ALGORITMOS Y ESTRUCTURAS DE DATOS III - 1er Parcial
Fecha examen: 17-MAY-2014 / Fecha notas: a determinar

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

1. 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 $d(v) < 5$.

2. Determinar los valores de n y h para los cuales 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) árbol binario completo de altura $h \geq 0$.

3. 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 tiene a lo sumo $n - 1$ puentes.
   (b) Demostrar que G tiene $n - 1$ puentes si y sólo si G es un árbol.
   SUGERENCIA: Para la ida demostrar primero que G es conexo.
   (c) Demostrar que todo eje de G es un puente si y sólo si G es un bosque.

4. 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).

5. Sea $G = (V, E)$ un grafo o digrafo con longitudes no negativas asociadas a sus ejes. Sean $v_1, v_2 \in V$. 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 y que tenga longitud total mínima entre todos los caminos de ese tipo. 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.

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