

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA – IFET-CE  
CURSO DE SISTEMAS DE INFORMAÇÃO  
ALGORITMOS E ESTRUTURAS DE DADOS II  
PRIMEIRA LISTA DE EXERCÍCIOS

- 1) Explique como pode ser feita a análise do tempo de execução de um algoritmo.
- 2) A análise de algoritmos mede o tempo de execução de um algoritmo em função de que?
- 3) As classes de comportamento assintótico são usadas para análise do comportamento do tempo de execução de um algoritmo. Quais são as principais classes de comportamento assintótico existentes?
- 4) O que acontece em termos de tempo de execução quando aumentamos o tamanho da entrada de um algoritmo polinomial e de um algoritmo exponencial?
- 5) O que é um algoritmo polinomial, um algoritmo exponencial e um problema intratável?
- 6) É certo dizer que se aumentarmos em dez vezes a velocidade de uma máquina que executava um programa qualquer em determinado tempo, ela executará o programa com uma entrada 10 vezes maior no mesmo tempo que executava o programa na situação original? Por quê?
- 7) O que é pior caso, caso médio e melhor caso de um algoritmo?
- 8) Cite dois diferentes parâmetros que podem ser analisados através de técnicas de análise de algoritmos.
- 9) Qual é a diferença entre um método empírico de avaliação e a análise científica formal de um algoritmo?
- 10) Dado um problema particular, o que se deve fazer para obter um algoritmo de menor custo para resolvê-lo?
- 11) O que é uma solução ótima para um algoritmo?
- 12) Uma função de complexidade  $f(n)$  representa o tempo em segundos de execução de um algoritmo? Verdadeiro ou Falso? Justifique.
- 13) Porque analisamos o comportamento de um algoritmo quando a entrada do algoritmo tende ao infinito? Podemos analisar algoritmos através de entradas pequenas?
- 14) As classes de complexidade  $O(1)$  e  $O(2)$  são equivalentes?
- 15) Dado dois algoritmos: o primeiro requer  $n^5$  passos e o segundo requer  $2^n$  passos. a) Considerando uma entrada pequena ( $n < 23$ ), é melhor utilizar qual algoritmo? b) Considerando uma entrada que tende ao infinito (tal que  $n \geq 23$ ), é melhor utilizar qual algoritmo?

- 16) Apresente um algoritmo para obter o maior e o segundo maior elemento de um conjunto. Apresente também uma análise do algoritmo. Você acha o seu algoritmo eficiente? Por quê? Procure comprovar suas respostas.
- 17) Cite uma característica geral de algoritmos que pertencem a cada uma das seguintes classes de complexidade: a) Constante. b) Logarítmica. c) Linear. d) Logaritmo-Linear. e) Quadrática. f) Cúbica. g) Exponencial.
- 18) O gráfico ilustrado na Figura 1 mostra o comportamento do tempo de execução dos algoritmos A1, A2, A3, A4, A5 conforme variação no tamanho da entrada. Com isto, verifique se as seguintes sentenças são falsas ou verdadeiras. JUSTIFIQUE.

