



Universidad de Valladolid

Departamento de Informática

Estructuras de Datos, 2º I.T. Informática de Sistemas

Examen extraordinario.

6 de septiembre de 2006

Apellidos

Nombre

D.N.I.

- 1.- El tiempo disponible es de 1.5 horas.
- 2.- Todas las preguntas valen lo mismo y las respuestas erróneas no restan puntos.
- 3.- El examen se evaluará sobre 10, 9 u 8 puntos dependiendo del número de prácticas entregadas por el alumno

1. Al analizar la eficiencia de algoritmos, utilizamos como unidades de tiempo:

- Segundos
- Número de ciclos de reloj
- Número de operaciones elementales
- Número de llamadas recursivas

2. Indicar cuál es el orden del siguiente trozo de código suponiendo que deseamos contar las operaciones producto:

```
for i := 1 to n do
  for j := 1 to n do
    if i*j < n then
      for k := 1 to n do
        x := x+1;
```

- $\Theta(\sqrt{n})$
- $\Theta(n)$
- $\Theta(n\sqrt{n})$
- $\Theta(n^2)$

3. Indicar cuál es el orden de complejidad más ajustado para el siguiente procedimiento suponiendo que deseamos contar las operaciones producto:

```
function f(n: integer) : integer;
var x : integer;
begin
  if n <= 0 then f := 1 else
  begin
    x := f(n div 2)*f(n div 2);
    x := x + f(n div 2);
    f := 2*x;
  end
end;
```

- $\Theta(\log n)$
- $\Theta(n)$
- $\Theta(n^{\lg 3})$
- Este otro orden:

4. Indicar cuál es el orden de complejidad más ajustado para la función anterior suponiendo que ahora lo que deseamos es calcular el máximo espacio ocupado en una llamada a la función:

- $O(1)$
- $\Theta(\log n)$
- $\Theta(n)$
- Este otro orden:

5. Escriba en el recuadro siguiente una función, diseñada mediante la estrategia divide y vencerás, que calcule el producto de dos enteros positivos no nulos, a y b , utilizando únicamente sumas, restas, productos por 2 y divisiones por 2.

```
function producto(a,b: integer) : integer;
{ Prec: a > 0, b > 0 }
begin
  if b = 1 then { caso base, a*1 = a }
    result := a
  else { a*b = 2*a*b/2 }
    if odd(b) then { b impar }
      result := 2*prod(a, b div 2) + a
    else { b par }
      result := 2*prod(a, b div 2)
  end;
```

6. ¿Cuales de los siguientes algoritmos han sido diseñados mediante la técnica divide y vencerás? (Nota: Puede marcar 0,1, 2, 3 o 4 casillas)

- Búsqueda binaria
- Ordenación por fusión
- Algoritmo de Dijkstra
- Algoritmo de Floyd

7. Se desea ordenar un vector cuyos elementos se encuentran almacenados en orden inverso. Marque los algoritmos de ordenación para los cuales esta situación provoca que caigan en el peor caso respecto al tiempo de ejecución (Nota: Puede marcar 0, 1, 2, 3 o 4 casillas):

- Ordenación por inserción
- Ordenación burbuja
- Ordenación rápida con pivote primer elemento
- Ordenación rápida con pivote elemento mitad

8. Se dispone de un vector de n elementos cuya clave tiene m posibles valores distintos. Se ordena el vector mediante residuos-recuento dividiendo la clave en tres subclaves de igual número de bits. ¿Cual sería el orden del tiempo de ejecución del algoritmo?

- $\Theta(n \cdot \sqrt[3]{m})$
- $\Theta(n \log_3 m)$
- $\Theta(n \cdot m^3)$
- Ninguno de los anteriores. $\Theta(n + \sqrt[3]{m})$

9. En el problema anterior, ¿Cuál sería el orden del espacio adicional (sin contar el vector ordenado) necesario para la ejecución el algoritmo?

$$\Theta(n + \sqrt[3]{m})$$

10. ¿Cuál es el TAD para el que las operaciones fundamentales son la inserción por valor y el acceso y borrado del elemento i -ésimo?

- TAD Lista indexada
- TAD Lista ordenada
- TAD Cola de prioridad
- TAD Montículo

11. Marque aquellas representaciones en las que la operación de encontrar el *segundo* mínimo tarda un tiempo cuyo orden es menor que $\Theta(n)$, siendo n el número de elementos almacenados (Nota: Puede marcar 0, 1, 2, 3 o 4 casillas):

- Vector ordenado
- Montículo
- Árbol AVL
- Tabla de dispersión

12. Se dispone de un montículo almacenado en el siguiente vector:

1	2	3	4	5	6	7	8	9	10
10	20	50	35	30	70	65	45	55	

Escriba el contenido del vector tras realizar la operación de insertar un elemento con valor 15:

1	2	3	4	5	6	7	8	9	10
10	15	50	35	20	70	65	45	55	30

13. Se dispone de un grafo con n vértices y $\Theta(n^2)$ aristas, representado mediante matriz de adyacencia. ¿Cuál sería el orden asintótico del tiempo que tardaría el algoritmo de Dijkstra aplicado sobre éste grafo suponiendo que se ha implementado utilizando montículos?

$$\Theta(n^2 \log n)$$

14. En una tabla de dispersión cerrada con exploración lineal y borrado perezoso de tamaño 10 se realiza la siguiente secuencia de operaciones (en el orden indicado): Se insertan las claves 51, 42, 30, 70, se borran las claves 51 y 42 y se insertan las claves 10 y 12. Escriba el contenido final de la tabla (no importa si muestra o no los elementos borrados)

0	1	2	3	4	5	6	7	8	9
30	10	12	70						

Firma del alumno:

$$\begin{array}{l}
 T(n) = a \cdot T(n/b) + O(n^k) \\
 \Downarrow \\
 \left\{ \begin{array}{l} T(n) \in \Theta(n^k) \quad \text{si } a < b^k \\ T(n) \in \Theta(n^k \cdot \lg n) \quad \text{si } a = b^k \\ T(n) \in \Theta(n^{\log_b a}) \quad \text{si } a > b^k \end{array} \right.
 \end{array}
 \qquad
 \begin{array}{l}
 T(n) = a \cdot T(n-b) + O(n^k) \\
 \Downarrow \\
 \left\{ \begin{array}{l} T(n) \in \Theta(n^{k+1}) \quad \text{si } a = 1 \\ T(n) \in \Theta(a^{n/b}) \quad \text{si } a > 1 \end{array} \right.
 \end{array}$$