Universidade Federal da Paraíba
Centro de Informática
Disciplina: Análise e Projeto de Algoritmos
Professor: Bruno Bruck

Lista de exercícios 1

1) Considere dois algoritmos A_1 e A_2 que resolvem o mesmo problema e cujas funções de tempo execução são, respectivamente, $T_{A_1}(n) = n^2 - n + 1$ e $T_{A_2}(n) = 6n \log_2 n + 2n$. Sendo assim, sabemos que $T_{A1}(n) = \Theta(n^2)$ e $T_{A2}(n) = \Theta(n \log_2 n)$.

a) Podemos afirmar que o algoritmo A_2 é sempre mais eficiente que o algoritmo A_1 dado que A_2 possui complexidade assintótica melhor? Justifique sua resposta.

b) A_1 é mais eficiente que A_2 para valores de n suficientemente pequenos? Justifique sua resposta.

c) Suponha que exista um terceiro algoritmo A_3 cujo tempo de execução seja dado pela função $T_{A_3}(n) = 10^3 n$, ou seja, $T_{A3}(n) = \Theta(n)$. Visto que A_3 possui complexidade assintótica linear, é correto afirmar que A_3 seria sempre a melhor escolha em termos de eficiência? Justifique sua resposta.

2) Considere o seguinte pseudocódigo do algoritmo de ordenação *Bubble Sort*:

Algoritmo 1: Bubble Sort

```
Função BubbleSort(entrada: Lista) : Lista
    repita
        houveTroca = Falso;
        para i = 0 até Tamanho(entrada) - 1 faça
            se entrada[i] > entrada[i+1] então
                Troca(entrada[i], entrada[i+1]);
                houveTroca = Verdadeiro;
            fim
        fim
    até houveTroca;
fim
```

Responda às seguintes perguntas considerando como operações básicas expressões de avaliação, atribuição de valores a variáveis, indexação de arrays e retorno de funções.

a) Identifique em que situação ocorre o pior caso do algoritmo e determine sua função $T(n)$ por meio da contagem de operações básicas.

b) Faça o mesmo que no item (a), mas considerando agora o melhor caso do algoritmo.

c) Calcule o limite assintótico superior mais apropriado para o algoritmo no pior caso e prove que tal limite é correto.

d) Determine um limite assintoticamente restrito para o algoritmo no melhor caso e prove que tal limite é correto.

e) Podemos dizer que esse algoritmo é $\Omega(1)$ para qualquer tipo de entrada? Justifique sua resposta.

3) Considere as seguintes funções:

- $f(n) = n^3$
- $g(n) = n \log n$
- $h(n) = n^2$
- $p(n) = n!$
- $k(n) = 3n^2 + 2n$

Determine, justificando sua resposta, quais das seguintes afirmações estão corretas. No caso de afirmações incorretas identifique o porquê.

a) $f(n) = \Omega(g(n))$ e $g(n) = O(f(n))$.

b) $g(n) = \Omega(f(n))$ e $h(n) = \Omega(g(n))$.

c) $g(n) = O(p(n))$ e $f(n) = O(k(n))$.

d) $k(n) = \Omega(f(n))$ e $p(n) = O(g(n))$.

e) $h(n) = O(f(n))$ e $k(n) = O(g(n))$.

4) Discorra sobre a veracidade das seguintes afirmações:

a) Um algoritmo A_1 tem seu pior caso em $O(n^2)$ e melhor caso em $\Omega(n)$. Sendo assim, podemos afirmar que tal algoritmo está em $\Theta(n^2)$.

b) O melhor caso de um algoritmo A_2 possui $T(n) = 2n^2 + n$. Sendo assim, é possível concluir que $T(n) = O(n^3)$.

c) Se um algoritmo tem seu pior caso em $O(n^2)$, então podemos concluir que tal algoritmo está em $O(n^2)$ para qualquer tipo de entrada.

d) Sejam $f(n)$ e $g(n)$ duas funções quaisquer. Então $f(n) = O(g(n))$ se, e somente se, existem duas constantes c e n_0 tais que $f(n) \leq cg(n)$ para algum valor de $n \geq n_0$.

5) Resolva a seguinte recorrência utilizando o Método Iterativo.

$$\begin{cases} T(0) = 1 \\ T(n) = T(n-1) + n, \text{ se } n > 0 \end{cases}$$

6) Para a fórmula fechada encontrada no exercício (5), determine:

a) Um limite assintótico superior (notação O);

b) Um limite assintótico inferior (notação Ω);

c) O limite assintótico restrito (notação Θ);

d) Um limite assintótico superior diferente do limite restrito encontrado;

e) Um limite assintótico inferior diferente do limite restrito encontrado.

7) Com relação aos algoritmos de ordenação CountingSort e RadixSort, responda às seguintes questões:

a) Identifique o motivo pelo qual o CountingSort é considerado um algoritmo pseudolinear e justifique sua resposta provendo detalhes sobre como isso pode afetar sua eficiência;

b) O que significa um algoritmo de ordenação ser estável? O CountingSort é um algoritmo estável? Justifique sua resposta.

c) É possível que o CountingSort seja mais eficiente que uma versão do RadixSort que utilize o CountingSort para a implementação da função OrdenaPorDigito? Justifique sua resposta;

8) Vimos que o algoritmo MergeSort possui complexidade $\Theta(n\ lg\ n)$ em qualquer cenário, enquanto que o algoritmo QuickSort possui complexidade $\Theta(n\ lg\ n)$ no melhor caso e no caso médio, mas no pior caso é $\Theta(n^2)$. Sendo assim, disserte sobre a seguinte afirmação: Na prática, o QuickSort tende a ser uma melhor escolha para a maioria dos casos em relação ao MergeSort.

9) Para cada uma das seguintes recorrências, identifique qual caso do Teorema Mestre se aplica justificando sua resposta. Caso seja possível utilizar o Teorema Mestre, determine um limite *assintoticamente restrito*.

a) $T(n) = 2T(n/3) + n$

b) $T(n) = 2T(n/10) + n$

c) $T(n) = 2T(n/2) + nlogn$

d) $T(n) = 4T(n/2) + 1$

e) $T(n) = 3T(n-1) + n$

f) $T(n) = 2T(n/2) + n^2$