Universidad de Valladolid

Departamento de Informática

Estructuras de Datos, 2º I.T.I. de Gestión

Examen ordinario. 8 de febrero de 2002

Apellidos	
Nombre	D.N.I.

1.- El tiempo disponible para la realización del examen es de 1,5 horas.

2.- Sólo hay una respuesta válida para cada pregunta del test.

3.- En el test, una respuesta válida suma 1/2 de punto, una respuesta errónea resta 1/6 de punto y dejar una cuestión sin responder ni suma ni resta puntos. La puntuación máxima corresponde a 8 puntos (los 2 puntos restantes corresponden a la práctica)

---

1. El propósito del apartado **ecuaciones** de un TAD es:

   ☐ Definir el modo en que se implementan las operaciones.
   ■ Definir el modo en que afectan las operaciones a su estado.
   ☐ Definir la entrada y salida de las operaciones.
   ☐ Definir el modo en que se almacenan los datos.

2. ¿Cómo se indicaría el contenido o estado de una estructura de datos mediante la notación de tipos abstractos de datos?

   ☐ Enumerando los elementos y la posición que ocupan.
   ■ Mediante la secuencia de operaciones que se han llevado a cabo sobre la estructura.
   ☐ Mediante el arbol de Huffman asociado a la estructura.
   ☐ Depende de la implementación concreta del TAD.

3. Para cualquier tamaño de la entrada, el algoritmo **A** tarda 1000 veces más que el algoritmo **B** en resolver el mismo problema. ¿Cuál de las siguientes afirmaciones es la correcta?

   ☐ **A** tiene una cota estricta menor que la de **B**.
   ■ **A** tiene la misma cota estricta que **B**.
   ☐ **A** tiene una cota estricta mayor que la de **B**.
   ☐ No se puede indicar la relación entre cotas sin conocer la constante de proporcionalidad.

4. Indique cuál de las siguientes afirmaciones es <u>falsa</u>:

   ☐ $n^2 + n \in \Omega(1)$
   ☐ $n^2 + n \in \Omega(n)$
   ☐ $n^2 + n \in \Omega(n^2)$
   ■ $n^2 + n \in \Omega(n^3)$

5. Indique cuál es el orden de complejidad de la siguiente función suponiendo que queremos contar las operaciones producto:

```
function f(n: integer) : real;
var r: real; i: integer;
begin
  r := 1.0;
  if n > 1 then
    for i := 1 to 4 do r := r*f(n div 2);
  f := r
end;
```

   ☐ $\Theta(1)$
   ☐ $\Theta(n \lg n)$
   ■ $\Theta(n^2)$
   ☐ $\Theta(4^{n/2})$

6. El algoritmo de **ordenación entrelazada** consiste en, dado un vector, se ordena recursivamente el subvector formado por los elementos en posición impar, a continuación el subvector formado por los elementos en posición par y por último todo el vector resultante de entrelazar los subvectores anteriores se ordena por el método de inserción. ¿Cuál de las siguientes estrategias ha sido utilizada para diseñar éste algoritmo?

   ■ Divide y vencerás.
   ☐ Backtracking.
   ☐ Algoritmo voraz.
   ☐ Ninguna de las anteriores.

7. Si suponemos que en algoritmo anterior las etapas de división en subvectores y entrelazamiento son $O(1)$ y la última etapa de ordenación por inserción tarda un tiempo $\Theta(n)$ en el mejor caso y $\Theta(n^2)$ en el peor. ¿Cuál de las siguientes afirmaciones sobre la complejidad <u>en todos los casos</u> de la ordenación entrelazada es la más adecuada?

   ☐ La ordenación entrelazada es $\Theta(n^2)$
   ■ La ordenación entrelazada es $\Omega(n \lg n)$ y $O(n^2)$
   ☐ La ordenación entrelazada es $\Omega(n^2)$ y $O(n \cdot \lg n)$
   ☐ La ordenación entrelazada es $\Theta(n \lg n)$

8. Indique cual de las siguientes afirmaciones es <u>falsa</u>:

   ☐ En el peor caso la ordenación rápida es más lenta (en términos asintóticos) que la ordenación por fusión.
   ☐ El método de elección del pivote en la ordenación rápida debe elegirse en función de las características de los datos.
   ■ La ordenación por montículos utiliza un espacio adicional de orden $\Theta(\lg n)$.
   ☐ La ordenación por fusión requiere un espacio adicional de orden $\Omega(n)$.

