

Centro de Investigación y de Estudios Avanzados del IPN
Departamento de Ingeniería Eléctrica
Sección de Computación

Programa de Maestría en Ingeniería Eléctrica
con Opción en Ciencias de la Computación

Análisis y Diseño de Algoritmos
Enero-Abril 2005

Lista de Ejercicios No. 3

1. Analizando el programa recursivo para resolver el problema de las torres de Hanoi, calcule el número de movimientos de discos que realiza el programa.
2. Muestre las acciones del algoritmo divide y vencerás para la multiplicación de los siguientes enteros: 1011 y 1101.

3. Demuestre que la recurrencia

$$T(1) = c$$
$$T(n) = 2T(n-1) + d$$

es un $O(2^n)$.

4. Demuestre que el valor

$$\binom{i+1}{i}$$

puede ser acotado por $\Omega(2^n / \sqrt{n})$.

5. Mediante la siguiente relación es posible calcular el coeficiente binomial

$$\binom{n}{k} = \begin{cases} 1 & \text{si } k=0 \text{ o } k=n \\ \binom{n-1}{k-1} + \binom{n-1}{k} & \text{si } 0 < k < n \\ 0 & \text{en otro caso} \end{cases}$$

- a) Escriba una función recursiva para el cálculo del coeficiente binomial
- b) Examine el árbol de llamados de la función recursiva
- c) Determine una mejor manera de calcular el coeficiente mediante programación dinámica
Sugerencia: El algoritmo construye una tabla que es conocida como el triángulo de Pascal.
- d) Calcule el tiempo de ejecución del algoritmo de programación dinámica

6. El siguiente problema se conoce como la subsecuencia común más larga. Dadas dos secuencias de símbolos X y Y , se dice que Z es una subsecuencia común de A y B , si Z representa tanto una secuencia de A como de B . Por ejemplo, si $X = ABCBDAB$ y $Y = BDCABA$, la secuencia $Z = BCA$ es una secuencia común tanto de A como de B . Z no es la secuencia más larga, ya que $BCBA$ también es una secuencia común de A y de B . Mediante programación dinámica, escriba un algoritmo para calcular la secuencia común más larga de dos cadenas de símbolos.
7. Escriba un procedimiento para encontrar la asociación de matrices óptima (no la cantidad de operaciones) del problema de parentización óptima de matrices. Analice su procedimiento e indique el tiempo de ejecución.
8. Escriba un algoritmo ávido para encontrar un recorrido (no necesariamente el mejor) para el problema del agente viajero.
9. Escriba un algoritmo ávido para el problema del corte máximo en gráficas dirigidas.
10. Escriba un algoritmo ávido para el problema de la cubierta mínima en gráficas no dirigidas.
11. Muestre que el algoritmo ávido para el problema Knapsack fraccional produce una solución óptima.
12. Considere el problema de dar cambio para n usando la menor cantidad de monedas. Suponga que las monedas disponibles están en denominaciones c^0, c^1, \dots, c^k , para algunos enteros $c > 1$ y $k \geq 1$. Muestre que el algoritmo ávido siempre produce una solución óptima.
13. Desarrolle un algoritmo branch and bound para el problema del agente viajero en donde se inicia siempre en el vértice 1 y, en cada nivel, se decide uno a cuál vértice ir (en lugar de cuál arista tomar). ¿Cuál es una cota inferior adecuada para esta estrategia? ¿Cómo se comporta el algoritmo para el ejemplo visto en clase?
14. Desarrolle un algoritmo branch and bound para el problema del corte máximo. ¿Cómo se puede estimar una cota superior al corte máximo?
15. Desarrolle un algoritmo branch and bound para el problema de la cubierta mínima. ¿Cómo se puede estimar una cota inferior a la cubierta mínima?
16. Considere una gráfica $G=(V,E)$ la cual tiene la propiedad de ser un árbol. Proporcione un algoritmo eficiente para encontrar una cubierta mínima de vértices.
17. Sea $G=(V,E)$ un árbol con pesos asociados a sus vértices tal que el peso de un vértice es igual al grado del vértice. Proporcione un algoritmo eficiente para encontrar una cubierta mínima pesada de G , esto es, la suma de los pesos de los vértices debe ser mínima.

18. Sea $G=(V,E)$ un árbol con pesos arbitrarios asociados a sus vértices. Proporcione un algoritmo eficiente para encontrar una cubierta mínima pesada de G .
19. Diseñe una máquina de Turing que decida al lenguaje $L = \{0^n 1^n\}$ en tiempo $O(n \log n)$.
Sugerencia: No intente marcar un 1 por cada 0, en lugar de eso, trate de verificar primero que la cadena esté formada por 0's y luego por una secuencia de 1's. Una vez que se haya asegurado de ello, determine la paridad de la cantidad de 0's y de la cantidad de 1's marcando algunos 0's y 1's. Si ambos son pares ó impares, siga verificando la paridad con los 0's y 1's no marcados. Si la paridad es diferente (par, impar), (impar, par) entonces rechace la palabra.
20. Demuestre que si B se reduce en tiempo polinomial a A , entonces, si $A \in P \Rightarrow B \in P$, donde P es la clase de problemas resueltos en tiempo polinomial.
21. Muestre que el problema de satisfactibilidad para fórmulas proposicionales es NP-completo. Sugerencia: construya un algoritmo que reduzca en tiempo polinomial el problema del CIRCUIT-SAT al problema de satisfactibilidad.
22. Muestre que el problema de hallar un circuito Hamiltoniano para una gráfica dirigida acíclica es NP-completo (tenga cuidado con este ejercicio).
23. Demuestre que la unión de dos lenguajes decidibles es decidible.
24. Muestre que los siguientes problemas acerca de programas en un lenguaje de programación real son no decidibles:
 - a) Si un programa dado puede ciclarse infinitamente sobre alguna entrada
 - b) Si un programa dado produce alguna salida
 - c) Si dos programas producen la misma salida sobre todas las entradas posibles
25. Demuestre que si P y coP son recursivos numerables, entonces, ambos son decidibles.
26. Probar que L_1 es reducible a L_2 y L_1 es insoluble, entonces, L_2 es insoluble (recursivo numerable pero no decidible).

Un problema final para reflexionar:

27. Dado que las computadoras son cada vez más veloces y el trabajo de programación relativo al costo de una computadora es cada vez más caro, ¿tiene sentido invertir tiempo en optimizar la programación para mejorar la eficiencia? ¿Puede indicar una lista de aplicaciones en donde la mejora en eficiencia es importante? Así también, ¿puede indicar una lista de aplicaciones en donde invertir en programas eficientes no tenga sentido?