

Guía de ejercicios C2

COM24502 Programación Avanzada

Pregunta 1

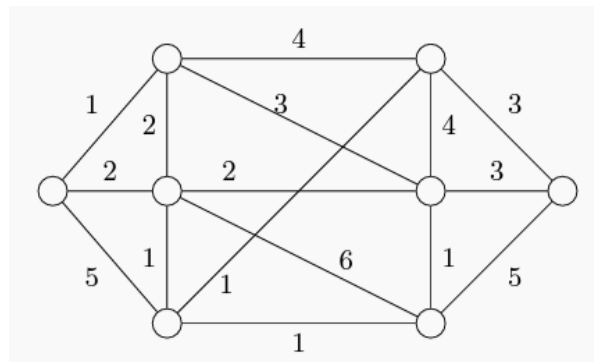
Defina los siguientes conceptos, proveyendo ejemplos en cada caso:

1. Teorema Maestro de recurrencias,
2. BackTracking,
3. Dividir para Reiniciar,
4. Algoritmo glotón,
5. Problema de ordenamiento.

Pregunta 2

Decida si cada afirmación siguiente es verdadera o falsa, argumentando cada respuesta.

1. El siguiente algoritmo de Dividir para Reiniciar tiene complejidad de tiempo $O(n)$: Dividimos la instancia en tres partes de tamaño $n/3$ que resolvemos de manera recursiva, y luego las mezclamos en tiempo $\Theta(n)$.
2. En el siguiente grafo, todo árbol cobertor de peso mínimo tiene peso total 10.



3. El problema de la mochila admite un algoritmo glotón de tiempo $O(n \log(n))$ que es óptimo.
4. Recurrencias de la forma $T(n) = aT(n/b) + \Theta(\log^t(n))$, donde $a > 0, b > 1$ y $t > 0$ constantes, no admiten solución finita.
5. Recurrencias de la forma $T(n) = aT(n/b) + \Theta(n^k)$, donde $a > 0, 0 \leq b < 1$ y $k > 0$ constantes, no admiten solución finita.

Pregunta 3

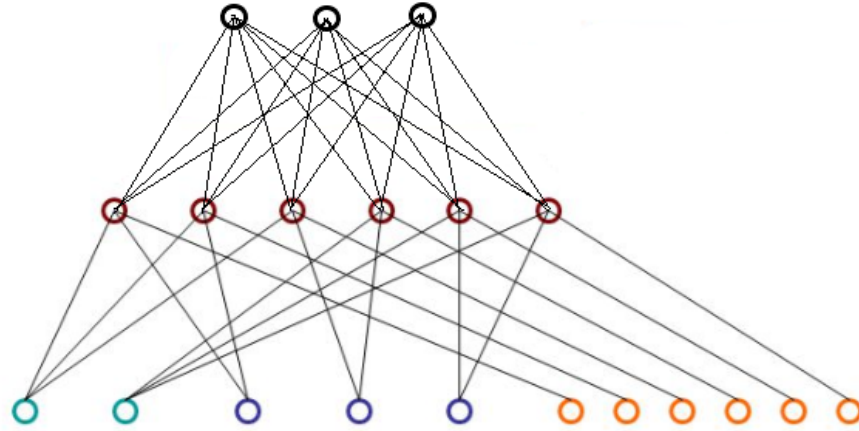
Analice las siguientes recurrencias, y decida si cumplen las hipótesis del Teorema Maestro. En caso de hacerlo, proponga un algoritmo (de juguete, no necesariamente para un problema concreto) cuya complejidad esté descrita por la recurrencia en cuestión y resuélvala.

1. $T(n) = 3T(n/2) + \Theta(1)$
2. $T(n) = T(n/2) + T(n/4) + \Theta(n^2)$
3. $T(n) = 4T(n/4) + \Theta(n)$
4. $T(n) = 2T(n/2) + \Theta(n \log(n))$
5. $T(n) = T(n/2) + \Theta(n)$

Pregunta 4

En el problema de *Cubrimiento por Vértices*, recibimos como entrada un grafo no-dirigido (V, E) y el objetivo es calcular un conjunto de nodos $V' \subseteq V$ de tamaño mínimo tal que, para toda arista $e = \{v_1, v_2\} \in E$, $v_1 \in V'$ ó $v_2 \in V'$.

1. Considere el siguiente algoritmo glotón: Partiendo con $V' = \emptyset$, consideramos iterativamente una arista arbitraria, y si ninguno de sus extremos está en V' , agregamos alguno de ellos a V' . El algoritmo termina cuando todas las aristas estén cubiertas. Demuestre que este algoritmo no es óptimo, y más aún puede retornar soluciones muy malas respecto al valor óptimo.
2. Considere el siguiente algoritmo glotón: Partiendo con $V' = \emptyset$, iterativamente elegimos el nodo de mayor grado, lo agregamos a V' y removemos las aristas que están cubiertas por el nodo agregado; el algoritmo termina cuando todos los nodos tienen grado cero. Ejecútelo en el siguiente grafo y luego argumente que la solución retornada podría tener más que el doble de aristas de la solución óptima.



3. Considere el siguiente algoritmo iterativo: Partiendo con $V' = \emptyset$, iterativamente escogemos una arista tal que sus dos extremos no están en V' , y los agregamos ambos a V' ; el algoritmo termina cuando toda arista tiene algún extremo en V' . Demuestre que hay instancias donde este algoritmo retorna el doble de aristas que el óptimo. Es posible demostrar que nunca retorna más del doble.

Pregunta 5

Considere el siguiente problema: Recibimos como entrada un arreglo de números ordenados de menor a mayor, posiblemente con repeticiones. El objetivo es, dado un número x , retornar el número de veces que x aparece en el arreglo.

Diseñe un algoritmo de tipo Dividir para Reinar que resuelva este problema en tiempo $O(\log(n))$