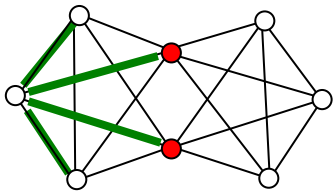
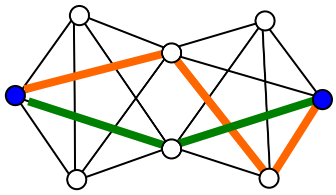
SOLUCIONES

1. **(2 puntos)**

   (a) Diseña una red que tolere el fallo de uno de sus nodos y de 3 de sus aristas. ¿Cuál debe ser su grado mínimo? Enuncia el teorema de Whitney y compruébalo para dos nodos de tu red.
   (b) Demuestra que un grafo simple G es 2-conexo si y sólo si para cada par de aristas e y e' de G existe un ciclo de G que contiene a las dos aristas.

(a) La red debe ser 2-conexa y 4-aristoconexa. El grado mínimo debe ser, por tanto, 4.
Por ejemplo la red de la figura:

Hay que eliminar al menos dos vértices (por ej. los rojos) para desconectar el grafo. Y hay que eliminar al menos 4 aristas (por ejemplo las verdes) para desconectar el grafo.

El teorema de Whitney (para un grafo 2-conexo como el de la figura) dice que para cualquier par de vértices u, v existen 2 caminos disjuntos de u hasta v. En la figura de la derecha aparecen dos caminos disjuntos entre dos de los vértices del grafo.

(b) Nos piden que demostremos una caracterización (condición necesaria y suficiente) para que un grafo sea 2-conexo. Así que debemos hacer dos demostraciones.

$\Rightarrow$) Si G es un grafo 2-conexo entonces para cada par de aristas e y e' de G existe un ciclo de G que contiene a ambas aristas. (Condición necesaria)

Construimos G' insertando un vértice x de grado dos en la arista e y otro vértice y en la arista e'. G' sigue siendo un grafo 2-conexo. Por el teorema de Whitney existen dos caminos disjuntos en G' de x a y. Es decir, un ciclo C' en G' que contiene a los vértices x y z. Suprimiendo los vértices x e y tenemos un ciclo C en G pasando por e y e'

$\Leftarrow$) Si en un grafo G para cada arista e y cada par de aristas e y e' de G existe un ciclo de G que contiene a ambas aristas, entonces el grafo G es 2-conexo. (Condición suficiente)

Lo demostramos por reducción al absurdo. Supongamos que G no es 2-conexo. Entonces hay en G un vértice-corte a. Elegimos una arista e en una componente conexa de G – v y otra arista e' en la otra componente. Así NO existe ningún ciclo que contenga a ambas aristas, en contradicción con la hipótesis de partida.

2. **(1 punto)**

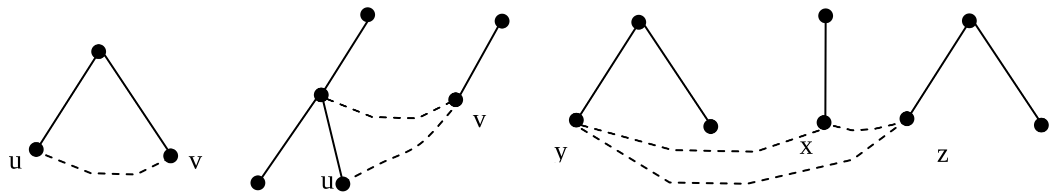
   Describe un algoritmo para detectar si un grafo tiene algún 3-ciclo. Analiza su complejidad.

(Este problema fue discutido en clase. Me ha sorprendido la cantidad de respuestas en blanco)

Primera respuesta:
Un 3-ciclo está determinado por 3 vértice adyacentes dos a dos. Dados 3 vértices u, v, w podemos determinar en tiempo O(n) si forman un triángulo, pues basta mirar, por ejemplo, en las correspondientes listas de adyacencia que pueden tener tamaño lineal. Como el número de ternas es $O(n^3)$, la complejidad de esta estrategia es $O(n^4)$

