \leq_p \Pi_2\\
&\Leftrightarrow \Pi_2^c \in X \wedge (\forall \Pi_1^c \in X)\ \Pi_1^c \leq_p \Pi_2^c\\
&\Leftrightarrow \Pi_2^c \in X\text{-completo}\\
&\Leftrightarrow \Pi_2 \in \text{co-}(X\text{-completo})
\end{aligned}$$

Combinados II

432. Un grafo se dice planar maximal si y sólo si es simple y planar pero deja de serlo si se le agrega cualquier eje entre vértices existentes.

   Sea G un grafo planar maximal de $n \geq 3$ vértices.

   (a) Demostrar que G es conexo.
   (b) Demostrar que G tiene al menos un ciclo simple.
   (c) Mostrar que en toda representación planar de G el borde de toda región es un ciclo simple de 3 ejes.
   (d) Demostrar que en toda representación planar de G hay $r = 2n - 4$ regiones y que G tiene $m = 3n - 6$ ejes.
   (e) Demostrar que si $\chi(G) = 3$ entonces G es euleriano.
   SUGERENCIA: Por absurdo.

   SOLUCIÓN:

   (a) Informal. Si no lo fuera, podríamos aumentar la cantidad de ejes uniendo dos componentes conexas, sin perder planaridad.
   (b) Informal. Si no los tuviera sería un árbol, con una única región (la exterior). Como $n \geq 3$ entonces G no sería completo, y se le podría agregar un eje entre dos nodos no adyacentes, sin perder planaridad.
   (c) Informal. Como tiene ciclos, el borde de toda región es un ciclo. Si algún borde no fuera un triángulo, podríamos agregar ejes en la región sin perder planaridad.
   (d) Como $2m$ es la suma de todos los bordes, por el punto anterior tenemos $2m = 3r$. Además, por Euler sabemos que $n-m+r = 2$. Despejando m o r en la primera ecuación y remplazando en la segunda se obtienen las igualdades buscadas.
   (e) Por absurdo supongamos que G no es euleriano, pero como es conexo, debe tener un nodo de grado impar. Sea v tal nodo. No puede ser $d(v) = 1$ porque habría una región que no sería un triángulo, de modo que es $d(v) \geq 3$. Consideremos una representación planar de G, y en particular el nodo v con sus adyacentes, que forman una estrella con los adyacentes consecutivos unidos entre sí, porque todas las regiones deben ser triángulos. Es así que el subgrafo inducido por v y sus adyacentes es una rueda de borde impar, que requiere 4 colores, y por lo tanto G no es 3-coloreable.

   TOMADO: 2013C2P2 02-DIC-2013 (excepto el cuarto punto).

433. Sea G un grafo planar de n vértices, y sea $\alpha(G)$ la cantidad de vértices de un conjunto independiente máximo de G. Demostrar que $\alpha(G) \geq \lceil n/4 \rceil$.

   SOLUCIÓN: Según el Ejercicio 327 se cumple $n \leq \chi(G)\alpha(G)$, y como G es planar $\chi(G) \leq 4$. Por lo tanto $n \leq 4\alpha(G)$, que equivale a $\alpha(G) \geq n/4$, que a su vez equivale a $\alpha(G) \geq \lceil n/4 \rceil$ ya que $\alpha(G)$ es entero.

   TOMADO: 2018C1R2 20-JUL-2018.

434. Dado un grafo G, se definen

   - $\nu(G)$ = cantidad de aristas de una correspondencia máxima de G; y
   - $\rho(G)$ = cantidad de aristas de un cubrimiento de vértices por aristas mínimo de G.

   Sea G un grafo de n vértices.

   (a) Demostrar que $\chi(G) + 2\nu(G^c) \geq n$.
   SUGERENCIA: Equivalentemente, $\chi(G) \geq n - 2\nu(G^c)$.
   SUGERENCIA: Demostrar que G tiene un subgrafo completo de $n - 2\nu(G^c)$ vértices.
   (b) Demostrar que $\chi(G) + \nu(G^c) \leq n$.
   SUGERENCIA: Equivalentemente, $\chi(G) \leq \nu(G^c) + (n - 2\nu(G^c))$.

(c) Demostrar que si G no es completo y U es el conjunto de vértices universales de G entonces $\chi(G) - |U| \leq \rho((G - U)^c)$.
SUGERENCIA: Considerar primero el caso $U = \emptyset$.

SOLUCIÓN: Es sabido que si H es un grafo de n vértices entonces $\alpha(H) + \tau(H) = n$, y si H no tiene vértices aislados entonces $\nu(H) + \rho(H) = n$. Además $\alpha(H^c) = \omega(H) \leq \chi(H)$.

(a) Sea M una correspondencia máxima de G^c, y consideremos los $n - 2\nu(G^c)$ vértices que no son extremos de los ejes de M. Estos vértices forman un conjunto independiente en G^c, ya que caso contrario M no sería máxima. Equivalentemente, en G forman un subgrafo completo, de modo que $\omega(G) \geq n - 2\nu(G^c)$. En base a esto y las propiedades mencionadas tenemos que $\chi(G) \geq \omega(G) \geq n - 2\nu(G^c)$.
Alternativamente, usando las propiedades mencionadas y el Ejercicio 338 resulta $n = \alpha(G^c) + \tau(G^c) = \omega(G) + \tau(G^c) \leq \chi(G) + 2\nu(G^c)$.

(b) Sea M una correspondencia máxima de G^c. Los extremos de cada eje de M se pueden pintar del mismo color en G porque no son adyacentes. En total se usan $\nu(G^c)$ colores para pintar esos vértices, que son $2\nu(G^c)$ porque los ejes de M no comparten extremos. Quedan $n - 2\nu(G^c)$ vértices para pintar, lo cual puede hacerse asignando un color distinto a cada uno de ellos. La cantidad de colores usados para pintar los vértices de G es en total $\nu(G^c) + n - 2\nu(G^c) = n - \nu(G^c)$, que es al menos $\chi(G)$ por ser este valor el óptimo. Por lo tanto $\chi(G) \leq n - \nu(G^c)$ que equivale a lo que pide el enunciado.
Alternativamente, usando el tercer punto y las propiedades mencionadas con $H = (G - U)^c$ tenemos $\chi(G) - |U| \leq \rho((G - U)^c) = n - |U| - \nu((G - U)^c)$, lo que implica $\chi(G) \leq n - \nu((G - U)^c)$. Como los vértices de U son vértices aislados en G^c, resulta $\nu((G - U)^c) = \nu(G^c)$. Por lo tanto $\chi(G) \leq n - \nu(G^c)$ que equivale a lo que pide el enunciado.

(c) En $G - U$ no hay vértices universales, porque si los hubiera serían universales en G. Por lo tanto $(G - U)^c$ no tiene vértices aislados, de modo que $\rho((G - U)^c)$ está bien definido. Sea R un cubrimiento mínimo de vértices por aristas de $(G - U)^c$. Los extremos de cada eje de R se pueden pintar del mismo color en $G - U$ porque no son adyacentes. En total se usan $\rho((G - U)^c)$ colores para pintar esos vértices, que son todos los de $G - U$ porque R es un cubrimiento. Es posible que algún vértice tenga más de un eje de R que incida sobre el mismo en $(G - U)^c$; el tal caso se elige arbitrariamente el eje por el cual se le va a asignar el color. Quedan los vértices de U para pintar, lo cual puede hacerse asignando un color distinto a cada uno de ellos. La cantidad de colores usados para pintar los vértices de G es en total $\rho((G - U)^c) + |U|$, que es al menos $\chi(G)$ por ser este valor el óptimo. Por lo tanto $\chi(G) \leq \rho((G - U)^c) + |U|$ que equivale a lo que pide el enunciado.
Alternativamente, se puede hacer el camino inverso de la demostración alternativa del segundo punto.

