Exercícios da Unidade III
Fundamentos de Análise de Algoritmos

Camila Morereira Lopes[1]

[1]Instituto de Ciências Exatas e Informática - Pontífica Universidade Católica de Minas Gerais

camila.lopes.1264894@sga.pucminas.br

1. Exercícios Feitos

Exercício resolvido 10 - pg. 59

O aluno deve escolher a primeira opção, pois a pesquisa sequencial tem custo Θ(n). Já a segunda opção tem Θ((n+1) $\times$ ln n)

Exercício resolvido 11 - pg. 65

a) $3n^2 + 5n + 1$ é $O(n)$: falsa
b) $3n^2 + 5n + 1$ é $O(n^2)$: verdadeira
c) $3n^2 + 5n + 1$ é $O(n^3)$: verdadeira
d) $3n^2 + 5n + 1$ é $\Omega(n)$: verdadeira
e) $3n^2 + 5n + 1$ é $\Omega(n^2)$: verdadeira
f) $3n^2 + 5n + 1$ é $\Omega(n^3)$: falsa
g) $3n^2 + 5n + 1$ é $\Theta(n)$: falsa
h) $3n^2 + 5n + 1$ é $\Theta(n^2)$: verdadeira
i) $3n^2 + 5n + 1$ é $\Theta(n^3)$: falsa

1.1. Exercício resolvido 12 - pg. 84

Apresente a função e a complexidade para os números de comparações e movimentações de registros para o pior e melhor caso

```java
public static void imprimirMaxMin(int[] array, int n){
    int maximo, minimo;

    if(array[0] > array[1]){
        maximo = array[0]; minimo = array[1];
    }else{
        maximo = array[1]; minimo = array[0];
    }

    for(int i = 2; i < n; i++){
        if(array[i] > maximo){
            maximo = array[i];
        } else if(array[i] < minimo){
            minimo = array[i];
        }
    }
}
```

<u>Função de complexidade</u>

	MOV	CMP
PIOR	$f(n) = 2 + (n\breve{}2)$	$f(n) = 1 + 2(n\breve{}2)$
MELHOR	$f(n) = 2 + (n\breve{}2)x0$	$f(n) = 1 + (n\breve{}2)$

<u>Complexidade</u>

	MOV	CMP
PIOR	$O(n), \Omega(n) e \Theta(n)$	$O(n), \Omega(n) e \Theta(n)$
MELHOR	$O(1), \Omega(1) e \Theta(1)$	$O(n), \Omega(n) e \Theta(n)$

1.2. Exercício resolvido 13 - pg. 86

Apresente a função e a complexidade para o número de subtrações para o pior e melhor caso

```
        i = 0;

        while(i < n){
            i++;
            a--;
        }

        if(b > c){
        i--
        } else{
            i--;
            a--;
        }
```

	função	complexidade
PIOR	$f(n) = n + 2$	$O(n), \Omega(n) e \Theta(n)$
MELHOR	$f(n) = n + 1$	$O(n), \Omega(n) e \Theta(n)$

1.3. Exercício resolvido 14 - pg. 88

Apresente a função e a complexidade para o número de subtrações para o pior e melhor caso

```
    for(i = 0; i < n; i++){
        for(j = 0; j < n; j++){
            a--;
            b--;
        }
        c--;
    }
```

	função	complexidade
TODOS	$f(n) = (2n + 1)$	$O(n^2), \Omega(n^2) e \Theta(n^2)$

1.4. Exercício resolvido 15 - pg. 90

Apresente a função e a complexidade para o número de subtrações para o pior e melhor caso

```
    for(i = 0; i < n; i++){
        for(j = 0; j <= n; j*=2){
            b--;
        }
    }
```

	função	complexidade
TODOS	$f(n) = n * lg(n) + n$	$O(n \times lg(n)), \Omega(n \times lg(n)) e \Theta(n \times lg(n)$

1.5. Exercício resolvido 16 - pg. 94

Apresente o tipo de crescimento que melhor caracteriza as funções abaixo (Khan Academy, adaptado)

	Constante	**Linear**	**Polinomial**	**Exponencial**
3n		■		
1	■			
(3/2)n		■		
$2n^3$			■	
2^n				■
$3n^2$			■	
1000	■			
$(3/2)n^n$				■

1.6. Exercício resolvido 17 - pg. 97

Classifique as funções $f_1(n) = n^2$, $f_2(n) = n$, $f_3(n) = 2^n$, $f_4(n) = (3/2)^n$, $f_5(n) = n^3$ e $f_6(n) = 1$ de acordo com o crescimento, do mas lento para o mais rápido (Khan Academy, adaptado)

1. $f_6(n) = 1$
2. $f_2(n) = n$
3. $f_1(n) = n^2$
4. $f_5(n) = n^3$
5. $f_4(n) = (3/2)^n$
6. $f_3(n) = 2^n$

