Examen. 5 de septiembre de 2.006

Resolver cada pregunta en una hoja distinta.
No hay que entregar esta hoja con el examen.

1. (2 puntos) Un árbol trie contiene la conjugación en tiempo presente del verbo ver ("veo", "ves", "ve", etc.). Mostrar la estructura del árbol resultante. Indicar cómo quedaría esa estructura implementando los nodos del trie con las dos técnicas vistas en clase: con listas de nodos y con arrays de punteros. Hacer una estimación de la memoria ocupada en cada caso, introduciendo las variables oportunas y explicando claramente la obtención del resultado. ¿Cuál es más adecuada para este ejemplo concreto?

2. (2 puntos) Se define el *índice radial* de una tabla de dispersión abierta como el número de cubetas de la tabla multiplicado por el tamaño de la cubeta con mayor número de elementos. Diseñar y escribir un algoritmo adecuado para calcular el índice radial de una tabla de dispersión. Indicar claramente la definición del tipo o tipos de datos usados para la tabla. ¿Cuál sería el índice radial óptimo en una tabla de B cubetas con n elementos en total?

3. (3 puntos) Un desfile se realiza por las calles de un pueblo. Los puntos de inicio y finalización del desfile son fijos. Queremos que el recorrido pase por el mayor número de calles posible, pero sin repetir ninguna calle ni intersección. Supongamos que hay n intersecciones entre calles y que disponemos de una matriz C de booleanos, de tamaño n x n, en la que $C[i,j]$ es cierto si existe una calle de i a j, y falso en caso contrario. Seleccionar alguno de los algoritmos de recorrido en grafos estudiados en clase y modificarlo para resolver el problema, devolviendo el camino para el desfile.

4. (2 puntos) Escribir una especificación formal, usando el método axiomático, del TAD **lista**, con al menos las siguientes operaciones:

- **colocar**: dada una lista c, un natural n y un elemento e, devolvería la lista resultante de insertar en c el elemento e en la posición n desplazando los siguientes. Si la lista tiene n o menos elementos, se inserta por el final.
- **obtener**: dado un natural n y una lista, devuelve el elemento que se encuentra en la posición n de la lista dada.
- **longitud**: devuelve el número de elementos de una lista.

Añadir los constructores y las operaciones que consideres necesarias, especificándolas de forma adecuada.

Nota: La pregunta 4 no deben hacerla los alumnos que tengan aprobada la práctica de especificaciones formales.

Examen. 5 de septiembre de 2.006

Resolver cada pregunta en una hoja distinta.
No hay que entregar esta hoja con el examen.

1. (3 puntos) Resolver las dos siguientes cuestiones:
   a) (1,5 puntos) Ordena las siguientes cotas de complejidad de menor a mayor, indicando también las que son iguales: $O(0.5\cdot 2^n)$, $O(n+1)$, $O(2^{n/2})$, $O(\log_{10} n)$, $O((n+1)^2)$, $O(n\cdot\log n)$, $O(2\log_6 n)$, $O(n^n)$, $O(3n^2)$, $O(n!)$, $O(n(n+\log n))$, $O(1)$, $O(1+1/n)$, $O(n^{n-2})$. Justifica brevemente las relaciones del orden obtenido. Indica por lo menos 6 ejemplos de algoritmos conocidos que tengan algunos de esos órdenes de complejidad. Por ejemplo, $O(n^3)$ → algoritmo clásico de multiplicación de matrices cuadradas de tamaño n.
   b) (1,5 puntos) Resuelve la siguiente ecuación de recurrencia, calculando también el valor de las constantes que puedan surgir.

      $t(n)= 5t(n/2) - 4t(n/4) + n^3/2$ Si $n>3$

      $t(n) = n-1$ Si $n<4$

      Comprobar que el resultado obtenido es correcto, verificando (por lo menos para dos valores de n) que el valor de la fórmula recurrente es el mismo que el de la fórmula no recurrente.

2. (2 puntos) Considerar que tenemos un sistema monetario con monedas de tres tipos, con valores: 2, 3 y 4. Queremos dar la cantidad 9, usando el menor número posible de monedas.

   Aplicar el algoritmo de programación dinámica visto en clase sobre el ejemplo. Deducir razonadamente la forma de la ecuación de recurrencia, con sus casos base. Mostrar la tabla resultante para este caso concreto. A partir de ella, obtener el número de monedas de cada tipo, explicando el proceso. ¿Cómo se puede saber si la solución óptima es única?

3. (2,5 puntos) En un tablero de ajedrez de tamaño **a** x **b** queremos colocar el mayor número posible de reinas. Como en el problema clásico de "las **n** reinas", no queremos que las reinas se puedan comer unas a otras. Pero, a diferencia de ese problema, permitimos que cada reina esté amenazada como máximo por otras dos reinas. Resolver el problema mediante un algoritmo voraz, encontrando una buena solución, aunque no se garantice la óptima.

4. (2,5 puntos) Resolver el problema del ejercicio 3 de forma óptima por backtracking o por ramificación y poda. Se deberán utilizar los esquemas vistos en clase, que se pueden dar por supuestos.

Nota: Los alumnos que tengan aprobada la práctica 4 pueden convalidar el ejercicio 3 o el 4. En ese caso, el ejercicio convalidado valdrá 2 puntos y el no convalidado 3.