

# Algorítmica.

## 3º de Grado en Informática.

### Febrero de 2013.

**Tiempo: 2 horas 30 minutos.**

#### **Preguntas cortas.**

1. ¿De qué factores depende la eficiencia o complejidad de un algoritmo?. ¿Cómo influyen entre sí?. Indica algún ejemplo donde se ponga de manifiesto esta influencia mutua.
2. Si implementamos un algoritmo en dos ordenadores distintos, ¿podría variar su orden de complejidad?. Justifica tu respuesta.
3. ¿Porqué el algoritmo del quicksort se basa en el método divide y vencerás?
4. Si tuvieses que almacenar en una cinta de cassette una serie de canciones de distinta duración, ¿En qué orden las almacenarías?. Justifica tu respuesta. Recuerda que para escuchar una canción de han de recorrer todas las anteriores.
5. Enuncia el principio de optimalidad de Belman. ¿Se cumple siempre este principio?. Indica un ejemplo que justifique tu respuesta.
- 6. ¿Cuales son los dos aspectos más importantes que influyen en la eficiencia de un algoritmo que se resuelva usando el método del Backtracking?.

**Nota:** Todas las preguntas puntúan igual.

#### **Preguntas largas.**

1. Calcula el orden de complejidad del siguiente algoritmo.

**Algoritmo A(n; ;)**

**inicio**

**si  $n < 2$  entonces**

**devolver 1**

**sino**

**devolver  $n * A(n / 2; ;)$**

**finsi**

**fin.**

2. Se sabe que dos métodos de ordenación son de orden  $O(n^2)$  y ambos los tenemos implementados. Explica detalladamente que pasos seguirías para determinar cual de ellos es más eficiente desde el punto de vista temporal. ¿Cómo podrías obtener tiempos aproximados para valores de n lo suficientemente grandes como para que el vector no pudiese ser almacenado en memoria interna?.
3. Algoritmo de las torres de Hanoi. Indica de manera intuitiva (sin demostrar) cual es su orden de complejidad. ¿Que ocurre para un número de discos grande.
4. Describe el algoritmo que implementaste en la práctica 3 (Ordenación por fusión o multiplicación de enteros grandes). ¿Porqué se basan en la técnica del divide y vencerás?.
5. Resuelve el problema del cambio usando un método voraz y programación dinámica para el caso siguiente: Cambio = 8, valores de las monedas = {1, 4, 5}. Indica paso a paso las monedas que se obtienen de cada tipo al usar la programación dinámica.
6. Describe el algoritmo probabilista para resolver el problema de las n-reinas.

**Nota:** Todas las preguntas puntúan igual.