Segunda respuesta: **Matriz de adyacencia**
Vimos en el laboratorio que si A es la matriz de adyacencia del grafo, los 3-ciclos "aparecen" en la diagonal de A^3. Esto significa que si hay elementos no nulos en dicha diagonal entonces el grafo tendrá 3-ciclos. Por tanto basta calcular la matriz A^3.
Claro que primero habrá que calcular A^2. Hay que calcular n^2 elementos, cada uno de ellos nos cuesta 2n operaciones. Por tanto, $O(n^3)$ operaciones. El cálculo posterior de A^3 conlleva otras $O(n^3)$ operaciones. Por tanto la complejidad de esta estrategia es $O(n^3)$

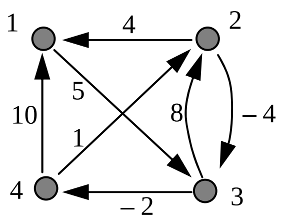
Tercera respuesta: **Búsqueda en anchura (BFS)**.
Al efectuar una búsqueda en anchura, un 3-ciclo solo puede formarse con vértices de dos niveles consecutivos. Y un 3-ciclo puede tener dos aristas, una o ninguna en el árbol de búsqueda.



En el primer caso se detecta la existencia de 3-ciclo en tiempo constante porque basta comprobar si los vértices u y v tienen el mismo padre.
En el segundo caso el 3-ciclo se detecta al analizar el vértice u y comprobar que su vecino v ya está en el árbol. En tiempo lineal se comprueba si v es vecino del padre de u.
En el tercer caso el 3-ciclo se detecta al analizar el vértice x. La arista yx se había detectado previamente desde el vértice y. Al detectar un vecino z de x, que también está en el árbol de búsqueda y en el mismo nivel que x, se debe comprobar si se cierra 3-ciclo con cada uno de los vértices del mismo nivel. Esta comprobación es lineal para cada z, pero se debe efectuar con cada vértice ya analizado del nivel y éstos pueden ser una cantidad lineal. Por tanto, el coste de la detección en este caso es $O(n^2)$

Esta tercera estrategia consiste, por tanto, en construir el árbol BFS (en tiempo $O(n^2)$) y en cada vértice comprobar si se detecta 3-ciclo según el análisis descrito en los párrafos anteriores. La complejidad total de la estrategia es $O(n^2) \cdot O(n^2) = O(n^4)$ (Se puede rebajar a $O(n^3)$ con una buena estructura de datos para el análisis del tercer caso)

3. **(1 punto)**
   Describe brevemente el algoritmo de Floyd-Warshall ¿Qué problema resuelve? Aplícalo al digrafo de la figura. Basta una iteración obteniendo la matriz W^3 a partir de la matriz W^2.



$$W^2 = \begin{pmatrix} 0 & \infty & 5 & \infty \\ 4 & 0 & -4 & \infty \\ 12 & 8 & 0 & -2 \\ 5 & 1 & -3 & 0 \end{pmatrix}$$

Se pide una descripción del algoritmo de Floyd-Warshall, ¿qué significa esto? Que con las frases que escribamos quede bien claro para qué sirve este algoritmo, a qué grafos se aplica y cuáles son los elementos básicos de su estrategia.

El algoritmo de Floyd-Warshall responde a este problema: Dado un grafo (o digrafo) con pesos en las aristas hallar el camino mínimo (y su peso) entre dos vértices cualesquiera del grafo. El algoritmo permite que haya pesos negativos y detecta ciclos de peso negativo si los hubiera.

El algoritmo construye n matrices $W^0, W^1, \dots, W^n$ de forma que en la matriz W^k aparecen los pesos de los caminos mínimos entre cada par de vértices permitiendo sólo los vértices 1, 2 … , k como vértices intermedios.

Así en la iteración 3 del algoritmo se calcula el peso del camino mínimo entre cada par de vértices permitiendo sólo los vértices 1, 2 y 3 como vértices intermedios.
Los elementos de la matriz W^3 se calculan a partir de la matriz W^2 así:

$$w^3_{ij} = \min\{\, w^2_{ij} \,,\, w^2_{i3} + w^2_{3j} \,\}$$

Por lo que dicha matriz es:

$$W^3 = \begin{pmatrix} 0 & 13 & 5 & 3 \\ 4 & 0 & -4 & -6 \\ 12 & 8 & 0 & -2 \\ 5 & 1 & -3 & -5 \end{pmatrix}$$

