MO417 – Complexidade de Algoritmos
Segundo Semestre de 2008
Terceira Lista de Exercícios

Método da Substituição

1. Mostre que a solução da recorrência $T(n) = T(\lceil n/2 \rceil) + 1$ pertence a $O(\lg n)$.
2. Mostre que a solução de $T(n) = 2T(\lceil n/2 \rceil) + n$ é $\Theta(n \lg n)$.
3. Mostre que a solução de $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n$ é $\Omega(n \lg n)$.
4. Resolva a recorrência $T(n) = T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + 1$.

Método da Iteração e Árvore de Recorrência

1. Determine um bom limite superior assintótico para a recorrência $T(n) = 3T(\lfloor n/2 \rfloor) + n$ usando o método da iteração.
2. Argumente que a solução da recorrência $T(n) = T(n/3) + T(2n/3) + n$ é $\Omega(n \lg n)$ usando o método da árvore de recorrência. Não se preocupe com arredondamentos.
3. Desenhe a árvore de recorrência para $T(n) = 4T(\lfloor n/2 \rfloor) + n$ e obtenha a classe Θ a qual a solução pertence.
4. Use o método da iteração para resolver a recorrência $T(n) = T(n - a) + T(a) + n$ onde $a \geq 1$ é um inteiro.
5. Use o método da árvore de recorrência para resolver a recorrência $T(n) = T(\alpha n) + T((1 - \alpha)n) + n$ onde $0 < \alpha < 1$ é uma constante.

Teorema Master

1. Use o Teorema Master para resolver as recorrências abaixo.
   (a) $T(n) = 4T(n/2) + n$
   (b) $T(n) = 4T(n/2) + n^2$
   (c) $T(n) = 4T(n/2) + n^3$
2. O tempo de execução de um algoritmo A é descrito pela recorrência $T(n) = 7T(n/2) + n^2$. Outro algoritmo A' tem complexidade de tempo descrita por $T'(n) = aT'(n/4) + n^2$. Qual é o maior inteiro a tal A' é assintoticamente mais rápido que A?
3. O Teorema Master pode ser aplicado à recorrência $T(n) = 4T(n/2) + n^2 \lg n$? Justifique sua resposta. Obtenha um bom limite superior assintótico para esta recorrência, sem usar o Teorema Master diretamente.