



## Examen de Teoría de la Programación

E. U. ING. TEC. EN INFORMÁTICA DE OVIEDO

Final septiembre – Curso 2003-2004

10 de septiembre de 2004



DNI \_\_\_\_\_ Nombre \_\_\_\_\_ Apellidos \_\_\_\_\_  
Titulación: ☐ Gestión ☐ Sistemas

3. (1 punto) El algoritmo de Strassen para multiplicar matrices lleva a cabo la multiplicación de dos matrices  $N \times N$  a través de siete llamadas recursivas que calculan el producto de dos matrices  $N/2 \times N/2$ . La complejidad del algoritmo sin tener en cuenta las llamadas recursivas es cuadrática.

a) ¿Cuál es la complejidad del algoritmo de Strassen? (justifica la respuesta)

b) Si puedes elegir entre el algoritmo clásico de multiplicación de matrices o el de Strassen. ¿Cuál elegirías si tuvieses que resolver una matriz de muchos elementos? ¿Y si tuvieses pocos elementos? (razona la respuesta)

4. (2 puntos) Tenemos un vector  $v$  de  $n$  números enteros positivos. Queremos resolver el siguiente problema: formar 3 subconjuntos disjuntos, a partir del vector inicial, cuyos elementos sumen el mismo valor y que este valor sea mayor que cero. Por tanto, cada elemento del vector inicial sólo puede pertenecer a uno de los subconjuntos solución o a ninguno de ellos.

Basándose en la técnica de *backtracking* (vuelta atrás), escribir, en Java, el método principal para el *backtracking*, su llamada y las declaraciones necesarias para encontrar una solución para el problema propuesto.

5. (2 puntos) El jefe de un proyecto debe publicar documentación en papel sobre el producto que están finalizando. Pide presupuestos a cuatro imprentas sobre el coste de las distintas operaciones que conlleva la publicación, además quiere disponer de la publicación en el plazo más breve posible por lo que decide realizar cada una de las operaciones en paralelo encargándoselas cada una a una empresa diferente. Lógicamente también quiere tener la documentación con el mínimo coste.

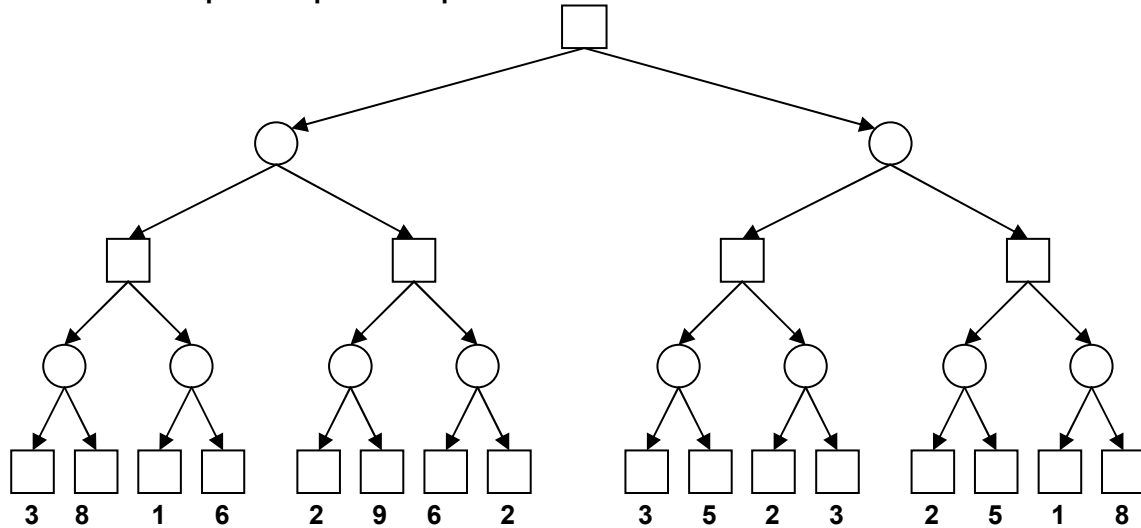
	Fotocopias b/n	Encuader- naciones	Fotocopias en color	Impresión libro
Imprenta A	11	12	18	40
Imprenta B	14	15	13	22
Imprenta C	11	17	19	23
Imprenta D	17	14	20	28

Utilizando la técnica de *ramificación y poda*; se pide resolver los siguientes apartados teniendo en cuenta los datos de la tabla anterior:

a) Describir en lenguaje natural los heurísticos de ramificación y poda adecuados.

b) Dibujar el árbol de estados representando pesos y estados podados hasta alcanzar la primera solución.

c) Representar el árbol de estados representando pesos y estados podados hasta alcanzar la solución final óptima al problema planteado.



6. (1,5 puntos) Desarrollar la poda  $\alpha$ - $\beta$  para conocer que jugada debe realizar el jugador MAX.

- Rellenar los nodos que haya que desarrollar,
- Escribir las cotas  $\alpha$  y  $\beta$  al lado del nodo correspondiente,
- Marcar los cortes con una cruz e indicar de que tipo son,
- Por último, indicar que jugada debe elegir MAX para situarse en la mejor posición posible.

Notas: El jugador que realiza el primer movimiento en el árbol es MAX. Los nodos del árbol se desarrollan de izquierda a derecha.

Ejemplo de indicaciones:

