

LISTA DE EXERCÍCIOS II ANÁLISE DE ALGORITMOS

Professor: Diego Raphael Amancio

PAE: Marcos Vinícius C. Alves

1. Explique as seguintes afirmações:

i) $g(n) \in \mathcal{O}(f(n))$

ii) $g(n) \in \Omega(f(n))$

iii) $g(n) \in \Theta(f(n))$

2. Verifique se as seguintes proposições estão corretas:

i) $7 \in \mathcal{O}(n)$

ii) $n \in \mathcal{O}(1)$

iii) $n + 7 \in \mathcal{O}(n)$

iv) $n + 7 \in \mathcal{O}(1)$

v) $n^2 + 2 \in \mathcal{O}(n)$

vi) $n + 2 \in \mathcal{O}(n^2)$

vii) $3n^3 + n \in \mathcal{O}(n^3)$

viii) $2n^4 \in \mathcal{O}(n^4)$

ix) $n^4 \in \mathcal{O}(2n^4)$

x) $3n^4 + 2n^3 \in \mathcal{O}(2n^4)$

xi) $2n^4 \in \mathcal{O}(3n^4 + 2n^3)$

xii) $\log n \in \mathcal{O}(1)$

xiii) $\log n + 1 \in \mathcal{O}(\log n)$

xiv) $\log n + 1 \in \mathcal{O}(n)$

xv) $\log n + 1 \in \mathcal{O}(n^2)$

xvi) $n * \log n \in \mathcal{O}(1)$

xvii) $n * \log n \in \mathcal{O}(\log n)$

xviii) $n * \log n \in \mathcal{O}(n)$

xix) $n * \log n \in \mathcal{O}(n^2)$

xx) $2 \log n \in \mathcal{O}(n * \log n)$

xxi) $3n * \log n \in \mathcal{O}(\log n)$

xxii) $2n + n \in \mathcal{O}(2^3)$

xxiii) $n^2 \in \mathcal{O}(2^n)$

xxiv) $100n^4 \in \mathcal{O}(2^n)$

xxv) $100n^4 \in \mathcal{O}(n^n)$

xxvi) $2^n \in \mathcal{O}(100n^4)$

xxvii) $2^n \in \mathcal{O}(n^n)$

xxviii) $n^n \in \mathcal{O}(2^n)$

xxix) $n^{100} \in \mathcal{O}(n^n)$

xxx) $\frac{1}{2}n^2 \in \Theta(n^2)$

xxxi) $6n^3 \in \Theta(n^2)$

xxxii) $2^{\frac{n}{2}} \in \Theta(n^2)$

3. 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.

4. Mostre que a seguinte relação de recorrência é $\mathcal{O}(n \log n)$

$$\begin{cases} T(n) = 2T\left(\frac{n}{2}\right) + 7n + 2 & n \geq 1 \\ T(1) = 1 \end{cases}$$

5. Determine o **número máximo** de operações primitivas executadas pelos algoritmos descritos na Página 2, em função da entrada n . Analise a complexidade assintótica destes algoritmos com a notação *big-O*.

Algorithm 1 Valor Máximo

```
1: função arrayMax(A, n)
2:   atualMax  $\leftarrow$  A[0]
3:   para i  $\leftarrow$  1 até n - 1 faça
4:     se A[i] > atualMax então
5:       atualMax  $\leftarrow$  A[i]
6:     fim se
7:     //incremento implícito de i
8:   fim para
9:   devolve atualMax
10: fim função
```

Algorithm 2 Fatorial

```
1: função fat(n)
2:   P  $\leftarrow$  1
3:   para i  $\leftarrow$  1 até i  $\leq$  n faça
4:     P  $\leftarrow$  P * i
5:     //incremento implícito de i
6:   fim para
7:   devolve P
8: fim função
```

Algorithm 3 Ordena Vetor

```
1: função ordenar(A, n)
2:   para j  $\leftarrow$  (n - 1) até 1 faça
3:     para i  $\leftarrow$  0 até j - 1 faça
4:       se A[i] > A[i + 1] então
5:         aux  $\leftarrow$  A[i]
6:         A[i]  $\leftarrow$  A[i + 1]
7:         A[i + 1]  $\leftarrow$  aux
8:       fim se
9:       //incremento implícito de i
10:    fim para
11:    //decremento implícito de j
12:  fim para
13:  devolve A
14: fim função
```

6. Considerando o Algoritmo 3 da questão anterior, quando este algoritmo executaria o número mínimo de operações (melhor caso)?
7. Ordene as seguintes funções de acordo com suas taxas de crescimento assintótico (justifique suas escolhas): $4n * \log(n)$, $2n$, 2^{10} , $2^{\log(n)}$, $3n$, $100 * \log(n)$, $4n$, 2^n , 3^{3n} , $n^2 + 10$, n^3 , $n * \log(n)$
8. Para as relações de recorrência abaixo, quando aplicável, use o Teorema Mestre e determine a notação Θ :
- a)
$$\begin{cases} T(n) = T(n-1) + 3n + 2 & n > 1 \\ T(1) = 1 \end{cases}$$
- b)
$$\begin{cases} T(n) = T(\frac{n}{2}) + 3 & n \geq 1 \\ T(1) = 1 \end{cases}$$
- c)
$$\begin{cases} T(n) = 2T(\frac{n}{2}) + 7n + 2 & n \geq 1 \\ T(1) = 1 \end{cases}$$
9. Encontre as fórmulas fechadas das relações de recorrências da questão anterior.
10. Considere o Algoritmo 4. Suponha que a operação crucial é o fato de inspecionar um elemento. o algoritmo inspeciona os n elementos de um conjunto e, de alguma forma, isso permite descartar $\frac{2}{5}$ dos elementos e então fazer uma chamada recursiva sobre os $\frac{3n}{5}$ elementos restantes.

Algorithm 4 Pesquisa

```

1: função pesq( $n$ )
2:   se  $n \leq 1$  então
3:     inspec-elem e termine
4:   senão
5:     para cada um dos  $n$  elementos inspec-elem
6:       pesq( $\frac{3n}{5}$ )
7:   fim se
8: fim função

```

- a) Escreva a equação de recorrência que descreva este comportamento.
- b) Converta esta equação para um somatório.
- c) Dê a fórmula fechada para este somatório.
11. Apresente um algoritmo para obter o maior e o segundo maior elementos de um conjunto. Apresente também a análise do algoritmo. Você acha seu algoritmo eficiente? Por quê? Procure comprovar suas respostas.

Referências

- [1] J. a. Luís, *Listas de Exercícios – SCC201*. ICMC–USP, Agosto 2009.
- [2] A. R. da Cruz, *Listas de Exercícios – SIN213*. UFV–CRP, Setembro 2011.
- [3] R. Campello, *Listas de Exercícios – SCC5763*. ICMC–USP, Abril 2014.