Lista de exercícios 1

Projeto e análise de algoritmos

1. Suponha um algoritmo A e um algoritmo B com funções de complexidade de tempo $a(n) = n^2 - n + 549$ e $b(n) = 49n + 49$, respectivamente. Determine quais são os valores de n pertencentes ao conjunto dos números naturais para os quais A leva menos tempo para executar do que B.
2. Suponha que n mede o tamanho das instâncias de um certo problema. A documentação de um algoritmo para o problema diz que o algoritmo consome $10^3n + 10^6$ unidades de tempo no melhor caso e $2n^2/10$ unidades de tempo no pior caso. Isso faz sentido?
3. Suponha que um algoritmo para um certo problema consome $O(n^2)$ unidades de tempo, sendo n o tamanho de uma instância. Alguém diz ter um outro algoritmo para o mesmo problema que consome apenas $\Omega(n)$ unidades de tempo. Devo ficar impressionado?
4. O que significa dizer que um algoritmo executa em tempo proporcional a n?
5. Sejam duas funções não negativas f (n) e g(n). Mostre que $\max(f(n), g(n)) = \Theta(f(n)+g(n))$.
6. Explique por que a seguinte afirmação não faz sentido: o algoritmo *A* consome pelo menos $O(n^2)$ unidades de tempo.
7. O algoritmo abaixo recebe um vetor $P[1 \,..\, n]$ com $n \geq 0$ e devolve um vetor $X[0 \,..\, n]$. Ao receber argumentos P e i, a função Teste devolve 1 ou 0 e consome O(i) unidades de tempo para dar a resposta. Calcule detalhadamente o consumo de tempo do Algoritmo no pior e no melhor casos.

```
Algoritmo (P, n)
para k crescendo de 0 até n
   i := k
   enquanto i ≥ 1 e Teste (P, i) = 0
      i := i − 1
   X[k] := i
devolva X
```

8. Determine a complexidade assintótica dos laços apresentados a seguir, no pior caso (notação O).

a)

```
c ← 1
   para (i = 0; i ∗ i < n; i + +) faça
      imprime(c)
   c ← c + 1
```

b)

```
c ← 1
para (i = 1; i ≤ n; i + +) faça
   para (j = 1; j ≤ n; j = j + i) faça
      imprime(c)
      c ← c + 1
```

9. Escreva um algoritmo que calcule a soma dos primeiros n números inteiros $1 + 2 + 3 + \cdots + n$. Encontre o invariante de laço e prove que ele está correto.
10. Escreva um algoritmo de busca linear para encontrar um elemento em uma lista. Prove, por invariante de laço, que o algoritmo está correto.
11. Escreva um algoritmo que calcula o fatorial de um número. Prove, por invariante de laço, que o algoritmo está correto.
12. São dados 2n números distintos distribuídos em dois arranjos com n elementos A e B ordenados de maneira tal que A[1] > A[2] > A[3] > ... > A[n] e B[1] > B[2] > B[3] > ... > B[n]. O problema é achar o n-ésimo maior número dentre estes 2n elementos.
    a. Obtenha um limite inferior para o número de comparações necessárias para resolver este problema.
    b. Apresente um algoritmo cuja complexidade no pior caso seja igual ao valor obtido na letra a), ou seja, um algoritmo ótimo.
13. Considere o problema de inserir um novo elemento em um conjunto ordenado A[1] > A[2] > A[3] > ... > A[n].
    a. Apresente um limite inferior para essa classe de problemas.
    b. Apresente uma prova informal para o limite inferior.
    c. Apresente um algoritmo para resolver o problema desta questão. O seu algoritmo é ótimo?
14. Dada uma lista ordenada de n elementos de valor inteiro, o problema de unificação de lista consiste em realizar seguidamente a operação de remover os dois elementos de menor valor da lista e inserir um novo elemento com valor igual à soma dos dois primeiros. A cada operação a lista passa a ter um elemento a menos. A unificação termina quando restar somente um elemento na lista.
    a. Apresente um algoritmo que realiza a unificação da lista em tempo O(n).
    b. É possível realizar a unificação da lista em tempo sublinear? Justifique a sua resposta.
    c. Qual o limite inferior para o problema da unificação?
15. Exiba três pares (c, n_0) tais que $100n^2 \leq c\frac{1}{100}n^3$ para todo $n \geq n_0$.
16. Seja f a função definida por $f(n) = 3n^2 + 7n - 8$ para todo inteiro não-negativo n. Mostre que $f(n) = O(n^2)$.
17. Mostre que $100n = O(2n)$, que $n + 100 = O(n)$, e que $100n = O(2n^2 + n)$.
18. Prove que $n^2 + 999n + 9999 = O(n^2)$.
19. É verdade que $\lg n^{10} = O(\lg n)$?
20. Escreva uma versão recursiva do algoritmo de ordenação por seleção. O algoritmo deve rearranjar em ordem crescente qualquer vetor dado A[p . . r].
21. Desenvolva um algoritmo iterativo de complexidade de tempo $\Theta(n)$ que recebe um vetor v com n inteiros e, utilizando espaço adicional constante, coloca os números pares no começo (lado esquerdo) do vetor e os números ímpares no fim (lado direito), devolvendo o índice do primeiro número ímpar, ou n + 1 se todos os números do vetor forem pares. Aplique análise assintótica para mostrar que a complexidade da sua solução é de fato linear.
22. Escreva o pseudocódigo para um algoritmo de divisão e conquista SOMA-MATRIZES-RECURSIVO que some duas matrizes $n \times n$ A e B particionando cada uma delas em quatro submatrizes $n/2 \times n/2$ e, então, somando recursivamente os pares

