Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Programa de Pós-Graduação em Ciência da Computação
Projeto e Análise de Algoritmos

Exercícios
Capítulos 1, 2 e 3 do Livro Texto

1. Considere duas funções $f(n) = 0,5n^2 - 3n$ e $g(n) = n^2$. Agora, usando a notação Big-O, mostre que $f(n)$ é $O(g(n))$.

2. Dado dois algoritmos A e B que resolvem o mesmo problema e possuem complexidade no tempo $8n^2$ e n^3, respectivamente, responda os itens abaixo:

a. Qual é o maior valor de entrada n para o qual o algoritmo B é mais eficiente que o algoritmo A?

b. Na sua opinião, qual algoritmo é mais eficiente?

3. Determine a complexidade no tempo do algoritmo abaixo que promete encontrar os elementos mínimo e máximo do vetor de entrada V de tamanho n e responda os itens a seguir. Todas as respostas devem ser devidamente justificadas.

```
MaxMin (V, n)
max = V[1]
min = V[1]
para i = 2 até n faça
  se V[i] > max então max = V[i]
  se V[i] < min então min = V[i]
```

a. Podemos dizer que o algoritmo $MaxMin$ para e é correto?

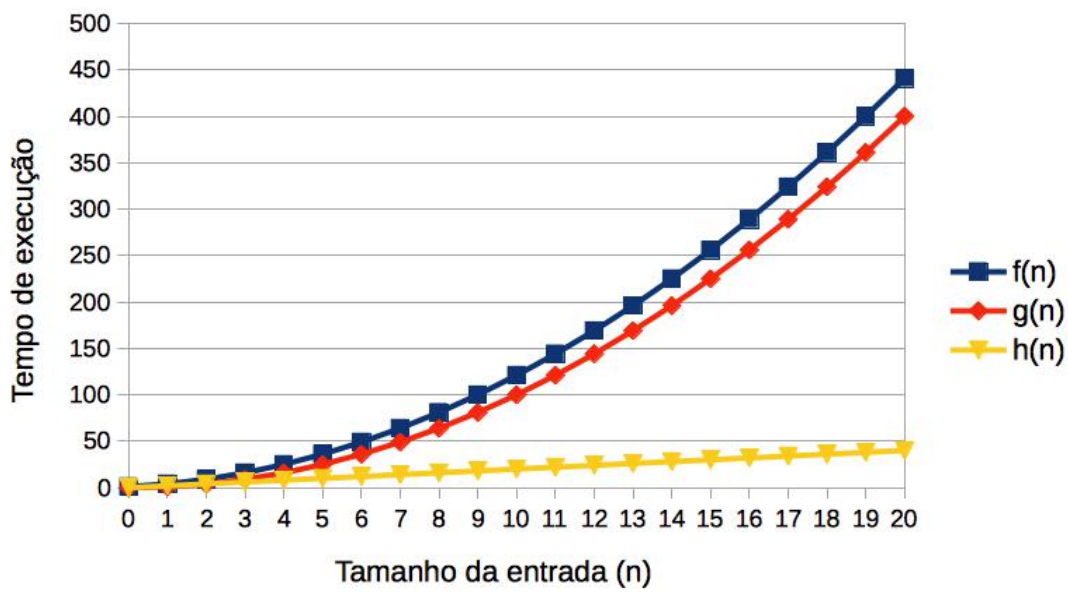
b. Existe melhor e pior caso?

c. O algoritmo $MaxMin$ é eficiente?

4. Usando notação assintótica, descreva a complexidade no tempo do algoritmo abaixo. Ele recebe dois vetores A e V de tamanhos n e w, respectivamente.

```
TESTE (A, V, n, w)
para i = 1 até n faça
  para j = 1 até w faça
    V[j] = A[i] + 2
max = V[1]
para j = 2 até w faça
  se max < V[j] então max = V[j]
```

5. Observe as funções representadas no gráfico abaixo.



Assinale a afirmativa correta sobre o crescimento assintótico dessas funções.

(a) $f(n) = O(h(n))$ e $f(n) = w(g(n))$.
(b) $g(n) = \Omega(f(n))$ e $f(n) = \Theta(h(n))$.
(c) $h(n) = w(g(n))$ e $g(n) = \Theta(f(n))$.
(d) $g(n) = O(f(n))$ e $g(n) = \Omega(h(n))$.
(e) $h(n) = \Omega(f(n))$ e $h(n) = O(g(n))$.

