Laboratório prático – Módulo 1

Exercícios de revisão

Parte 4: Análise de algoritmos – Módulo 1

1. Responda às questões seguintes, justificando convenientemente:

a) Para que é utilizada a notação assintótica O-grande (*Big-O*) e o que classifica?

b) Explique as diferenças entre a complexidade temporal de O(1), O(n), O(log n) e O(n^2)?

c) Explique qual é a ordem de complexidade, no pior dos casos, de um algoritmo de pesquisa linear, justificando.

d) Um algoritmo de pesquisa linear tem um "melhor dos casos"? Justifique.

e) Indique, justificando convenientemente, qual é a estratégia de desenho de algoritmos apresentada por um algoritmo de pesquisa binária.

2. **Escreva a fórmula** para o cálculo do trabalho de cada um dos seguintes algoritmos em pseudo-código **e calcule o limite superior** para esse trabalho. **Justifique** as suas respostas.

a)
```
Algoritmo Q(n):
    //Input: inteiro positivo n
    //Output: inteiro Q(n)

    Se n == 1: Devolver 0
    Se n == 2: Devolver 1
    Devolver 2 x Q(n−2) + n
```

b)
```
Algoritmo P(n):
    //Input: inteiro positivo n
    //Output: inteiro P(n)

    Se n == 1: Devolver 1
    Devolver P(n−1) x P(n−2) + 1
```

c)
```
Algoritmo R(a, i, k):
    //Input: sequência a; inteiros i,k > 0
    //Output: elemento de a

    Se k > 1 e i > 1:
        Se a[i] > a[k]:
               Devolver R(a, i, k-1)
        Devolver R(a, i+1, k)
```

d)
```
Algoritmo S(a, i, n):
    //Input: sequência a; n = len(a); i>0
    //Output: True/False

    Se n == m:
         Se n == 1: Devolver True
         Senão: Devolver False
    Senão:
         m = n/2
         Se a[n/2] == a[n]: Devolver R(a, m, n)
         Senão: Devolver R(a, i, m)
```

3. Considere o algoritmo seguinte:

```
Algoritmo Polinomio(a, x, n):
    //Input: a array; x, real; n, inteiro
    //Output: O valor do polinómio pn(x) =  an x^n + ... + a1 x +a0
   p ← a[0];
   px← 1;
   Para i a variar de 1 a n:
          px ← x * px;
          p ← p + a[i] * px
```

a) Prove a correção deste algoritmo, ou seja, dado o array de coeficientes reais, a[0 .. n] e , valor de x, que `Polinomio` devolve o valor de $p_n(x) = a_n x^n + \ldots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i$.

b) Calcule e indique, justificando, a ordem de complexidade deste algoritmo.

c) Explique, justificando, qual a estratégia geral que este algoritmo está a usar.

d) Consegue melhorar este algoritmo? Explique como e analise a ordem de complexidade nesse caso, justificando.

4. Considere o algoritmo seguinte que, caso encontre, numa dada string, um caracter A e, mais adiante, o caracter B, devolve a posição do primeiro A encontrado; senão, devolve uma indicação de padrão não encontrado:

```
Algoritmo padraoAB(s, A, B):
    //Input: s, string;
    //Output: posição do 1ºA se houver, depois, um B
    iA = 0
    iB = 0
    For i = 1 to n:
           If iA == 0 and s[i] == A: idxA ← i
           If iA > 0 and s[i] == B: idxB ← i
    If iA > 0 and iB > 0 and iA < iB: return iA
    Else indicar padrão não encontrado
```

a) Calcule a ordem de complexidade deste algoritmo. Justifique e explique a sua análise. (**Nota**: ignore o incremento ao iterador de ciclo e use exatamente as instruções e operações explicitadas na descrição do algoritmo.)

b) Explique, justificando, qual a estratégia geral que este algoritmo está a usar.

c) Consegue melhorar este algoritmo? Explique como e analise a ordem de complexidade nesse caso, justificando.

5. Considere o algoritmo seguinte:

```
Algoritmo matrizOrd?(M, n):
      //Input: M, matriz de dimensão n×n
      //Output: True se M estiver ordenada por ordem crescente
      ordenada = True
      Para i de 1 a n:
         Para j de 2 a n:
             Se M[i, j-1] > M[i, j]:ordenada = False
         Se i < n e ordenada == True: ordenada ← M[i, n] < M[i+1, 1]
      Devolver ordenada
```

a) Este algoritmo está correto?

b) Calcule e indique, justificando, a ordem de complexidade deste algoritmo.
(**Nota**: ignore os incrementos ao iterador de ciclo e use exatamente as instruções e operações explicitadas na descrição do algoritmo.)

c) Explique, justificando, qual a estratégia geral que este algoritmo está a usar.

d) Consegue melhorar este algoritmo? Explique como e analise a ordem de complexidade nesse caso, justificando.

6. Responda às alíneas seguintes.

a) Desenhe um algoritmo recursivo para calcular a potência de grau n de uma dada base, **Algoritmo powern(**base, n**),** e escreva o seu pseudocódigo.

b) Escreva a fórmula recursiva para o trabalho T(n) deste algoritmo, justificando.

c) Calcule a ordem assintótica do limite superior para o trabalho deste algoritmo, justificando.

d) Qual a estratégia geral deste algoritmo? Justifique a sua resposta.

7. Responda às alíneas seguintes.

a) Desenhe um algoritmo recursivo para calcular o factorial de n **- Algoritmo factor(**n**) -** e escreva o seu pseudocódigo.

b) Escreva a fórmula recursiva para o trabalho T(n) deste algoritmo, justificando.

c) Calcule a ordem assintótica do limite superior para o trabalho deste algoritmo, justificando.

d) Qual a estratégia geral deste algoritmo? Justifique a sua resposta.

8. Responda às alíneas seguintes.

a) Desenhe um algoritmo recursivo para calcular se um dado número n é um número par **- Algoritmo par(**n**) -** e escreva o seu pseudocódigo. (Dica: o primeiro número par é o zero e o primeiro ímpar é o 1).

b) Escreva a fórmula recursiva para o trabalho T(n) deste algoritmo. Justifique a sua fórmula.

c) Calcule a ordem de complexidade para o trabalho deste algoritmo, justificando e explicando os dois casos possíveis: se n for par ou se n for ímpar. Termine indicando se a ordem é, ou não exata, e porquê.

d) Qual a estratégia geral deste algoritmo? Justifique a sua resposta.

e) Consegue melhorar este algoritmo? Explique como e indique, justificando, a ordem de complexidade nesse caso.