9. ¿Cuál sería la eficiencia de la ordenación por recuento aplicada a un vector de n elementos donde cada uno de ellos puede tomar uno de entre n^2 valores distintos?

   ☐ $\Theta(n)$
   ☐ $\Theta(n \lg n)$
   ■ $\Theta(n^2)$
   ☐ $\Theta(n^3)$

10. Indicar cuál de las siguientes implementaciones aplicada tanto a un TAD pila como a un TAD cola permite que las operaciones de inserción y borrado se lleven a cabo en $O(1)$.

☐ Vector lineal.
☐ Lista lineal doblemente enlazada.
■ Lista circular simplemente enlazada.
☐ Ninguna de las anteriores.

11. Se desea diseñar un TAD, que llamaremos **grupo**, que representa a una colección de enteros y donde las únicas operaciones son **insertar** y **todos_en_rango**. La operación **insertar** añade un entero al grupo, y la operación **todos_en_rango** tiene como parámetro el rango $[a,b]$ y devuelve un valor lógico indicando si <u>todos</u> los enteros del grupo son mayores o iguales que a y menores o iguales que b.

¿Cuál de las siguientes opciones implementaría de manera más eficiente la operación **todos_en_rango**?

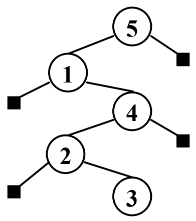
■ Lista circular ordenada.
☐ Montículo.
☐ Arbol AVL.
☐ Tabla de dispersión.

12. Si una aplicación utiliza el TAD definido en la cuestión anterior y sabemos que va a utilizar las dos operaciones con la misma frecuencia, ¿Cuál sería en este caso la implementación más eficiente para esa aplicación concreta?

Nota: Se puede suponer que el número de elementos del grupo no varía apreciablemente debido a las operaciones de inserción.

☐ Lista circular ordenada.
☐ Montículo.
■ Arbol AVL.
☐ Tabla de dispersión.

13. El arbol binario mostrado en la figura es:

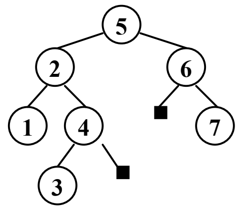


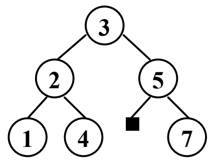
☐ Un montículo.
■ Un arbol binario de búsqueda.
☐ Un arbol AVL.
☐ Ninguno de los anteriores.

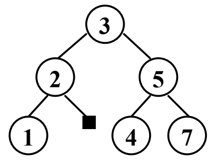
14. ¿Cuál sería el resultado de un recorrido <u>inorden</u> sobre el arbol de la cuestión anterior?

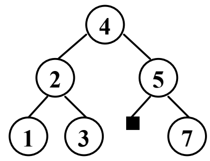
■ 1, 2, 3, 4, 5
☐ 5, 4, 3, 2, 1
☐ 5, 1, 4, 2, 3
☐ Ninguno de los anteriores.

15. Indicar cuál sería el resultado de eliminar el nodo con valor 6 del arbol AVL de la derecha:



☐


☐


■


☐ El arbol original no es AVL.

16. En una tabla de dispersión que almacena n elementos y cuyo factor de carga esta dado por el valor L. ¿Cuál es el orden asintótico del espacio utilizado por la tabla?

