Algoritmos y Complejidad

1.- Clasifica dentro de los conjuntos $O\,(f\,)$ notables las siguientes funciones:

a) $f\,(n) = 2n^2 - 3n + 7$; b) $f\,(n) = 3\log_b n + 34$; c) $f\,(n) = e^n + n^3 - n$; d) $f\,(n) = \dfrac{n\log n + n^2}{8}$

e) $f\,(n) = \sum_{i=1}^{n} i + n^4$; f) $f\,(n) = \sum_{i=1}^{n} i^2 + \sum_{i=1}^{n} i + n^4$

2.- Demuestra que $O\,(c^n) \subset O\,(n!\,)$.

3.- Sean f, g funciones de variable natural y supongamos que $f \in q(n^d)$, con d entero postivo, y que $g \in q(c^n)$, con c constante mayor que 1.

a) Demostrar que entonces $f \in O(g)$, comprobando que $f\,(n) < g(n)$ para n suficiente mente grande .

b) Si $f\,(n) = 10^6\, n^{10^6}$ y $g(n) = (1.000.001)^n$, encontrar el valor de n a partir del cual $f\,(n) < g(n)$.

4.- Sea $G = (V,E\,)$ un grafo simple tal que Card(V) = n y denotemos por $E(n\,)$ el número de aristas de G.

a) Demostrar que $E(n\,) \in O(n^2\,)$.

b) Si G es conexo, demostrar entonces que $E(n) \in W\,(n\,)$.

c) Sabiendo que el algoritmo de Hierholzer tiene complejidad $O(E(n\,))$, razonar porqué el problema de encontrar un ciclo euleriano en un digrafo euleriano tiene complejidad $q\,(E(n))$.

5.- ¿De cuántas formas se puede escribir un número entero no negativo n como suma ordenada de unos y doses? Sugerencia: si llamamos T_n a esta cantidad, para n = 4 por ejemplo se tiene que T_4= 5, ya que

4 = 1+1+1+1 = 2+1+1 = 1+2+1 = 1+1+2 = 2+2

6.- Resolver la relación de recurrencia $T_n = n\,T_{n-1},\ n \ge 1,\ T_0 = 1$.

7.- Calcular el orden de complejidad de un algoritmo cuya función de coste sea:

a) $T(n) = \begin{cases} n^2 & \text{si} \quad 0 \le n < 2 \\ T\left(\dfrac{n}{2}\right) + n^2 & \text{si} \quad n \ge 2 \end{cases}$; b) $T(n) = \begin{cases} n^4 & \text{si} \quad 0 \le n < 5 \\ 3T(n-5) + n^4 & \text{si} \quad n \ge 5 \end{cases}$;

c) $T(n) = \begin{cases} c & \text{si} \quad 0 \le n < 2 \\ 3T\left(\dfrac{n}{2}\right) + c & \text{si} \quad n \ge 2 \end{cases}$; d) $T(n) = \begin{cases} 0 & \text{si} \quad n = 0 \\ 5 & \text{si} \quad n = 1 \\ 3T_{n-1} + 4T_{n-2} & \text{si} \quad n \ge 2 \end{cases}$;

e) $T(n) = \begin{cases} n & \text{si} \quad n < 3 \\ 5T_{n-1} - 8T_{n-2} + 4T_{n-3} & \text{si} \quad n \ge 3 \end{cases}$

6.- (*La multiplicación del campesino ruso*) A continuación describiremos un algoritmo para multiplicar dos números enteros n y m que utilizaban los campesinos rusos:

Multiplicacción del campesino ruso

Forma dos columnas con m y n como primeros elementos
REPITE la siguiente secuencia:

- divide por 2 el último número de la primera columna y escribe debajo el cociente de esta división despreciando el resto
- multiplica por dos el último número de la segunda columna y es cribe debajo el resultado

HASTA QUE el último número de la primera columna sea 1.
Para cada número par que aparezca en la primera columna, borrar el correspondiente de la segunda columna. Ahora sumar los números no tachados en la segunda columna. Su suma es el producto de m por n.

Por ejemplo para hacer 18×37:

$$
\begin{array}{rr}
18 & \cancel{37} \\
9 & 74 \\
4 & \cancel{148} \\
2 & \cancel{296} \\
1 & \underline{592} \\
 & 666
\end{array}
$$

Determinar dependiendo m y n cuántas operaciones elementales (multiplicaciones por dos, divisiones por dos y sumas) requiere este método. Compararlo con el producto usual en base diez.