NOTA: Propuesto originalmente por Agustín Santiago Gutiérrez (segundo punto).

TOMADO: 2017C2P2 01-DIC-2017 (excepto el primer punto).

435. Sea $G = (V, E)$ un grafo. Sea $W \neq \emptyset$ el conjunto de vértices de grado impar de G. Demostrar que si el subgrafo inducido por W tiene un circuito hamiltoniano, entonces E puede particionarse en exactamente una correspondencia y cierta cantidad de circuitos simples (eventualmente ninguno).

SOLUCIÓN: Como la sumatoria de grados es par, la cardinalidad de W es par. Si tomamos ejes alternados en el circuito hamiltoniano del subgrafo inducido por W, obtenemos una correspondencia. Si ahora eliminamos los ejes de esa correspondencia de G, obtenemos un grafo donde todos los nodos tienen grado par. Por lo tanto, cada componente conexa de ese subgrafo es euleriana, de modo que existe una partición en circuitos simples de cada conjunto de ejes. La unión de todas esas particiones, junto con la correspondencia, es una partición de E.

NOTA: Propuesto originalmente por Leopoldo Taravilse.

TOMADO: 2016C2P2 30-NOV-2016.

436. El grafo Q_d, también llamado hipercubo de orden d, se define inductivamente de la siguiente manera: $Q_0 = K_1$, y Q_d con $d \in \mathbb{N}$ es el grafo que se obtiene al tomar dos copias de Q_{d-1} y agregar un eje entre cada vértice de una copia y su vértice correspondiente en la otra copia. Por ejemplo, $Q_1 = K_2$ y $Q_2 = C_4$ (ciclo simple de 4 vértices).

(a) Demostrar que Q_3 es planar.

(b) Exhibir un grafo planar euleriano con grado mínimo 4 y sin vértices de grado 4 adyacentes entre sí. Justificar.
SUGERENCIA: Agregar vértices de grado 4 en Q_3.
SUGERENCIA: Agregar un vértice en cada región de la representación planar de Q_3.

SOLUCIÓN: Hacer lo que dicen ambas sugerencias. Los nodos originales de Q_3 quedan de grado 6 y los nuevos son de grado 4. Los nodos de grado 4 son sólo adyacentes a los de grado 6. Como el grafo resultante es conexo y todos los grados son pares, resulta euleriano. Obtener una representación planar es inmediato.

NOTA: Encontrado originalmente en Internet.

TOMADO: 2016C2P2 30-NOV-2016.

437. El grafo Q_d, también llamado hipercubo de orden d, se define inductivamente de la siguiente manera: $Q_0 = K_1$, y Q_d con $d \in \mathbb{N}$ es el grafo que se obtiene al tomar dos copias de Q_{d-1} y agregar un eje entre cada vértice de una copia y su vértice correspondiente en la otra copia. Por ejemplo, $Q_1 = K_2$ y $Q_2 = C_4$ (ciclo simple de 4 vértices).

(a) Demostrar que Q_d es trivial o bipartito.

(b) Determinar los valores de d para los cuales Q_d es planar. Justificar.

(c) Determinar $\chi(Q_d)$. Justificar.

SOLUCIÓN:

(a) Ver demostración en el Ejercicio 61.

(b) Vamos a ver que Q_d es planar si y sólo si $d \leq 3$. Para todo $i < j$ se cumple que Q_i es subgrafo de Q_j. Por lo tanto, si Q_j es planar entonces Q_i también, y si Q_i no es planar entonces Q_j tampoco (contrarrecíproca). El grafo Q_3 es planar (basta dibujarlo), de modo que también lo son Q_0, Q_1 y Q_2 (que eventualmente también podrían dibujarse). Por el primer punto Q_4 es bipartito y con suficientes nodos/ejes como para que si fuera planar se cumpla $m(Q_4) \leq 2n(Q_4) - 4$. Sin embargo, no se cumple ya que $n(Q_4) = 16$ y $m(Q_4) = 32$. Es así que Q_4 no es planar, y por lo tanto en general no lo es Q_d con $d \geq 4$.

(c) $\chi(Q_0) = 1$ ya que es el grafo trivial. Para $d \geq 1$ se cumple que Q_d es bipartito con ejes, de modo que $\chi(Q_d) = 2$.

NOTA: Para volver más difícil el ejercicio se puede suprimir el primer punto.

TOMADO: 2014C1R2 23-JUL-2014 (excepto el primer punto).

438. El grafo Q_d, también llamado hipercubo de orden d, se define inductivamente de la siguiente manera: $Q_0 = K_1$, y Q_d con $d \in \mathbb{N}$ es el grafo que se obtiene al tomar dos copias de Q_{d-1} y agregar un eje entre cada vértice de una copia y su vértice correspondiente en la otra copia. Por ejemplo, $Q_1 = K_2$ y $Q_2 = C_4$ (ciclo simple de 4 vértices).

(a) Determinar los valores de d para los cuales Q_d tiene circuito euleriano. Justificar.

(b) Demostrar que si $d \geq 2$ entonces Q_d tiene circuito hamiltoniano.

(c) Una correspondencia de un grafo se dice perfecta si y sólo si todo vértice del grafo es extremo de algún eje de la correspondencia.
Sea $f(d)$ la cantidad de correspondencias perfectas que tiene Q_d. Demostrar que $f(0) = 0$, $f(1) = 1$, $f(2) = 2$ y $f(d) > 2^{\left(2^{(d-2)}\right)}$ para $d \geq 3$.

SOLUCIÓN:

(a) Como Q_d es conexo y d-regular (prueba formal por inducción), es euleriano si y sólo si d es par.

(b) La demostración es por inducción. El caso base es $d = 2$ que es trivial ya que $Q_2 = C_4$. Para el paso inductivo, tomamos un circuito hamiltoniano de una de las copias de Q_{d-1} y su copia en la otra copia de Q_{d-1}. Luego removemos las dos copias de un mismo eje en las dos copias de los circuitos hamiltonianos de cada copia de Q_{d-1}, y unimos los extremos de los caminos (hamiltonianos) que quedan con ejes que van de una copia a la otra de Q_{d-1}. Con eso formamos un circuito que resulta hamiltoniano porque cada copia original lo era.

(c) $f(0) = 0$ porque Q_0 no tiene ejes. $f(1) = 1$ porque Q_1 tiene un único eje que incide sobre sus dos nodos. $f(2) = 2$ por análisis de casos. En Q_d podemos armar una correspondencia perfecta combinando cualquier correspondencia perfecta de una de sus copias de Q_{d-1} con cualquier correspondencia perfecta de la otra copia de Q_{d-1}. Además, hay una correspondencia perfecta formada por los ejes que unen las dos copias. Es decir, para $d \geq 1$ tenemos $f(d) \geq f^2(d-1) + 1 > f^2(d-1)$. El resto de la prueba es por inducción en d. De acuerdo a la expresión anterior para el caso base tenemos $f(3) > f^2(2) = 2^2 = 2^{\left(2^{(3-2)}\right)}$, mientras que para el paso inductivo tenemos $f(d) > f^2(d-1) > \left[2^{\left(2^{(d-3)}\right)}\right]^2 = 2^{\left(2 \times 2^{(d-3)}\right)} = 2^{\left(2^{(d-2)}\right)}$.

NOTA: Propuesto originalmente por Michel Mizrahi (un punto).

TOMADO: 2014C1R2 23-JUL-2014.

439. Sea G un grafo de m ejes.

(a) Demostrar que $2m \geq (k-1)k$, donde k es la cantidad de vértices de un subgrafo completo máximo de G.

(b) Demostrar que $2m \geq (k-1)k$, donde k es $\chi(G)$.

SOLUCIÓN:

(a) Hay un eje distinto entre cada par de nodos distintos del subgrafo completo. La cantidad de pares es $(k-1)k/2$. Otra demostración posible es usando el punto siguiente y que $\chi(G) \geq \omega(G)$.