6. Mostre que $7x^2$ é $O(x^3)$. Também é verdade que x^3 é $O(7x^2)$? São equivalentes?

7. Mostre que x^2 é $\Omega(x)$. Também é verdade que x é $\Omega(x^2)$? São equivalentes?

8. As afirmações abaixo são verdadeiras ou falsas? Justifique.

a. $2n^2 + 1000$ é $\Omega(n^2)$.

b. $log(n^2)$ é $w(log(n))$.

c. 2^{n+1} é $O(2^n)$.

d. 2^{2n} é $O(2^n)$.

e. $log(n) + \sqrt{n}$ é $O(n)$.

9. Considere dois algoritmos $A1$ e $A2$, cujas funções de custo são, respectivamente, $T1(n) = n^2 - n + 1$ e $T2(n) = 6nlog_2n + 2n$. Para simplificar a análise, assuma que $n > 0$ é sempre uma potência de 2. Com relação ao enunciado, assinale a alternativa correta.

(a) $T1(n) = \Theta(n^2)$ e $T2(n) = \Theta(nlogn)$, então $A2$ é sempre mais eficiente que $A1$.
(b) O limite superior $T1(n) = O(n^3)$ é correto e assintoticamente restrito.
(c) O limite inferior $T2(n) = \Omega(n^3)$ é correto e assintoticamente restrito.
(d) $T1$ e $T2$ são assintoticamente equivalentes.
(e) $A1$ é mais eficiente que $A2$, para n suficientemente pequeno.

10. Sejam $T_A(n)$ e $T_B(n)$ os tempos de execução de pior caso de dois algoritmos A e B propostos para um mesmo problema computacional, em função de um certo parâmetro n. Dizemos que o algoritmo A é mais eficiente que o algoritmo B assintoticamente no pior caso quando

(a) $T_A(n) = o(T_B(n))$.
(b) $T_B(n) = o(T_A(n))$.
(c) $T_A(n) = O(T_B(n))$.
(d) $T_B(n) = O(T_A(n))$.
(e) $T_A(n) = \Theta(T_B(n))$.

11. Qual é o número mínimo de comparações necessário para encontrar o menor elemento de um conjunto qualquer não ordenado de n elementos?

(a) 1.
(b) $n - 1$.
(c) n.
(d) $n + 1$.
(e) $n\ log\ n$.

12. Um algoritmo tradicional e muito utilizado possui complexidade $n^{1,5}$, enquanto um novo algoritmo proposto é da ordem de $nlogn$:

$f(n) = n^{1,5}$
$g(n) = nlogn$

Qual algoritmo adotar? Por quê?

13. A função SORT abaixo ordena de forma crescente um vetor A de n elementos.

```
SORT (A, n)
para j = 1 até n - 1 faça
   menor = j
   para i = j + 1 até n faça
      se A[i] < A[menor] então menor = i
   aux = A[menor]
   A[menor] = A[j]
   A[j] = aux
```

Dado que $T(n)$ é o tempo de execução da função SORT para as entradas A e n, é possível afirmar que a ordem de $T(n)$ é

(a) $T(n) = O(1)$.
(b) $T(n) = O(log(n))$.
(c) $T(n) = O(n)$.
(d) $T(n) = o(n^2)$.
(e) $T(n) = O(n^2)$.

14. Considerando $f \equiv f(n)$, $g \equiv g(n)$ e k uma constante, determine se as sentenças abaixo são verdadeiras ou falsas. Caso sejam falsas reescreva corretamente.

a. $O(f+g) = O(f) + O(g)$.

b. $O(f.g) = O(f).O(g)$.

c. $O(k.g) = k.O(g) = O(g)$.

d. $5n + 8n^2 + 100n^3 = O(n^4)$.

e. $5n + 8n^2 + 100n^3 = O(n^2 log(n))$.

15. Assumindo que cada expressão representa o tempo de processamento de um algoritmo para resolver um problema de tamanho n. Selecione o termo dominante e especifique a complexidade no pior caso.

Expression	Dominant term(s)	$O(\ldots)$
$5 + 0.001n^3 + 0.025n$		
$500n + 100n^{1.5} + 50n \log_{10} n$		
$0.3n + 5n^{1.5} + 2.5 \cdot n^{1.75}$		
$n^2 \log_2 n + n(\log_2 n)^2$		
$n \log_3 n + n \log_2 n$		
$3 \log_8 n + \log_2 \log_2 \log_2 n$		
$100n + 0.01n^2$		
$0.01n + 100n^2$		
$2n + n^{0.5} + 0.5n^{1.25}$		
$0.01n \log_2 n + n(\log_2 n)^2$		
$100n \log_3 n + n^3 + 100n$		
$0.003 \log_4 n + \log_2 \log_2 n$		

16. Sejam f, g e h funções reais positivas da variável inteira n, assinale a alternativa **INCORRETA** de acordo com os conceitos de notações assintóticas.

