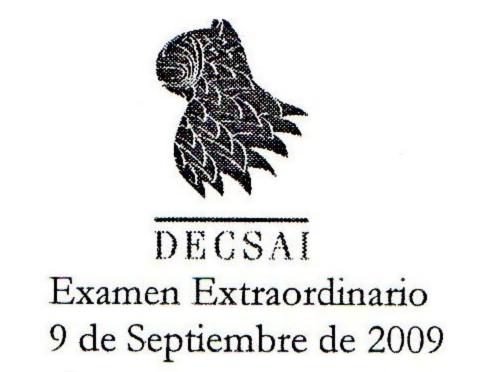


Teoría de Algoritmos 2º de Ingeniería Informática



1) Demostrar la veracidad o falsedad de la siguiente proposición: Sea b un entero ≥ 2,

$$f: N \rightarrow R^+$$

una función no decreciente a partir de un n_0 (f se llama eventualmente no decreciente) y tal que f(bn) está en O(f(n)) (se dice que f es b-armónica) y

$$t: N \rightarrow R^+$$

eventualmente no decreciente. Entonces, si $t(n) \in \Theta(f(n) \mid n=b^k)$ se cumple que t(n) está en $\Theta(f(n))$.

2) La siguiente recurrencia,

$$T(n) = \begin{cases} f(n) & \text{si } 0 \leq n < c \\ \frac{1}{2} \cdot T(n-c) + b \cdot n^k & \text{si } c \leq n \end{cases}$$

en la que $f(\cdot)$ es una función arbitraria, corresponde al tiempo de ejecución de cierto algoritmo. Resolverla para conocer el orden de dicho algoritmo cuando el valor de "a" es

- a) menor que uno,
- b) igual a uno y
- c) mayor que uno.
- 3) Razone si es posible o no, demostrándolo en cualquier caso, aplicar la técnica de Programación Dinámica al calculo de la sucesión de Fibonacci. En caso afirmativo, compruebe el Principio de Optimalidad de Bellman, y establezca la correspondiente ecuación de recurrencia.
- 4) Definir algoritmo de precomputación. Calcular los tiempos de ejecución de los algoritmos de precomputación que conozca para la evaluación repetida de polinomios, y explique en que consisten. Sobre esa base, explicar la importancia de las ecuaciones recurrentes en Teoría de Algoritmos.
- 5) Algoritmo de Floyd-Warshall. Explique el problema que resuelve y por que lo resuelve. Comente además la diferencia de ese algoritmo con el de de Floyd.

Notas Importantes

- 1. Tiempo total para el examen: Dos horas
- 2. Pueden quedarse con esta hoja