Por ejemplo, $w_{12}^3 = \min\{w_{12}^2, w_{13}^2 + w_{32}^2\} = \min\{\infty, 5+8\} = 13$

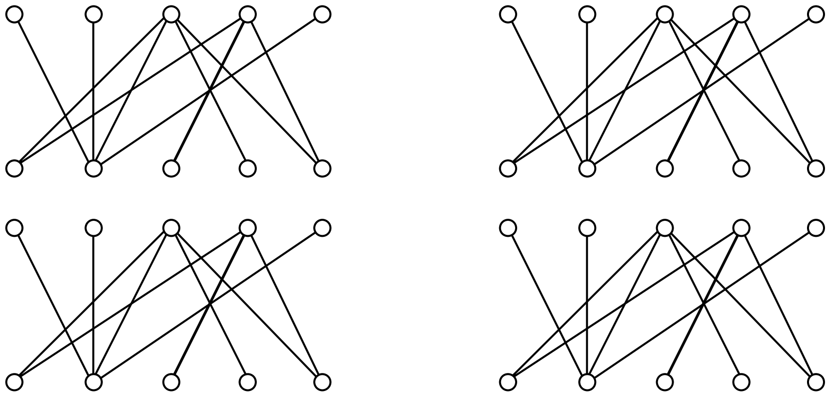
$w_{44}^3 = \min\{w_{44}^2, w_{43}^2 + w_{34}^2\} = \min\{0, -3+(-2)\} = -5$

Se observa que un término de la diagonal principal es negativo.
Esto significa que se ha detectado un ciclo negativo y el problema de caminos mínimos carece de sentido en este digrafo.

4. **(2 puntos)**

(a) Define emparejamiento y recubrimiento por vértices en un grafo.

(b) Construye un emparejamiento maximal no máximo, un emparejamiento máximo, un recubrimiento minimal no mínimo y un recubrimiento mínimo en el grafo de la figura, justificando todas las construcciones.

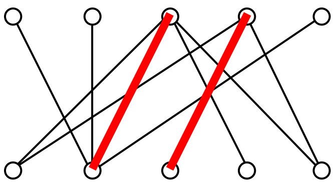


(c) Demuestra que si K* es un recubrimiento por vértices mínimo y M un emparejamiento maximal en un grafo G, entonces $|M| \le |K^*| \le 2|M|$

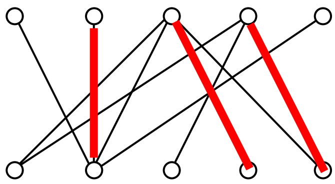
(a) Un **emparejamiento** de un grafo G = (V, A) es un **conjunto de aristas** M tal que dos aristas cualesquiera de M no tienen vértices comunes.

Un **recubrimiento** por vértices de G es un **conjunto de vértices** K tal que toda arista de G tiene alguno de sus extremos en K.
(Es asombrosa la cantidad de disparates que habéis escrito como definiciones)

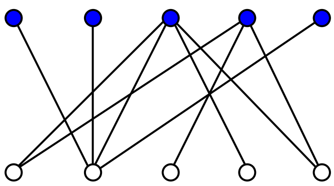
(b)



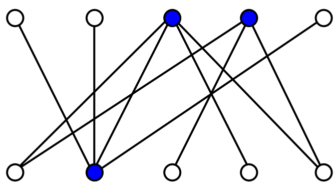
Emparejamiento maximal
(no se puede añadir ninguna arista)



Emparejamiento máximo M*



Recubrimiento minimal
(no se puede eliminar ningún vértice)



Recubrimiento mínimo K*

Como |M*| = |K*| = 3, M* es máximo y K* es mínimo, pues el cardinal de cualquier emparejamiento es menor o igual al de cualquier recubrimiento.