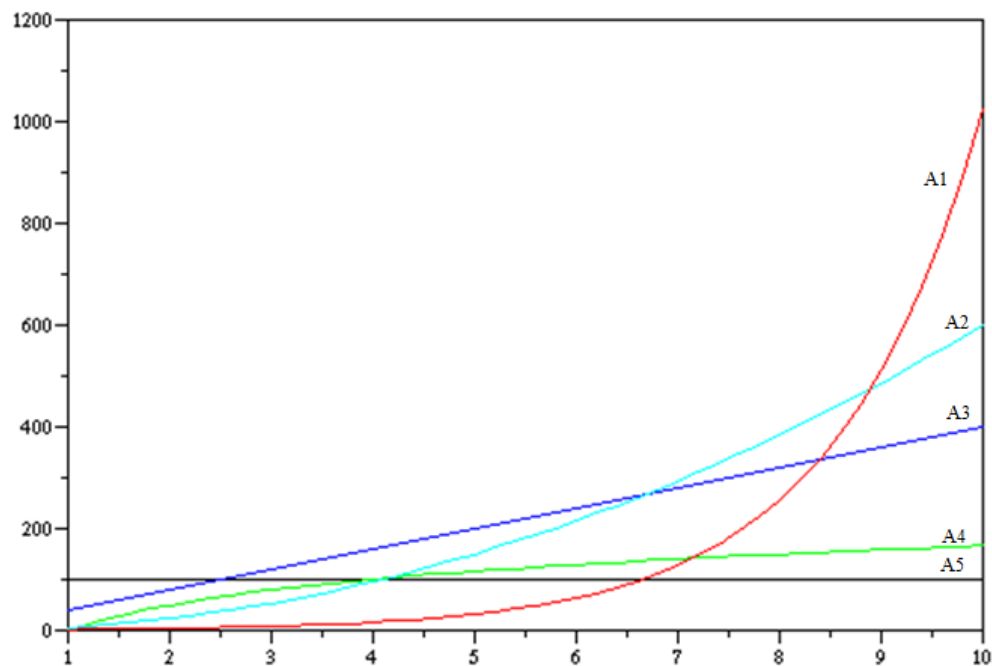


Figura 1: Comportamento de tempo de execução dos algoritmos A1, A2, A3, A4 e A5

- O algoritmo A5 tem complexidade  $O(1)$ .
  - O algoritmo A4 tem complexidade maior do que o algoritmo A2.
  - Supondo que um determinado problema pode ser resolvido pelo algoritmo A2 ou A3. Então, neste caso, a melhor escolha para resolver o problema é utilizar o algoritmo A2.
  - O algoritmo A3 tem complexidade  $O(n^2)$
  - Para entradas pequenas ( $n < 5$ ) o algoritmo A1 tem o melhor tempo de execução. Apesar disso, para tamanhos de entradas tendendo ao infinito é melhor utilizar os algoritmos A2, A3, A4 ou A5 em vez de A1.
- 19) Através de uma prova informal determine e justifique qual é a complexidade de pior caso do algoritmo de busca sequencial dado pelo fluxograma ilustrado na Figura 2.

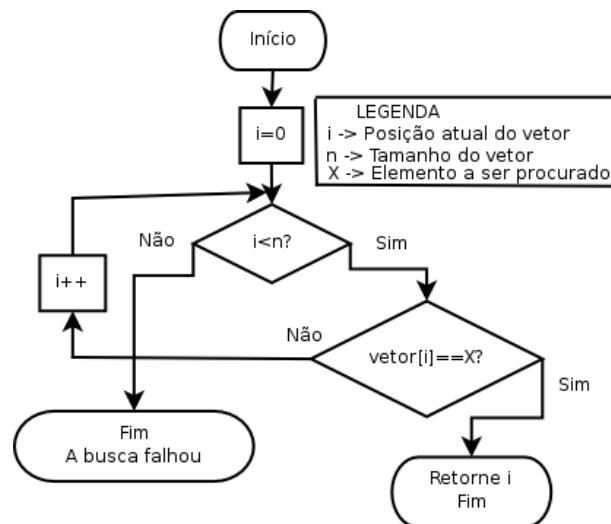


Figura 2: Algoritmo de busca sequencial

- 20) Através de uma prova informal determine e justifique qual é a complexidade de pior caso do algoritmo de busca binária dado pelo pseudocódigo ilustrado na Figura 2.

```

BUSCA-BINÁRIA (V[], início, fim, e)
  i recebe o índice do meio entre início e fim
  se (v[i] = e) então
    devolva o índice i  # elemento e encontrado
  fimse
  se (início = fim) então
    não encontrou o elemento procurado
  senão
    se (V[i] vem antes de e) então
      faça a BUSCA-BINÁRIA(V, i+1, fim, e)
    senão
      faça a BUSCA-BINÁRIA(V, início, i-1, e)
  fimse
  
```

Figura 3: Algoritmo de busca binária

- 21) Através de uma prova informal determine e justifique qual é a complexidade de pior caso das seguintes operações sob uma lista encadeada:
- Inserção de um novo nó no início da lista.
  - Inserção de um novo nó no fim da lista.
- 22) Um caixeiro viajante deseja visitar  $n$  cidades de tal forma que sua viagem inicie e termine em uma mesma cidade, e cada cidade deve ser visitada uma única vez. Supondo que sempre há uma estrada entre duas cidades quaisquer, descreva um algoritmo **exponencial** que encontre a menor rota para a viagem.
- 23) Determine a complexidade das seguintes funções:
- $f_1 = 34$ .
  - $f_2 = 3n + 5n^3 + 2n^2 + 2$
  - $f_3 = n \lg(n) + n + \lg(n)$
  - $f_4 = 3^n + \lg(n) + n^3$

24) Através de regras para análise de algoritmos determine a complexidade dos seguintes algoritmos ( $n$  determina o tamanho da entrada):

a)

```

tipo Matriz: Vetor [n,n] de Real;
var i, j, k: Inteiro;
var A,B,C: Matriz;
para (i ← 1; i ≤ n; i ← i+1) faça
    para (j ← 1; j ≤ n; j ← j + 1) faça
        início
            C[i,j] ← 0;
            para (k ← 1; k ≤ n; k ← k + 1) faça
                C[i,j] ← C[i,j] + A[i,k] * B[k,j];
        fim

```

b)

```

var i, j, p: Inteiro;
para (i ← 1; i ≤ n; i ← i+1) faça
    para (j ← 1; j ≤ n; j ← j + 1) faça
        p ← p + 1;

```

c)

```

var i, j, p: Inteiro;
para (i ← 1; i ≤ n; i ← i+1) faça
    para (j ← 1; j ≤ n*n*n; j ← j + 1) faça
        p ← p + 1;

```

d)

```

var i, j, p: Inteiro;
para (i ← 1; i ≤ n; i ← i+1) faça
    para (j ← (n-1); j ≤ n; j ← j + 1) faça
        p ← p + 1;

```

25) Calcule a complexidade de caso médio do algoritmo de busca sequencial (Figura 2).

Lembrete:  $\sum_{i=1..n} i = n * (n + 1) / 2$ .