MAC 5711 - Análise de Algoritmos

Departamento de Ciência da Computação

Segundo semestre de 2002

Lista 3

Devolução dos exercícios 3, 6, 7, 8 e 9: 9 de setembro de 2002

1. Mostre uma implementação do procedimento **Intercala** do MergeSort que execute em tempo $\Theta(n)$. Argumente que de fato a sua implementação funciona em tempo $\Theta(n)$.

2. Considere uma função $f(n)$ que satisfaz a seguinte recorrência:

$$f(n) = \begin{cases} c_1 & \text{se } n \leq n_0 \\ 2f(\lfloor \frac{n}{2} \rfloor) + c_2\, n & \text{se } n > n_0 \end{cases}$$

   Prove que $f(n) = \Theta(n \log n)$.

3. Considere uma função $f(n)$ que satisfaz a seguinte recorrência:

$$f(n) = \begin{cases} 1 & \text{se } n = 1 \\ 2f(\lfloor \frac{n}{2} \rfloor) + n \log n & \text{se } n > 1 \end{cases}$$

   Resolva a recorrência acima, determinando qual é o comportamento assintótico de $f(n)$ (ou seja, conclua que $f(n) = \Theta(g(n))$, para alguma função $g(n)$ o mais simples possível). Ao resolver a recorrência, pode considerar apenas o caso em que $n = 2^m$.

4. Mostre que
   a) $\lfloor \log n \rfloor = \lceil \log(n+1) - 1 \rceil$
   b) $\log(n!) = \Omega(n \log n)$ (**Dica:** Use a fórmula de Stirling.)
   c) Usando o mesmo raciocínio, é possível mostrar que $n! = \Omega(2^{n \log n})$?
   d) Se $f(n) = \Omega(g(n))$, é verdade que $2^{f(n)} = \Omega(2^{g(n)})$? É verdade que $\log f(n) = \Omega(\log g(n))$?

5. Resolva completamente (método de substituição e prova por indução) a seguinte recorrência:

$$f(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ f(\lfloor \frac{2}{3} n \rfloor) + f(\lceil \frac{n}{3} \rceil) + \Theta(n) & \text{se } n > 1 \end{cases}$$

6. Prove (por indução em h) que toda árvore binária de altura h tem no máximo 2^h folhas.

7. Considere um vetor V com n elementos distintos. Prove que o seguinte processo gera uma permutação de $V[1], \ldots, V[n]$ com distribuição de probabilidade uniforme (ou seja, cada permutação é gerada com probabilidade $1/n!$). (**Dica:** Use indução em n.)

```
enquanto n > 1 faça
  r := inteiro aleatório entre 1 e n (ditribuição uniforme)
  troque V[r] com V[n]
  n := n-1
```

8. Escreva uma função que receba um vetor $V[1..n]$ de inteiros e rearranje o vetor de modo que, para algum j em $1..n$, tenhamos

$$V[i] \leq 0, \text{ para todo } i \leq j \quad \text{e} \quad V[i] \geq 0, \text{ para todo } i > j.$$

   Não use vetor auxiliar. A sua função deve executar em tempo $\Theta(n)$. Explique o funcionamento da sua função (sem uma explicação razoável, escrita em bom português, sua resposta nem será considerada, portanto capriche) e argumente que ela de fato executa em tempo $\Theta(n)$.

9. Você tem que escrever um programa que tem como entrada dois inteiros n e m e duas listas de números distintos. A primeira lista tem n números distintos e a segunda tem m números. A cada leitura de um número da segunda lista, seu programa deve imprimir se este pertence ou não a primeira lista.

   Você pode escolher entre as seguintes opções:
   a) fazer uma busca linear a cada leitura.
   b) ordenar a primeira lista e fazer busca binária a cada leitura;
   Supondo que $m = \lfloor \log \log n \rfloor$, qual dos métodos é mais eficiente? E qual é mais eficiente se $m = \lfloor \sqrt{n} \rfloor$? Justifique suas respostas.