(c) Para cubrir las aristas de M se necesita un vértice distinto para cada una de ellas. Luego $|M| \leq |K^*|$

Por otra parte, los extremos de las aristas de un emparejamiento maximal M forman un recubrimiento K de cardinal $|K| = 2|M|$, ya que cualquier arista que no está en M tendrá un extremo en K por la maximalidad de M. Por tanto si K* es mínimo $|K^*| \leq |K| = 2|M|$

5. **(1 punto)**
Construye un emparejamiento estable (de dos formas distintas) para los conjuntos X={x,y,z,w} y A={a,b,c,d}, siendo las preferencias:

x: c > a > b > d	a: x > z > w > y
y: a > d > b > c	b: w > z > x > y
z: a > c > d > b	c: w > y > x > z
w: c > a > b > d	d: x > z > w > y

Aplicamos el algoritmo de Gale-Shapley eligiendo primero los elementos de X y luego los de A

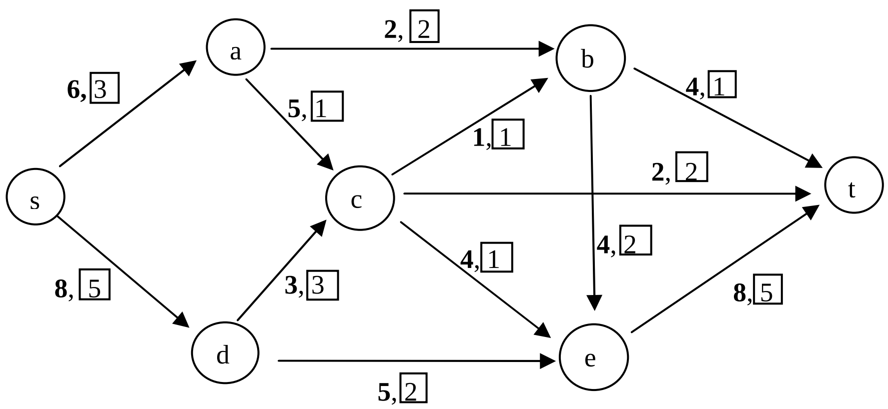
	1ª	2ª	3ª	4ª	5ª
x	~~c~~	a	a	a	a
y	~~a~~	d	d	~~d~~	b
z	a	~~a~~	~~c~~	d	d
w	c	c	c	c	c

	1ª	2ª	3ª	4ª
a	x	x	x	x
b	~~w~~	~~z~~	~~x~~	y
c	w	w	w	w
d	~~x~~	z	z	z

Así tenemos el emparejamiento estable M = {xa, yb, zd, wc} construido de dos formas diferentes.

6. **(3 puntos)**

Enuncia el Teorema de Ford-Fulkerson definiendo los conceptos que aparecen en el enunciado. **Demuestra el teorema de Ford-Fulkerson**. En la red de la figura circula un flujo f de valor 8. Las etiquetas de cada arista indican su capacidad (en negrita y en primer lugar) y el valor actual del flujo (en segundo lugar y en recuadro). Indica un camino de f-aumento en la red con arista de retroceso. Aplica el algoritmo de etiquetado para obtener un flujo de valor máximo. Comprueba el enunciado del teorema en esta red.



El teorema de Ford-Fulkerson dice que en una red de transporte N el máximo valor de un flujo coincide con la mínima capacidad de los cortes de N.

$$\max \{val(f) \mid f \text{ es un flujo en } N\} = \min \{cap(S,T) \mid (S,T) \text{ es un corte en la red } N\}$$

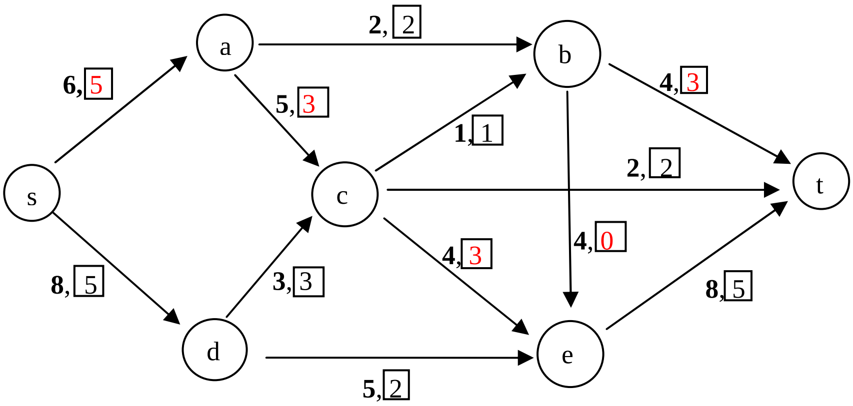
Casi todos los alumnos han olvidado leer el enunciado. La primera frase dice: “ … definiendo los conceptos que aparecen en el enunciado”. Y ni han definido red de transporte, ni flujo ni corte. Creo que sois capaces encontrar dichas definiciones en el material de la asignatura.