1.7. Exercício resolvido 18 - pg. 99

Classifique as funções $f_1(n) = n * \log_6(n)$, $f_2(n) = lg(n)$, $f_3(n) = \log_8(n)$, $f_4(n) = 8n^2$, $f_5(n) = n * lg(n)$, $f_6(n) = 64$, $f_7(n) = 6n^3$, $f_8(n) = 8^{2n}$ e $f_9(n) = 4n$ de acordo com o crescimento, do mas lento para o mais rápido (Khan Academy, adaptado)

1. $f_6(n) = 64$
2. $f_3(n) = \log_8(n)$
3. $f_2(n) = lg(n)$
4. $f_9(n) = 4n$
5. $f_1(n) = n * \log_6(n)$
6. $f_5(n) = n * lg(n)$
7. $f_4(n) = 8n^2$
8. $f_7(n) = 6n^3$
9. $f_8(n) = 8^{2n}$

1.8. Exercício resolvido 19 - pg. 100

Faça a correspondência entre cada função f(n) com sua g(n) equivalente, em termos de Θ. Essa correspondência acontece quando f(n) = Θ(g(n)) ()Khan Academy, adaptado)

f(n)	**g(n)**
n + 30	3n - 1
$n^2 + 2n - 10$	$n^2 + 3n$
$n^3 * 3$	n^4
lg(n)	lg(2n)

2. Exercícios

Exercício 01 e 02 - pg. 56 e 57

Segundo o livro: Projeto de Algoritmos com Implementações em JAVA e C++ do autor Ziviani o código criado, responsável por encontrar o maior e menor valor em um array de inteiros, possui as seguintes funções de complexidade de tempo:

- Pior caso O: 2(n - 1)
- Caso Médio Θ: $\frac{3n-3}{2}$
- Melhor Caso Ω: n - 1

Exercício 03 - pg. 69

	O(1)	O(lg n)	O(n)	O(n.lg(n))	O(n^2)	O(n^3)	O(n^5)	O(n^{20})
f(n) = lg(n)								
f(n) = n . lg(n)								
f(n) = 5n + 1								
f(n) = $7n^5 - 3n^2$								
f(n) = $99n^3 - 1000n^2$								
f(n) = $n^5 - 99999n^4$								

Exercício 04 - pg. 70

	$\Omega(1)$	$\Omega(\lg n)$	$\Omega(n)$	$\Omega(n.\lg(n))$	$\Omega(n^2$	$\Omega(n^3)$	$\Omega(n^5)$	$\Omega(n^{20})$
f(n) = lg(n)								
f(n) = n . lg(n)								
f(n) = 5n + 1								
f(n) = $7n^5$ - $3n^2$								
f(n) = $99n^3$ - $1000n^2$								
f(n) = n^5 - $99999n^4$								

Exercício 05 - pg. 71

	$\Theta(1)$	$\Theta(\lg n)$	$\Theta(n)$	$\Theta(n.\lg(n))$	$\Theta(n^2)$	$\Theta(n^3)$	$\Theta(n^5)$	$\Theta(n^{20})$
f(n) = lg(n)								
f(n) = n . lg(n)								
f(n) = 5n + 1								
f(n) = $7n^5$ - $3n^2$								
f(n) = $99n^3$ - $1000n^2$								
f(n) = n^5 - $99999n^4$								

Exercício 06 - pg. 73

Qual é a ordem de complexidade das operações:

a) 02 -> f(n) + g(n) - h(n)
b) 01 -> O(f(n) + O(g(n)) - O(h(n))
c) 04 -> f(n) $\times$ g(n)
d) 06 -> g(n) $\times$ l(n) + h(n)
e) 05 -> f(n) $\times$ g(n) $\times$ l(n)
f) 03 -> O(O(O(O(f(n)))))

Exercício 07 - pg. 76

Dada a definição da notação O:

a) Mostre um valor c e outro m tal que, para n $\geq$ m, $|3n^2 + 5n + 1| \leq c \times |n^2|$, provando que $3n^2 + 5n + 1$ é $O(n^2)$
Consideramos que o valor de c é 4 e m ocorre entre os valores 5 e 6 de n

n	$3n^2 + 5n + 1$	$4 \times n^2$
1	9	4
2	23	16
3	43	36
4	69	64
5	101	100
6	139	144
7	183	196

b) Mostre um valor c e outro m tal que, para $n \geq m$, $|3^2 + 5n + 1| \leq c \times |n^3|$, provando que $3n^2 + 5n + 1$ é $O(n^3)$
Consideramos que o valor de c é 1 e m ocorre entre os valores 4 e 5 de n

n	$3^2 + 5n + 1$	$1 \times n^3$
1	9	1
2	23	8
3	43	27
4	69	64
5	101	125
6	139	216
7	183	343

c) Prove que $3^2 + 5n + 1$ não é $O(n)$
Para a função ser O(n) seria necessário que em algum momento o valor de n fosse maior que o valor de f(n)

