

Ejercicio resuelto en clase en semana 6

Programación 3 - 2021
16 de setiembre

1. Relaciones de recurrencia

Determine el orden de crecimiento de $T(n)$ en las siguientes relaciones de recurrencia. Asuma $T(1) = 1$ y que c es una constante positiva.

a. $T(n) = T(1) + T(n - 1) + cn$

b. $T(n) = 8T(n/2) + cn^2$

c. $T(n) = aT(n/b) + cn^E$, donde $E = \log_b a$, $b \geq 2$, $a \geq 1$

2. Mediana de la unión de secuencias ordenadas

Sean A y B dos secuencias de largo n de enteros. Asuma que las secuencias están ordenadas, n es potencia de 2, no hay elementos repetidos y son dadas en arreglos. Diseñe un algoritmo que permite encontrar la *mediana* (el valor que si se ordenara toda la secuencia quedaría en la mitad) de $A \cup B$. El algoritmo debe admitir una implementación con tiempo de ejecución $O(\log n)$.

3. Conjunto de nodos independientes de peso máximo de un árbol

Sea T un árbol con raíz binario perfecto (todas las hojas están a la misma profundidad y todos los niveles están completos). Cada nodo tiene un peso positivo, w . Un conjunto de nodos de T es *independiente* si dado cualquier par de ellos ninguno es padre del otro. El peso de un conjunto de nodos es la suma de los pesos de los nodos. Diseñe un algoritmo que calcula el peso máximo que puede tener un conjunto de nodos independientes de T . El algoritmo debe admitir una implementación con tiempo de ejecución $O(n)$, donde n es la cantidad de nodos de T .