# Tarea 1: Complejidad Computacional

Profesores: Sergio Rajsbaum y Ricardo Strausz
Ayudante: Sebastián Bejos

Fecha de entrega: 24 de febrero 2009

1. ¿Es $O(2^n) = O(2^{O(n)})$? Justifica plenamente tu respuesta.

2. Considerando cualquier función cuadrática de la forma $f(n) = an^2 + bn + c$, donde $a$, $b$ y $c$ son constantes, y $a > 0$. Desechando los términos de grado más bajos $f(n) = \Theta(n^2)$. Formalmente si tomamos $c_1 = a/4$, $c_2 = 7a/4$ y $n_0 = 2[max((|b|/a), \sqrt{(|c|/a)})]$ entonces

$$0 \leq c_1 n^2 \leq an^2 + bn + c \leq c_2 n^2$$

para toda $n \geq n_0$, por lo que $f(n) = \Theta(n^2)$.

Verifica que la demostración sea correcta y muestra como se obtuvieron los valores de $c_1$, $c_2$ y $n_0$.

3. Sea

$$p(n) = \sum_{i=0}^{d} a_i n^i,$$

un polinomio de grado $d$ en $n$, donde $a_d > 0$, y sea $k$ una constante. Utilizando las definiciones de notación asintótica pruebe las siguientes propiedades.

*a*) Si $k \geq d$, entonces $p(n) = O(n^k)$.

*b*) Si $k \leq d$, entonces $p(n) = \Omega(n^k)$.

*c*) Si $k = d$, entonces $p(n) = \Theta(n^k)$.

*d*) Si $k > d$, entonces $p(n) = o(n^k)$.

*e*) Si $k < d$, entonces $p(n) = \omega(n^k)$.

4. ¿Cual es el número de cadenas binaras de tamaño $n$, que no tienen dos ceros consecutivos? Prueba tu respuesta.

5. Asumiendo que $L$ es un lenguaje cerrado bajo la operación de concatenación. Prueba que $L^+ = L$. Además especifica cuando $L^* = L$.

6. Probar que cualquier número $n$, tal que $n \bmod 4 = 3$, tienen un factor primo $p$ tal que $p \bmod 4 = 3$. Prueba también que hay infinitos números primos $p$ con $p \bmod 4 = 3$.

7. Esboza la prueba de que cualquier función *RAM* computable es también Turing computable.

8. Esboza de igual manera la prueba de que la función de Ackerman es *RAM* computable.

9. Escribe un resumen crítico sobre los artículos de Turing, Wegner y Davis.[1] El resumen de cada uno, no debe ser de más de una hoja.

[1] Los artículos los puedes encontrar en http://atenea.matem.unam.mx/moodle dentro del curso de Teoría de la Complejidad Computacional