☐ $\Theta(1)$
☐ $\Theta(L)$
■ $\Theta(n/L)$
☐ $\Theta(n)$

FÓRMULAS ÚTILES

$$\sum_{i=1}^{n} i^k \in O(n^{k+1})$$

$$T(n) = a \cdot T(n/b) + O(n^k)$$
$$\Downarrow$$
$$\begin{cases} T(n) \in \Theta(n^k) & \text{si } a < b^k \\ T(n) \in \Theta(n^k \cdot \lg n) & \text{si } a = b^k \\ T(n) \in \Theta(n^{\log_b a}) & \text{si } a > b^k \end{cases}$$

$$T(n) = a \cdot T(n-b) + O(n^k)$$
$$\Downarrow$$
$$\begin{cases} T(n) \in \Theta(n^{k+1}) & \text{si } a = 1 \\ T(n) \in \Theta(a^{n/b}) & \text{si } a > 1 \end{cases}$$

Universidad de Valladolid

Departamento de Informática

Estructuras de Datos, 2º I.T.I. de Gestión

Examen extraordinario (especial). 7 de septiembre de 2002

Apellidos	
Nombre	D.N.I.

1.- El tiempo disponible para la realización del examen es de 1,5 horas.

2.- Sólo hay una respuesta válida para cada pregunta.

3.- Cada respuesta válida suma 8/13 de punto, una respuesta errónea resta 8/39 de punto y dejar una cuestión sin responder ni suma ni resta puntos. La puntuación máxima corresponde a 8 puntos (los 2 puntos restantes corresponden a la práctica)

---

1. El propósito del apartado **operaciones** de un TAD es:

   ☐ Definir el modo en que se implementan las operaciones.
   ☐ Enumerar y especificar los parámetros de las operaciones.
   ☐ Especificar como cambia el estado del TAD al efectuar operaciones sobre él.
   ☐ Especificar las precondiciones de las operaciones.

2. ¿Cuál es el significado de indicar que el tiempo que tarda un algoritmo esta acotado inferiormente por n^2? (n_0 y k son constantes elegidas de manera adecuada)

   ☐ Para n inferior a n_0, el tiempo crece a un ritmo mayor o igual que $k \cdot n^2$.
   ☐ Para n inferior a n_0, el tiempo crece a un ritmo menor o igual que $k \cdot n^2$.
   ☐ Para n superior a n_0, el tiempo crece a un ritmo mayor o igual que $k \cdot n^2$.
   ☐ Para n superior a n_0, el tiempo crece a un ritmo menor o igual que $k \cdot n^2$.

3. Indique cuál de las siguientes afirmaciones es verdadera:

   ☐ Si $f(n) \in O(n^2)$ entonces $f(n) \notin O(n^3)$
   ☐ Si $f(n) \in \Omega(n^2)$ entonces $f(n) \notin \Omega(n^3)$
   ☐ Si $f(n) \in O(n^2)$ entonces $f(n) \notin O(n)$
   ☐ Si $f(n) \in \Omega(n^2)$ entonces $f(n) \notin O(n)$

4. Indique cuál es el orden de complejidad de la siguiente función suponiendo que deseamos contar las operaciones suma:

```
function f(n: integer) : integer;
var i,x: integer;
begin
  x := 1;
  for i := 1 to n do x := x+i;
  f := x*f(n div 2)*f(n div 2);
end;
```

   ☐ $\Theta(n)$
   ☐ $\Theta(n \lg n)$
   ☐ $\Theta(n^2)$
   ☐ $\Theta(2^n)$

5. Indicar cuál de los siguientes pares de funciones de cota describe la complejidad de la función **g** suponiendo que deseamos contar las operaciones de suma:

```
function g(n: integer) : integer;
begin
  if n < 1 then
    g := 0
  else
    g := g(n-1)*g(n-3)+g(n-2);
end;
```

   ☐ $\Omega(n)$ y $O(n^3)$
   ☐ $\Omega(2^{n/3})$ y $O(2^n)$
   ☐ $\Omega(3^{n/2})$ y $O(3^n)$
   ☐ $\Omega(3^{n/3})$ y $O(3^n)$

6. Un programador debe desarrollar una función que calcule la raíz cuadrada de un valor real, pero el lenguaje de programación que utiliza no dispone de operadores aritméticos o funciones matemáticas, salvo las elementales (suma, resta, producto y división) y no tiene acceso a ninguna fuente de información que le proporcione un algoritmo para hacer ese cálculo. ¿Cuál de las siguientes estrategias le permitiría, a pesar de todo, realizar una función que calculase la raíz cuadrada?

