Algoritmos y Estructuras de Datos
Tarea 2

Profesor: Sergio Rajsbaum
Ayudante: Francisco Javier Escalona González

fecha de hoy: 9 de septiembre 2004
fecha de entrega: 21 de septiembre 2004

— explica en detalle y con claridad <u>todas</u> tus respuestas —

— Tus algoritmos deberán ser lo más eficiente posibles —

— explica el funcionamiento de tus algoritmos informalmente, luego escribe el código, y luego demuestra correctez y complejidad.—

Se permite trabajar en equipos de dos personas.
Pero cada uno debe entregar la tarea resuelta por separado, e indicar el nombre de su compañero de equipo.

Tema: Recursión, Orden de Crecimiento

1. Escribe un resumen de no mas de dos cuartillas de las páginas 89 a la 101 del libro *Out of their Minds* de Shasha y Lazere, acerca de la vida y obra de Knuth.

2. Describe en tus propias palabras, añadiendo tus comentarios y opiniones, el "War Story: Mystery of the Pyramids," páginas 43 a la 45 del libro *The Algorithm Design Manual* de Skiena.

3. Diseña un algoritmo recursivo que recibe una lista $A[1], A[2], \ldots, A[n]$ que contiene enteros no ordenados, y recibe un número x. Como salida debe contestar si x esta o no en A, y en caso positivo, decir en que posición de A se encuentra. El algoritmo debe partir la lista en dos, y recursivamente buscar a x en cada parte. Discute brevemente la relación entre este algoritmo, el de busqueda secuencial, y el de busqueda binaria (cual es mejor, por que, etc).

4. Demuestra como se relacionan las siguientes funciones en cuanto a orden de crecimiento (O, Ω, Θ, o, ω).

   (a) $\log\log n$ con $\frac{\log n}{\log\log n}$
   (b) 2^n con 3^n
   (c) $\log 2^n$ con $\log 3^n$
   (d) $n^{0.5}$ con $n/logn$

5. Para cada una de las siguientes preguntas, explica brevemente su contestación.

   (a) ¿Si pruebo que un algoritmo toma tiempo $O(n^2)$ en el peor caso, es posible que tome $O(n)$ en algunas entradas?

(b) ¿Si pruebo que un algoritmo toma tiempo $O(n^2)$ en el peor caso, es posible que tome $O(n)$ en todas las entradas?

(c) ¿Si pruebo que un algoritmo toma tiempo $\Theta(n^2)$ en el peor caso, es posible que tome $O(n)$ en algunas entradas?

(d) ¿Si pruebo que un algoritmo toma tiempo $\Theta(n^2)$ en el peor caso, es posible que tome $O(n)$ en todas las entradas?

(e) ¿Es cierto que la función $f(n) \in \Theta(n^2)$, donde $f(n) = 100n^2$ para n par y $f(n) = 20n^2 - n \log n$ para n impar?

No se aceptan tareas despues de la fecha límite.

Para cualquier duda contactar a Paco o a Sergio con suficiente anticipación:
pax@ciencias.unam.mx, rajsbaum@matem.unam.mx