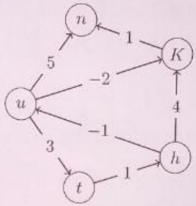
1. Utilizar el algoritmo de Ford para calcular los caminos mínimos desde el vértice u hacia todos los vértices del digrafo que aparece en la figura. Presentar pseudocódigo del algoritmo y realizar un seguimiento del mismo. Hecho esto, si se ordenan los vértices de acuerdo a su distancia al vértice inicial, podrá leerse el apellido del creador de TEX, en el cual está basado LATEX. 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.

   Sea G un grafo funcional.

   (a) Demostrar que los ciclos de G no comparten vértices. 0.75 p.

   (b) Demostrar que cada componente débilmente conexa de G tiene un único ciclo. 1.25 p.

   SUGERENCIA: Existencia usando un ejercicio visto en clase. Unicidad usando el punto anterior.

3. Sea s una cadena de caracteres de longitud n. Sea d (diccionario) una lista de p cadenas de caracteres (palabras), cada una de longitud $O(1)$. Diseñar un algoritmo que decida si s puede partirse en subcadenas que están en d. Ejemplos: 2 p.

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

   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.

4. 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 $J = G_1 + G_2$. 2 p.

   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.

5. Un grafo (simple) se dice 1-árbol si es un árbol con un eje agregado. 2 p.

   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.

   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.