Examen julio 2013, preguntas

Algoritmos (Universidade da Coruña)

Examen de Algoritmos
Grado en Ingeniería Informática
3 de julio de 2013

Apellidos:	Nombre:	DNI:

1. *(1.5 puntos)* A partir de la siguiente estructura de datos para la implementación de *conjuntos disjuntos*:

```
tipo
  Elemento = entero;
  Conj = entero;
  ConjDisj = vector [1..N] de entero
```

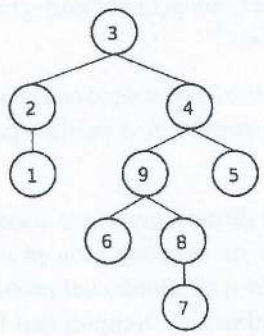
y del siguiente pseudocódigo para la unión de dos conjuntos:

```
procedimiento Unir (C, raíz1, raíz2)
          { supone que raíz1 y raíz2 son raíces }
  si raíz1 < raíz2 entonces C[raíz2] := raíz1
  sino C[raíz1] := raíz2
fin procedimiento
```

   a) Escriba el correspondiente pseudocódigo de `Buscar (C, x) : Conj`, que devuelve el nombre del conjunto (es decir, su representante) de un elemento dado.

   b) Razone cuál sería la complejidad computacional de una secuencia de m búsquedas y $n-1$ uniones.

2. *(0.5 puntos)* ¿Cómo quedaría el siguiente árbol, que representa un conjunto, tras buscar el elemento 8 usando la técnica de *compresión de caminos*?



3. *(2 puntos)* Mostrar el estado del siguiente vector tras cada una de las iteraciones de su ordenación mediante los algoritmos siguientes:

   a) ordenación de Shell con incrementos de Hibbard

   b) ordenación por montículos

10	4	9	7	8	5	11	3	0	6	1	12	2

4. *(1.5 puntos)* Ejemplos de aplicación del teorema de resolución de recurrencias *Divide y Vencerás*. Enuncie el teorema. A continuación, complete una tabla con las columnas siguientes:

   - *caso* para el que se aplica el teorema (mejor, medio, peor, o todos)
   - *caracterización* del caso (¿cuándo se produce el caso especificado en la columna anterior?)
   - *ecuación* de recurrencia (de la forma $T(n) = lT(n/b) + cn^k, n > n_0$)
   - *resultado* de la aplicación del teorema

   En cada línea de la tabla se considerará un algoritmo diferente:

   a) *bb*: búsqueda binaria

   b) *mergesort*: ordenación por fusión

   c) *quicksort*: ordenación rápida

5. *(1.5 puntos)* Escriba el pseudocódigo de un algoritmo voraz que genere una *ordenación topológica* de los nodos de un grafo dirigido acíclico. Identifique en él los elementos característicos de los algoritmos voraces. Complete su respuesta analizando el algoritmo e identificando su peor caso.

6. *(1.5 puntos)* Construya la tabla con la que podría determinarse en programación dinámica la manera óptima de pagar una cantidad de 17 unidades de valor con un mínimo de monedas, sabiendo que el sistema monetario considerado está constituido por monedas de 1, 3, 8 y 12 unidades de valor. Indique la(s) solución(es) al problema dibujando una traza en la tabla anterior para justificar cómo la(s) obtiene. ¿Porqué descartaría el uso de la técnica voraz para resolver este problema?

7. *(1 punto)* Represente mediante un grafo decorado todas las situaciones de juego que podrían alcanzarse a partir de un montón de 5 palillos para la variante del *juego de Nim* vista en clase.

8. *(0.5 puntos)* Supongamos que descubrimos una máquina de Turing determinista que resuelve el problema de la suma de subconjuntos en un número de pasos acotado por una función polinómica p(n), siendo n el tamaño del problema. ¿Qué podremos deducir entonces sobre la relación entre las clases de complejidad P y NP?