(a) Por exemplo, se $f = n^2 - 1$, $g = n^2$ e $h = n^3$, então f é $O(g)$, f é $O(h)$, g é $O(f)$, mas h não é $O(f)$. Consequentemente, f é $\Theta(g)$, mas f não é $\Theta(h)$.

(b) A notação Θ exprime o fato de que duas funções possuem a mesma ordem de grandeza assintótica.

(c) Por exemplo, se $f = 5 + 2log(n) + nlog(n)$ e $g = n^2$, então f é $O(g)$, porém f não é $\Theta(g)$. No caso, f é $\Theta(n)$.

(d) Assim como a notação O é útil para descrever limites superiores assintóticos, a notação Ω é empregada para limites inferiores assintóticos.

(e) Por exemplo, se $f = n^2 - 1$, então são válidas as igualdades $f = \Omega(n^2)$, $f = \Omega(n)$ e $f = \Omega(1)$, mas não vale $f = \Omega(n^3)$.

17. Um algoritmo quadrático com tempo de processamento $T(n) = cn^2$ consome $T(N)$ segundos para processar N itens. Assumindo $N = 100$ e $T(N) = 1ms$, quanto tempo este algoritmo consome para processar uma entrada de 5.000 itens?

18. Os algoritmos A e B, para uma entrada de tamanho n, consomem exatamente $T_A = 5nlog(n)$ e $T_B = 25n$ microssegundos. Qual algoritmo apresenta a melhor complexidade no pior caso? Para quais tamanhos de entrada um é melhor que o outro?

19. Os algoritmos A e B, para uma entrada de tamanho n, consomem exatamente $T_A = 0,1n^2log(n)$ e $T_B = 2,5n^2$ microssegundos. Informe qual algoritmo apresenta a melhor complexidade no pior caso e encontre o valor k tal que para qualquer $n > k$ o algoritmo informado apresente melhor desempenho. Se o problema apresenta entradas $n \leq 10$, qual algoritmo você recomendaria para resolver o problema?

20. Os algoritmos A e B, para uma entrada de tamanho n, consomem exatamente $T_A = c_A nlog(n)$ e $T_B = c_B n^2$ microssegundos. Diga qual o melhor algoritmo para processar uma entrada de tamanho $n = 2^{20}$ se o algoritmo A consome 10 microssegundos para processar 1.024 itens e o algoritmo B consome 1 microssegundo para processar a mesma quantidade.

21. Quais das seguintes igualdades são verdadeiras?

I. $n^2 = O(n^3)$

II. $2n + 1 = O(n^2)$

III. $n^3 = O(n^2)$

IV. $3n + 5nlog(n) = O(n)$

(a) Somente I e II.
(b) Somente II, III e IV.
(c) Somente III e IV.
(d) Somente I, II e III.
(e) Somente I, III e IV.

22. Qual das seguintes afirmações sobre crescimento assintótico de funções **NÃO** é verdadeira?

(a) $2n^2 + 3n + 1 = O(n^2)$.
(b) Se $f(n) = O(g(n))$, então $g(n) = O(f(n))$.
(c) $log(n^2) = O(log(n))$.
(d) Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, então $f(n) = O(h(n))$.
(e) $2^{n+1} = O(2^n)$.

23. Sejam $T_1(n) = 100n + 15$, $T_2(n) = 10n^2 + 2n$ e $T_3(n) = n^3 + n^2 + 3$ as equações que descrevem a complexidade de tempo dos algoritmos $Alg1$, $Alg2$ e $Alg3$, respectivamente, para entradas de números inteiros de tamanho $n > 0$. Assinale a alternativa correta.

(a) As complexidades assintóticas de $Alg1$, $Alg2$ e $Alg3$ em notação Big-O estão, respectivamente, em $O(100)$, $O(10)$, $O(1)$.
(b) $Alg1$, $Alg2$ e $Alg3$ são assintoticamente equivalentes.
(c) $Alg1$ é sempre mais eficiente que $Alg2$.
(d) A complexidade assintótica de $Alg3$ está em $o(n^3)$.
(e) $Alg1$ é assintoticamente mais eficiente que $Alg3$.