(b) Sea V_i el conjunto de nodos pintados con el color i en un coloreo óptimo de G $(i = 1, 2, \ldots k)$. Hay al menos un eje distinto entre cada par de conjuntos distintos V_i y V_j, ya que caso contrario se podrían pintar los nodos de ambos conjuntos con el mismo color, y el coloreo no sería óptimo. La cantidad de pares es $(k-1)k/2$.

Otra forma de demostarlo es como sigue. Si G es color crítico, por el Ejercicio 10.7.b de la Práctica todo nodo v cumple $d(v) \geq k-1$, de modo que $2m = \sum_v d(v) \geq \sum_v k - 1 = (k-1)n \geq (k-1)k$. Si G no es color crítico y $k = 1$ la desigualdad se cumple trivialmente. Si G no es color crítico y $k \geq 2$, por el Ejercicio 10.6.b de la Práctica contiene un subgrafo color crítico de igual número cromático; como G tiene al menos tantos ejes como el subgrafo, para el cual ya vimos que se cumple la desigualdad, entonces la desigualdad también se cumple para G.

Otra forma de demostrarlo es usar que $2m$ es la sumatoria de grados, y luego usar argumentos como los de arriba para acotar inferiormente algunos grados.

NOTA: Propuesto originalmente por Federico Lebrón (segundo punto).

TOMADO: 2015C1P2 04-JUL-2015.

A Ejercicios mencionados de las Prácticas

3.10 Dar un algoritmo de programación dinámica para el problema de dar el vuelto con el menor número de monedas posible presentado en el ejercicio 3.6. Decir si funciona en todos los casos o sólo en algunos casos (infinito número de monedas disponibles, para monedas de cualquier valor, etc.).

3.15
a. El problema de la suma de subconjuntos consiste en, dado un conjunto $S = \{a_1, a_2, \dots a_n\}$ de números naturales y otro número natural w, determinar si existe un subconjunto S' de S tal que la suma de los elementos de S' sea igual a w. Escribir un algoritmo para resolver este problema que consista en revisar todos los subconjuntos de S para determinar si alguno suma w. Este tipo de algoritmo se llama algoritmo de "fuerza bruta".
b. Analizar la complejidad del algoritmo de a.
c. ¿Se le ocurre alguna idea mejor para resolver este problema?

5.4 Probar que en cualquier grupo de dos o más personas, hay al menos dos que tienen la misma cantidad de amigos dentro del grupo.

5.6 Probar que un grafo es conexo si y sólo si para toda partición de V en dos subconjuntos V_1 y V_2 ($V_1 \neq \emptyset, V_2 \neq \emptyset, V_1 \cap V_2 = \emptyset, V_1 \cup V_2 = V$) hay un eje de G que une un nodo de V_1 con uno de V_2.

5.7 Probar que un grafo de n vértices que tiene más de $((n-1)(n-2))/2$ aristas es conexo.

5.8
a. Probar que la unión de dos caminos simples distintos entre dos nodos p y q contiene un circuito simple.
b. ¿Qué pasa si los caminos no son simples?

5.10 Dado un digrafo G, un vértice v es alcanzable desde un vértice u si existe un camino orientado de u a v. Un digrafo se dice fuertemente conexo si dados dos vértices cualesquiera u y v, u es alcanzable desde v.

a. Probar que si un digrafo es fuertemente conexo entonces el grafo subyacente es conexo, pero la recíproca es falsa.
b. Orientar un grafo es darle una dirección a cada eje. Probar que un grafo conexo G es orientable de forma que se convierta en un digrafo fuertemente conexo si y sólo si cada eje de G pertenece a un circuito simple de G.
c. ¿Qué condición debe cumplir el mapa de calles de una ciudad para que se puedan hacer todas las calles de una sola mano?

