



Algorítmica  
Grado en Ingeniería Informática

**Prácticas: Semana 1**  
**Eficiencia: órdenes de eficiencia**

1. Se dispone de un programa formado por dos bloques de código A y B, que se ejecutan de forma consecutiva, y cuyos tiempos de ejecución en instancias del caso peor vienen dados por las expresiones:

$$T_A(n) = 19 \cdot n^3 + n$$
$$T_B(n) = \log_2(n) + n^4$$

Para instancias del caso mejor, el tiempo de ejecución de cada bloque de código es:

$$T_A(n) = 400 \cdot n + \log_2(n)$$
$$T_B(n) = 1235 \cdot n^2$$

Se pide responder a las siguientes cuestiones:

- Haciendo uso de las propiedades de los órdenes de eficiencia, indique cuál es la eficiencia del programa.
  - ¿Podría decirse que el programa es  $O(n^5)$ ?
  - ¿Tiene el algoritmo orden exacto?
2. Existen diferentes algoritmos que resuelven un problema dado, con eficiencias respectivas en el caso peor de  $O(n^5)$ ,  $O(n!)$ ,  $O(2^n)$ ,  $O(2^{n+1})$ ,  $O(1.5^n)$ ,  $O(n \cdot \log(n))$ ,  $O(\log(n))$ . Ordene, de mejor a peor, los órdenes de eficiencia e indique cuál o cuáles son los mejores y peores algoritmos en el caso peor.
  3. **(código de ejemplo ejercicio3.cpp)** Se trata de resolver el problema de buscar la posición de un elemento en un vector que ya está ordenado. Se dispone de dos algoritmos A y B que resuelven el problema, con eficiencia  $A \in O(f(n))$  y  $B \in O(g(n))$ , donde  $f(n) = n$  y  $g(n) = \log_2(n)$ . En ambos algoritmos, el caso peor se da cuando el elemento a buscar no se encuentra en el vector. Se pide:
    - Calcular la constante K del orden de eficiencia en ambos casos.



- Mostrar gráficamente, para varios tamaños de casos, el tiempo de ejecución real de cada algoritmo frente al orden teórico-práctico estimado  $K_1 \cdot f(n)$  para A, y  $K_2 \cdot g(n)$  para B (puede usar Libreoffice Calc, Office Excel, GNUPlot, o cualquier software de visualización de gráficas para ello). ¿Qué irregularidad observa en el algoritmo B? ¿A qué puede deberse?
  - Indique, a nivel teórico, cuál es el mejor algoritmo para resolver el problema, comparando los órdenes de eficiencia de ambos.
  - ¿Cuánto tardaría (como mucho) el algoritmo A, atendiendo a la constante K calculada y el orden de eficiencia del algoritmo, en resolver una instancia del problema de tamaño **40000000**?
4. Se dispone de dos algoritmos de ordenación: Burbuja y Selección, ambos con eficiencia  $O(n^2)$ . Generando varios vectores aleatorios a ordenar para diferentes tamaños de casos, se pide:
- Calcular la constante K del orden de eficiencia en ambos casos.
  - Mostrar gráficamente, para varios tamaños de casos, el tiempo de ejecución real de cada algoritmo frente al orden teórico-práctico estimado  $K_1 \cdot n^2$  para Burbuja, y  $K_2 \cdot n^2$  para Selección, usando la misma herramienta de generación de gráficas del ejercicio anterior.
  - Indique, a nivel teórico, cuál es el mejor algoritmo para resolver el problema, comparando los órdenes de eficiencia de ambos.
  - Indique, a nivel práctico, cuál es el mejor algoritmo, justificando su respuesta en términos del orden y de las constantes ocultas calculadas.