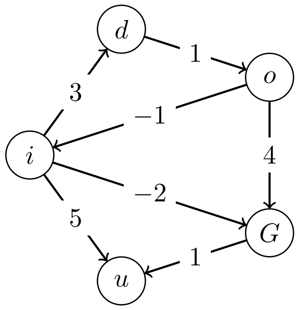
ALGORITMOS Y ESTRUCTURAS DE DATOS III - 1$^{\underline{\text{er}}}$ Recuperatorio
Fecha examen: 12-DIC-2016 / Fecha notas: a determinar

	Nº Orden	Apellido y nombre	L.U.	Cant. hojas[1]
Completar:				
	Nota (Nº)	Nota (Letras)	Docente	
No completar:				

1. 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. 2 p.



2. 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.

   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. 2 p.

3. Un grafo simple se dice k-árbol si es un árbol con k 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.

4. 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). 1 p.

   (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. 1 p.

5. 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: 2 p.

   - 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`", no se puede 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.

[1] Incluyendo a esta hoja. Entregar esta hoja junto al examen.