5.19 Sean dos grafos G y H isomorfos.

a. Mostrar que G y H tienen igual cantidad de vértices.
b. Mostrar que G y H tienen la misma cantidad de aristas.
c. Mostrar que para cada $d \geq 0$ el número de vértices de grado d es igual en los grafos G y H.
d. Mostrar que para cada ℓ el número de ciclos simples de longitud ℓ en los grafos G y H es el mismo.
e. Mostrar que G y H tiene la misma cantidad de componentes conexas.
f. Mostrar un ejemplo de dos grafos no isomorfos que cumplan todas las condiciones anteriores.
g. Analizar conceptos similares para digrafos.

5.21 Un grafo G se dice autocomplementario si G y $\overline{G}$ son isomorfos.

a. Hallar un grafo autocomplementario no trivial.
b. Mostrar que cualquier grafo autocomplementario debe tener $4k$ o $4k+1$ vértices, para algún entero k.

6.2 ¿Es cierto que:

a. Si un grafo no tiene puentes entonces todo nodo tiene grado par?
b. Un grafo conexo con n ejes tiene exactamente un circuito?

c. Existe un grafo conexo con $n+1$ ejes que tiene exactamente dos circuitos?

(Tanto en b. como en c., n es la cantidad de nodos del grafo.)

6.5 Probar que todo árbol con dos o más vértices tiene al menos dos hojas.

6.6 Se dice que un grafo es un bosque si no tiene ciclos. Probar que G es un bosque si y sólo si al sacar cualquier eje de G aumenta el número de componentes conexas.

6.7 Probar que el grafo complemento de un árbol es conexo o tiene un vértice aislado y el resto forma un subgrafo completo.

6.8 Probar que la secuencia gráfica $(d_1, d_2, \dots d_n)$, $d_i \in \mathbb{N}$, es la secuencia de un árbol si y sólo si $\sum_i d_i = 2(n-1)$ y es conexo.

6.11 a. Probar que si un grafo conexo tiene un único árbol generador entonces es un árbol.

b. ¿Es posible reconstruir un grafo si se conocen todos sus árboles generadores? ¿Cómo?

c. Dar un método para determinar el número de árboles generadores de un grafo sin listarlos explícitamente.

6.22 Probar que si todos los arcos de un grafo G tienen distinta longitud entonces G tiene un único árbol generador mínimo.

8.7 Un grafo se dice arbitrariamente atravesable desde un vértice v_0 si con el siguiente procedimiento siempre se construye un circuito euleriano:

Paso 1: `Salir de` v_0 `por cualquier eje incidente`

Paso 2: `Mientras sea posible hacer`

Paso 3: `Al llegar a un vértice` u`, salir por un eje no utilizado`

Paso 4: `Informar la lista de ejes utilizados`

a. Probar que un grafo que tiene un circuito euleriano es arbitrariamente atravesable desde un vértice v_0 si y sólo si todo ciclo simple contiene a v_0.

b. Probar que si G es arbitrariamente atravesable desde v_0 entonces v_0 tiene grado máximo.

8.14 Un digrafo G se dice completamente conexo si cada par de vértices está conectado con exactamente un eje orientado en una de las dos posibles direcciones. Probar que si un digrafo G es completamente conexo entonces tiene un camino hamiltoniano orientado.

8.15 Probar que un grafo bipartito con un número impar de vértices no contiene un circuito hamiltoniano.

9.2 Probar que si T es un árbol, T es planar.

9.5 a. Probar que todo grafo planar tiene por lo menos un vértice de grado menor o igual a 5.

b. Probar que todo grafo planar con menos de 12 vértices tiene un vértice de grado a lo sumo 4.

c. Probar que si G tiene 11 o más vértices, entonces G o su complemento son no planares.

10.6 Un grafo G es color crítico si sacando un vértice cualquiera disminuye el número cromático de G.

a. Para cada grafo del ejercicio 10.1, decidir si al sacar cualquier vértice del grafo disminuye el número cromático.

b. Probar que todo grafo k-cromático ($k \geq 2$) contiene un subgrafo k-cromático color crítico.