   ☐ Programación dinámica.
   ☐ Divide y vencerás.
   ☐ Backtracking.
   ☐ Fuerza bruta.

7. Indique cual de las siguientes afirmaciones es falsa:

   ☐ La ordenación por fusión esta basada en una estrategia *divide y vencerás*.
   ☐ La ordenación rápida esta basada en una estrategia *divide y vencerás*.
   ☐ El elemento pivote es el valor que garantiza una partición en dos subvectores de igual tamaño.
   ☐ La ordenación por fusión es $\Theta(n \lg n)$

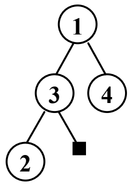
8. Se desean implementar operaciones de pila sobre un conjunto de datos almacenados en una estructura de tipo cola. Suponiendo que la cola tiene una implementación óptima y que el espacio adicional disponible es $O(1)$, indique cual sería la mejor implementación posible de las operaciones añadir y quitar de pila:

   ☐ No es posible implementar las dos operaciones de pila.
   ☐ Ambas serán $O(1)$.
   ☐ Una será $O(1)$ y la otra $O(n)$.
   ☐ Ambas serán $O(n)$.

9. Indicar cual de las siguientes afirmaciones es <u>falsa</u>:

   - ☐ Es posible crear un montículo a partir de un vector desordenado en un tiempo $O(n)$.
   - ☐ Eliminar el elemento de prioridad máxima se realiza de forma más eficiente en una lista ordenada que en un montículo.
   - ☐ Un montículo es un arbol binario completo.
   - ☐ Un montículo es un arbol binario perfectamente equilibrado.

10. El árbol binario mostrado en la figura es:

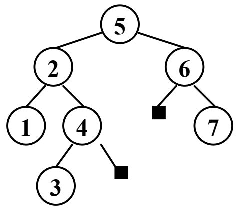


   - ☐ Un montículo.
   - ☐ Un árbol binario estricto.
   - ☐ Un árbol AVL.
   - ☐ Ninguno de los anteriores.

11. ¿Cuál sería el resultado de un recorrido <u>postorden</u> sobre el árbol de la cuestión anterior?

   - ☐ 1, 2, 3, 4
   - ☐ 4, 3, 2, 1
   - ☐ 2, 4, 3, 1
   - ☐ Ninguno de los anteriores.

12. ¿Cuántas rotaciones se deben realizar al borrar el nodo con valor 6 del árbol AVL mostrado en la figura?



   - ☐ Ninguna.
   - ☐ Una.
   - ☐ Dos.
   - ☐ El árbol no es AVL.

13. Se desea implementar una estructura de datos que almacene un gran número de elementos y las operaciones principales serán la inserción, borrado y búsqueda, estas dos últimas mediante campo clave. De entre las siguientes implementaciones, ¿Cual sería la más eficiente, suponiendo que no existen problemas para realizarla?

   - ☐ Una lista enlazada.
   - ☐ Un montículo.
   - ☐ Un arbol AVL.
   - ☐ Una tabla de dispersión.

FÓRMULAS ÚTILES

$$\sum_{i=1}^{n} i^k \in O(n^{k+1})$$

$$T(n) = a \cdot T(n/b) + O(n^k)$$

$$\Downarrow$$

$$\begin{cases} T(n) \in \Theta(n^k) & \text{si } a < b^k \\ T(n) \in \Theta(n^k \cdot \lg n) & \text{si } a = b^k \\ T(n) \in \Theta(n^{\log_b a}) & \text{si } a > b^k \end{cases}$$

$$T(n) = a \cdot T(n-b) + O(n^k)$$

$$\Downarrow$$

$$\begin{cases} T(n) \in \Theta(n^{k+1}) & \text{si } a = 1 \\ T(n) \in \Theta(a^{n/b}) & \text{si } a > 1 \end{cases}$$

Firma del alumno