Programación Avanzada – Notación Asintótica

Tarea 6

Fecha de entrega: 13 de marzo, 2015

1. Considera el problema de ordenar n números almacenados en un arreglo A encontrando primero el elemento más pequeño en A e intercambiándolo por el primer elemento almacenado en el arreglo: $A[1]$. Luego, encontramos el segundo elemento más pequeño en A y lo cambiamos con $A[2]$. Continuamos de esta forma para los primeros $n-1$ elementos de A. Escribe el pseudocódigo para este algoritmo, conocido como **selection sort**. ¿Cuál es la invariante al ciclo que mantiene este algoritmo? Da los tiempos de cálculo en términos de la notación Θ.

2. Utiliza inducción matemática (método de sustitución) para mostrar que cuando n es una potencia exacta de 2, la solución a la recurrencia:

$$T(n) = \begin{cases} 2 & \text{if } n = 2, \\ 2T(n/2) + n & \text{if } n = 2^k, \text{ for } k > 1 \end{cases}$$

   es $T(n) = n \lg n$.

3. Podemos expresar **insertion sort** como un procedimiento recursivo como sigue: Para ordenar $A[1 \ldots n]$, ordenamos recursivamente $A[1 \ldots n-1]$ y luego insertamos $A[n]$ en el arreglo ordenado $A[1 \ldots n-1]$. Escribe la recurrencia para expresar el tiempo de cálculo de esta versión recursiva de **insertion sort**.

4. Muestra que para cualquier constantes a y b, donde $b > 0$,

$$(n+a)^b = \Theta(n^b).$$

5. Clasifica las siguientes funciones por orden de crecimiento; es decir, encuentra un arreglo $g_1, g_2, \ldots, g_{24}$ de las funciones que satisfacen $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots g_{29} = \Omega(g_{24})$. Divide tu lista en clases de equivalencia, tal que las funciones $f(n)$ y $g(n)$ estén en la misma clase si y solo si $f(n) = \Theta(g(n))$,

n	2^n	$(\sqrt{2})^{\lg n}$	n^2	$n!$	$(\lg n)!$
$\left(\frac{3}{2}\right)^n$	n^3	$\lg^2 n$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n}$
$\ln \ln n$	$n \lg n$	$n \cdot 2^n$	$n^{\lg \lg n}$	$\lg n$	1
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$4^{\lg n}$	$(n+1)!$	$\sqrt{\lg n}$