10.7 Probar que si G es k-cromático y color crítico entonces:

a. G es conexo.

b. Todo vértice de G tiene grado mayor o igual a $k-1$.

c. G no tiene un vértice tal que al sacarlo quede un grafo disconexo.

d. G no contiene un conjunto de corte que sea un subgrafo completo de G.

10.8 El grafo junta de los grafos G y H, denotado $H + G$, se define agregando un eje entre cada vértice de H y cada vértice de G. Probar que $\chi(G + H) = \chi(G) + \chi(H)$. Concluir que el grafo junta de dos grafos color críticos es color crítico.

10.12 Probar que si G es un grafo de n vértices y $\overline{G}$ es su complemento, entonces:

a. $\chi(G) + \chi(\overline{G}) \leq n + 1$

b. $n \leq \chi(G) \times \chi(\overline{G}) \leq \left(\frac{n+1}{2}\right)^2$

c. $\chi(G) + \chi(\overline{G}) \geq 2\sqrt{n}$

10.13 Sea G un grafo regular de n vértices. Probar que si $\chi(G) + \chi(\overline{G}) = n + 1$ entonces G es uno de los siguientes grafos:

a. el grafo formado por n vértices aislados

b. K_n

c. un circuito simple de longitud impar

10.14 Analizar el siguiente algoritmo secuencial para colorear un grafo $G = (V, X)$ con $V = \{v_1, v_2, \ldots v_n\}$:

`Paso 1:` `Poner` $f(v_1) = 1$

`Paso 2:` `Para` $i = 2, 3, \ldots n$

`Paso 3:` `Poner` $f(v_i) = \min\{k : k \geq 1$ `y` $f(v_j) \neq k$ `para` $1 \leq j < i$ `y` v_j `adyacente a` $v_i\}$

a. Determinar la complejidad del algoritmo.

b. Decidir si el algoritmo siempre calcula el número cromático de un grafo. ¿De qué depende que esto suceda?

c. Dar ejemplos donde el algoritmo no encuentra el número cromático, si existen.

d. Demostrar que para todo grafo G existe un orden de rotulado de los vértices para el cual el algortimo determina el número cromático de G.

e. Deducir que si se corre el algoritmo para todas las posibles permutaciones de los rótulos de un grafo G, se obtiene un algoritmo exacto para determinar el número cromático de G. ¿Cuál es la complejidad de este algoritmo?.

10.25 Sea $m(G)$ el tamaño máximo de un matching en G.

a. Demostrar que $\chi'(G)\, m(G) \geq\mid E(G) \mid$.

b. Usando el ejercicio anterior, hallar una cota inferior para el número cromático de un grafo línea $L(G)$.

11.2 Decidir si es verdadera o falsa cada una de las siguientes afirmaciones. Si es verdadera probarla y si es falsa encontrar un contraejemplo.

a. Todo grafo tiene al menos una correspondencia.

b. Todo grafo tiene al menos un recubrimiento de vértices por aristas.

c. Todo grafo tiene al menos un conjunto independiente.

d. Todo grafo tiene al menos un recubrimiento de aristas por vértices.

e. En todo grafo, cualquier recubrimiento de vértices por aristas tiene una cantidad de elementos mayor o igual que la de cualquier correspondencia.

f. En todo grafo, cualquier recubrimiento de aristas por vértices tiene una cantidad de elementos mayor o igual que la de cualquier conjunto independiente.

g. Cualquier recubrimiento de vértices por aristas contiene una correspondencia máxima.

11.3 En este ejercicio, recubrimiento indica recubrimiento de vértices por aristas. Probar que:

a. Un grafo tiene un recubrimiento si y sólo si no tiene vértices aislados.

b. Si un grafo tiene n vértices, cualquier recubrimiento tiene al menos $\lceil n/2 \rceil$ aristas.

c. Todo recubrimiento contiene un recubrimiento minimal.

d. Un recubrimiento minimal no contiene circuitos.

e. ¿Cuál es el máximo número de aristas que puede tener un recubrimiento minimal?

f. Un recubrimiento es minimal si y sólo si no contiene caminos ni circuitos simples de longitud mayor o igual a 3.