correspondentes de submatrizes. Suponha que o particionamento de matrizes use cálculos de índice com tempo $\Theta(1)$. Escreva uma recorrência para o tempo de execução do pior caso de SOMA-MATRIZES-RECURSIVO e resolva sua recorrência. O que acontece se você usar a cópia com tempo $\Theta(n^2)$ para implementar o particionamento em vez de cálculos de índice? (Consulte o livro do Cormen 4ª edição seção 4.1 para mais informações).

23. Resolva as recorrências abaixo usando o método iterativo (assuma que T(1) = 1).
    a. $T(n) = T(n-1) + 3$
    b. $T(n) = T\left(\frac{n}{2}\right) + 3$
    c. $T(n) = T(n-1) + 2n$
    d. $T(n) = T(n-1) + 3n + 1$
24. Use o método da substituição para mostrar que cada uma das seguintes recorrências definida sobre valores reais tem a solução assintótica especificada.
    a. $T(n) = T(n - 1) + n$ tem solução $T(n) = O(n^2)$.
    b. $T(n) = T(n/2) + \Theta(1)$ tem solução $T(n) = O(\lg n)$.
    c. $T(n) = 2T(n/2) + n$ tem solução $T(n) = \Theta(n \lg n)$.
    d. $T(n) = 2T(n/2 + 17) + n$ tem solução $T(n) = O(n \lg n)$.
    e. $T(n) = 2T(n/3) + \Theta(n)$ tem solução $T(n) = \Theta(n)$.
    f. $T(n) = 4T(n/2) + \Theta(n)$ tem solução $T(n) = \Theta(n^2)$.
25. Para cada uma das recorrências a seguir, esboce sua árvore de recursão e escolha um bom limite assintótico superior em sua solução. Depois, use o método da substituição para verificar sua resposta.
    a. $T(n) = T(n/2) + n^3$.
    b. $T(n) = 4T(n/3) + n$.
    c. $T(n) = 4T(n/2) + n$.
    d. $T(n) = 3T(n - 1) + 1$.
26. Resolva as recorrências abaixo usando o método mestre.
    a. $T(n) = T\left(\frac{2n}{3}\right) + n^2$
    b. $T(n) = T\left(\frac{n}{2}\right) + 1$
    c. $T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$
    d. $T(n) = 3T\left(\frac{n}{3}\right) + n$
    e. $T(n) = 8T\left(\frac{n}{2}\right) + n^3$