

Algorítmica
Hoja 1 de problemas
Introducción a la Teoría de la Complejidad

Diseñar, expresar en pseudocódigo y estudiar la complejidad del algoritmo clásico para resolver los siguientes problemas:

1. Sumar dos matrices cuadradas de $n \times n$ componentes.
2. Calcular la traza de una matriz cuadrada de $n \times n$ componentes (la traza es la suma de los elementos de la diagonal principal).
3. Decidir si una matriz cuadrada de $n \times n$ componentes es simétrica (esto es, si $a_{ij} = a_{ji}$ para todos i, j). Ser eficiente, es decir, salir sin terminar el recorrido si se conoce que no lo es.
4. Multiplicar dos matrices cuadradas de $n \times n$ componentes.
5. Comparar si dos vectores de n componentes son iguales.
6. Calcular el producto escalar de dos vectores de n componentes.
7. Obtener la suma de los n primeros múltiplos de k para un k dado.
8. Decidir si un número entero n es primo.

Resolver los siguientes problemas:

9. Un algoritmo es capaz de resolver todos los casos de tamaño n de un problema en tiempo $t(n)$ segundos en una computadora C_1 . Si se compra una computadora C_2 que es 1000 veces más rápida, averiguar la ganancia de tamaño de problema que se logra resolver en C_2 si:
a) $t(n) = 2^n$ $t(n) = n^3$ $t(n) = n^k$ $t(n) = \log n$
10. Dado un problema resoluble por medio de los algoritmos A_1 y A_2 en tiempos $t_1(n) = n^2$ segundos y $t_2(n) = n$ horas, averiguar el tiempo requerido de cómputo para que A_2 sea más rápido que A_1 .
11. Encontrar un contraejemplo para la siguiente afirmación: $f(n) \in O(s(n))$ y $g(n) \in O(r(n))$ implica que $f(n) \cdot g(n) \in O(f(n) \cdot g(n))$.
12. Encontrar un contraejemplo para la siguiente afirmación: $f(n) \in O(s(n))$ y $g(n) \in O(r(n))$ implica que $f(n)/g(n) \in O(f(n)/g(n))$.
13. ¿Cuáles de las siguientes afirmaciones son verdaderas? Demostrar las respuestas usando la definición de $O(f(n))$.
a) $\log n \in O(n)$.
b) $n \in O(\log n)$.
14. Demostrar que si $f(n) \in O(n)$ entonces $(f(n))^2 \in O(n^2)$. ¿Se puede generalizar el resultado? En caso afirmativo, enunciarlo.

15. Probar que la notación O es transitiva, es decir, que si $f(n) \in O(g(n))$ y $g(n) \in O(h(n))$, entonces $f(n) \in O(h(n))$.

16. Probar que la notación Ω es transitiva, es decir, que si $f(n) \in \Omega(g(n))$ y $g(n) \in \Omega(h(n))$, entonces $f(n) \in \Omega(h(n))$ de dos formas:

a) Directamente según la definición de Ω .

b) Usando la regla de la dualidad y el problema 15.

17. Demostrar que $O(f(n)) \subset O(g(n))$ si y sólo si $f(n) \in O(g(n))$ pero $f(n) \notin \Omega(g(n))$ donde el símbolo \subset denota la inclusión estricta, es decir, que $A \subset B$ significa que hay elementos en el conjunto en B que no están en A

18. Si ε es una constante real positiva tal que $0 < \varepsilon < 1$, establecer una relación entre los órdenes correspondientes a las siguientes funciones por medio de la inclusión estricta ($O(f(n)) \subset O(g(n))$) y la igualdad cuando esta se dé ($O(f(n)) = O(g(n))$):

$$n \cdot \log n \quad n^8 \quad n^{1+\varepsilon} \quad (1+\varepsilon)^n \quad n^2 / \log n \quad (n^2 - n + 1)^4$$

19. Repetir el problema anterior con las siguientes funciones:

$$n! \quad (n+1)! \quad 2^n \quad 2^{n+1} \quad 2^{2n} \quad n^n \quad n^{\sqrt{n}} \quad n^{\log n}$$

Obtener la fórmula explícita y calcular el orden cuando sea conveniente de las siguientes recurrencias:

20. $t(1) = 1, t(n) = t(n-1) + n/2$

21. $t(1) = 1, t(2) = 4, t(n) = 8t(n-1) - 15t(n-2)$

22. $t(1) = c, t(n) = t(n/2) + b$

23. $t(0) = 0, t(1) = 1, t(n) = 5t(n-1) - 6t(n-2)$

24. $t(n) = 9n^2 - 15n + 106$ si $n = 0, 1, 2$, y $t(n) = t(n-1) + 2t(n-2) - 2t(n-3)$

25. $t(0) = 0, t(1) = 1, t(n) = 2t(n-1) - 2t(n-2)$

26. $t(n) = n$ si $n = 0, 1, 2, 3$ y $t(n) = t(n-1) - t(n-3) - t(n-4)$

27. $t(n) = n + 1$ si $n = 0, 1$, y $t(n) = 3t(n-1) + 2t(n-2) + 3 \cdot 2^{n-2}$

28. $t(n) = a$ si $n = 0, 1$, y $t(n) = t(n-1) + t(n-2) + c$

29. $t(n) = a$ si $n = 0, 1$, y $t(n) = t(n-1) + t(n-2) + cn$

30. $t(1) = 1, t(n) = 4t(n/2) + n$ si $n = 2^i$