11.5 Probar que si G es un grafo bipartito, se verifica que $m \leq \alpha\beta$, donde m es la cantidad de aristas, α es el cardinal de un conjunto independiente máximo y β es el cardinal de cualquier recubrimiento de aristas por vértices (en particular, uno mínimo).

11.7 Analizar el siguiente algoritmo para determinar un conjunto independiente en un grafo G:

Paso 1: Poner $I \leftarrow \emptyset$

Paso 2: Mientras $V \neq \emptyset$ hacer

Paso 3: Encontrar un vértice x tal que $d_G(x)$ sea mínimo

Paso 4: Poner $I \leftarrow I \cup \{x\}$

Paso 5: Poner $V \leftarrow V \backslash (\{x\} \cup \{y : y$ adyacente a $x\})$

Paso 6: Poner $G \leftarrow (V, X(V))$

Paso 7: Informar I

a. Probar que el algoritmo anterior encuentra un conjunto independiente en G.

b. Determinar la complejidad del mismo. ¿Qué técnica utiliza?

c. Mostrar que el algoritmo no necesariamente encuentra un conjunto independiente máximo.

d. Dado que no se conocen algoritmos de complejidad polinomial para el problema de determinar un conjunto independiente máximo en un grafo, se podría usar este algoritmo como algoritmo aproximado para resolver el problema? ¿Sería un "buen" algoritmo aproximado?

11.8 Analizar el siguiente algorimo para colorear un grafo G:

Paso 1: Poner $U \leftarrow V$, $k \leftarrow 0$

Paso 2: Mientras $U \neq \emptyset$ hacer

Paso 3: Poner $k \leftarrow k + 1$

Paso 4: Determinar un conjunto independiente máximo W en el subgrafo de G inducido por U

Paso 5: Asignar a los nodos de W el color k

Paso 6: Poner $U \leftarrow U \backslash W$

Paso 7: Informar k, que es el número de colores usados para colorear el grafo

a. Probar que el algoritmo produce un coloreo válido de G. ¿Es cierto que $k = \chi(G)$?

b. ¿Cuál es la complejidad del algoritmo?

c. Si el problema de encontrar un conjunto independiente máximo pudiera ser resuelto en forma polinomial, ¿esto implicaría, a partir del algoritmo anterior, que lo mismo ocurriría para el problema de coloreo de grafos?

d. Probar que reemplazando el paso 4 por el algoritmo del ejercicio 11.7, se obtiene una heurística para el problema de determinar el número cromático de un grafo, de complejidad polinomial. ¿Es una "buena" heurística?

11.9 Dado un grafo bipartito $G = (V, X)$ con $V = V_1 \cup V_2$ (todas las aristas del grafo son la forma (v, w) con $v \in V_1$ y $w \in V_2$), se dice que hay una correspondencia completa de V_1 en V_2 si existe una correspondencia C tal que todo vértice de V_1 es incidente a un arco de C.

a. Dar ejemplos de grafos bipartitos que tengan y no tengan una correspondencia completa de V_1 en V_2.

b. Probar que G tiene una correspondencia completa de V_1 en V_2 si y sólo si todo subconjunto de r vértices de V_1 es adyacente a r o más vértices de V_2, para todo r positivo.

12.13 a. Sea un grafo $G = (V, X)$ y $W \subseteq V$. Demostrar la equivalencia de las siguientes afirmaciones:

- $V - W$ es un recubrimiento de arcos de G.
- W es un conjunto independiente de G.
- W es una clique de G^c (el grafo complemento de G).

b. A partir del punto anterior, encontrar reducciones polinomiales entre los problemas Mínimo Recubrimiento de Arcos, Máximo Conjunto Independiente y Máxima Clique.

c. ¿A qué clase de complejidad pertenecen estos tres problemas? ¿Por qué?