Recuerdo también, porque casi todos lo habéis olvidado, que demostramos el teorema de Ford-Fulkerson en dos partes. En primer lugar hay que demostrar que para todo flujo f y para todo corte (S,T) se cumple que $val(f) \leq cap(S,T)$. Y en segundo lugar se demuestra que si f es un flujo de valor máximo se puede construir un corte cuya capacidad iguala dicho valor.

De nuevo casi todos los alumnos que han intentado demostrar el teorema se han limitado a esta segunda parte olvidando la primera. ¿Será porque nadie esperaba esta pregunta en el examen?

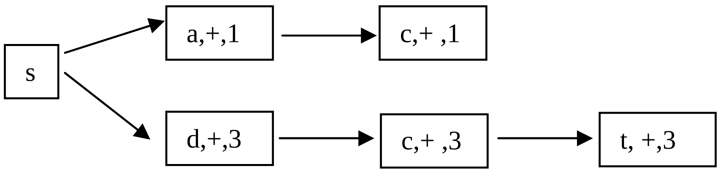
Aplicamos ahora el algoritmo de etiquetado (de Edmonds y Karp) para resolver la red del dibujo. Como tiene pocos nodos, una inspección visual nos permite detectar con facilidad un camino de f-aumento con arco de retroceso como nos piden.

El camino sacebt tiene residuo 2 y un arco de retroceso eb. Aumentamos el flujo a lo largo de ese camino obteniendo un flujo f_1 según se muestra en la siguiente figura en la que aparecen en rojo los flujos modificados:

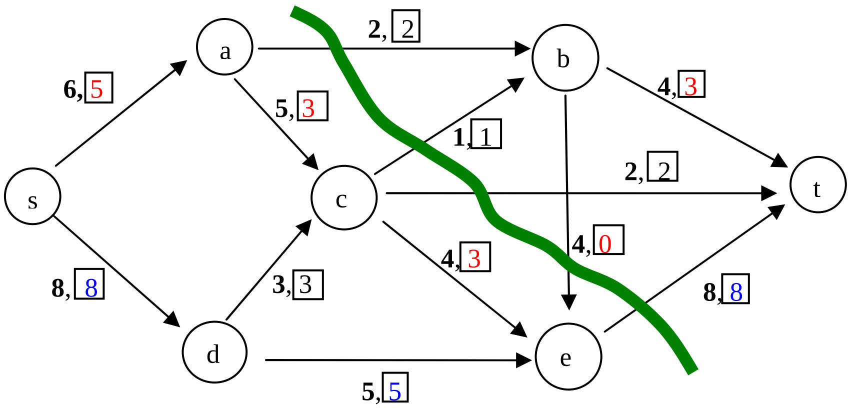


Algoritmo de etiquetado:

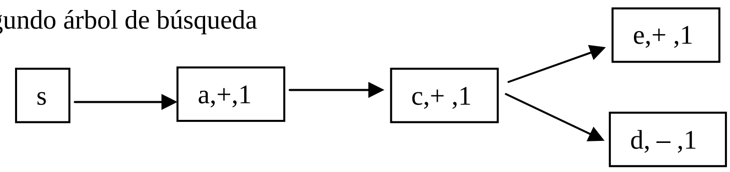
Primer árbol de búsqueda



Obtenemos el camino de f_1-aumento sdct con residuo 3. Aumentamos el flujo a lo largo del camino obteniendo un flujo f_2 que se muestra en la siguiente figura (en azul los flujos modificados)



Segundo árbol de búsqueda



Como este árbol NO alcanza el vértice destino t, el algoritmo termina, el flujo f_2 es de valor máximo y los vértices alcanzados en el árbol definen un corte de capacidad mínima (S,T) donde S = {s,a,c,e,d}. Este corte se indica en el dibujo con una línea verde.

Comprobamos que $val(f_2) = cap(S,T) = 13$ como dice el Teorema de Ford-Fulkerson.

(Algunos alumnos se han olvidado de esta comprobación obteniendo flujos IMPOSIBLES de valor 14)