Universidade Estadual de Maringá – Departamento de Informática
1218 – Análise de Algoritmos – Bacharelado em Informática
Prof. Marco Aurélio

Atividade 11

1. Mostre usando o método de substituição que a solução para $T(n) = T(\lceil n/2 \rceil) + 1$ é $O(\lg n)$.
2. Mostre usando o método de substituição que a solução para $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$ é $O(n \lg n)$.
3. Trace a árvore de recursão e forneça um limite assitótico restrito para $T(n) = T(n/5) + T(4n/5) + \Theta(n)$.
4. Resolva as seguintes recorrências:
   (a) $T(n) = 7T(n/2) + n^2$
   (b) $T(n) = T(9n/10) + n$
   (c) $T(n) = 3T(n/2) + n^2$
5. Proponha uma variação do mergesort que ao ínves de dividir o problema em dois subproblemas, divida em três subproblemas. Escreva a equação de recorrência para o seu algoritmo e encontre um limite superior.

Teorema mestre

Sejam $a \geq 1$ e $b > 1$ constantes, seja $f(n)$ uma função e seja $T(n)$ definida sobre os inteiros não negativos pela recorrência

$$T(n) = aT(n/b) + f(n)$$

onde interpretamos n/b com significado de $\lceil n/b \rceil$ ou $\lfloor n/b \rfloor$. Então, $T(n)$ pode ser limitado assintoticamente como a seguir:

1. Se $f(n) = O(n^{\log_b a - \epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$.
2. Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para alguma constante $\epsilon > 0$, e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e para todo n suficientemente grande, então $T(n) = \Theta(f(n))$.