n	$3^2 + 5n + 1$	n
1	9	1
2	23	2
3	43	3
4	69	4
5	101	5
6	139	6
7	183	7

Exercício 08 - pg. 78

Dada a definição da notação Ω:

a) Mostre um valor c e outro m tal que, para $n \geq m$, $|g(n)| \leq c \times |f(n)|$, provando que $3n^2 + 5n + 1$ é $\Omega(n^2)$
Consideramos que o valor de c é 3

n	$3n^2 + 5n + 1$	$3 \times n^2$
1	9	3
2	23	6
3	43	27
4	69	48
5	101	75
6	139	108
7	183	147

b) Mostre um valor c e outro m tal que, para $n \geq m$, $|g(n)| \leq c \times |f(n)|$, provando que $3n^2 + 5n + 1$ é $\Omega(n)$ Consideramos que o valor de c é 1

n	$3^2 + 5n + 1$	1 x n
1	9	1
2	23	2
3	43	3
4	69	4
5	101	5
6	139	6
7	183	7

c) Prove que $3^2 + 5n + 1$ não é $\Omega(n^3)$
Para a função ser $\Omega(n)$ seria necessário que em algum momento o valor de n^3 se tornasse menor que o valor de f(n)

n	$3^2 + 5n + 1$	n^3
1	9	1
2	23	8
3	43	27
4	69	64
5	101	125
6	139	216
7	183	343

Exercício 09 - pg. 80

Dada a definição da notação Θ:

a) Mostre um valor c_1, c_2 e m tal que, para $n \geq m$, $c_1 \times |f(n)| \leq |g(n)| \leq c_2 \times |f(n)|$, provando que $3n^2 + 5n + 1$ é $\Theta(n^2)$
Consideramos que o valor de c_1 é 3 e c_2 é 4

n	$3n^2 + 5n + 1$	$3 \times n^2$	$4 \times n^2$
1	9	3	4
2	23	6	16
3	43	27	36
4	69	48	64
5	101	75	100
6	139	108	144
7	183	147	196

b) Prove que $3n^2 + 5n + 1$ não é $\Theta(n)$
Consideramos que o valor de c_1 é 1 e c_2 é 4. Para a função ser $\Theta(n\ )$ seria necessário que em algum momento o valor de c_2 x n se tornasse maior que o valor de f(n)

n	$3^2 + 5n + 1$	1 x n	4 x n
1	9	1	4
2	23	2	8
3	43	3	12
4	69	4	16
5	101	5	20
6	139	6	24
7	183	7	28

c) Prove que $3^2 + 5n + 1$ não é $\Theta(n^3)$
Consideramos que o valores de c_1e c_2 são igual a 1. Para a função ser $\Theta(n^3\ )$ seria necessário que em algum momento o valor de c_1 x n^3 se tornasse menor que o valor de f(n)

n	$3^2 + 5n + 1$	n^3
1	9	1
2	23	8
3	43	27
4	69	64
5	101	125
6	139	216
7	183	343

Exercício 10 - pg. 83

Presente em outro pdf com o nome Análise de Algoritmos

Exercício 11 - pg. 92

Suponha um sistema de monitoramento contendo os métodos telefone, luz, alarme, sensor e câmera, apresente a função e ordem de complexidade para o pior e melhor caso: (a) método alarme; (b) outros métodos;

```
public static void sistemaMonitoramento(){
    if(telefone() == true && luz() = true) alarme(0);
    else alarme(1);
    for(int i = 2; i < n; i++{
        if(sensor(i-2) == true) alarme(i-2);
        else if(camera(i-2) == true) alarme(i-2+n);
    }
}
```

	Melhor Caso	Pior Caso
Telefone	1, $O(1)\Theta(1)\Omega(1)$	1, $O(1)\Theta(1)\Omega(1)$
Alarme	1, $O(1)\Theta(1)\Omega(1)$	$1 + 2(n-2), O(n)\Theta(n)\Omega(n)$
Luz	0, $O(0)\Theta(0)\Omega(0)$	$O(1)\Theta(1)\Omega(1)$
Sensor	$(n-2), O(n)\Theta(n)\Omega(n)$	$(n-2), O(n)\Theta(n)\Omega(n)$
Câmera	$(n-2), O(n)\Theta(n)\Omega(n)$	$(n-2), O(n)\Theta(n)\Omega(n)$

Exercício 12 - pg. 93

Apresente um código, defina duas operações relevantes e apresente a função e a complexidade para as operações escolhidas no pior e melhor caso.

Exercício 13 - pg. 102

No Exercício Resolvido (10), verificamos que quando desejamos pesquisar a existência de **um** elemento em um array de números reais é adequado executar uma pesquisa sequencial cujo custo é Θ(n). Nesse caso, o custo de ordenar o array e, em seguida, aplicar uma pesquisa binária é mais elevado, Θ(n * lg(n)) + Θ(lg(n)) = Θ(n * lg(n)). Agora, supondo que desejamos efetuar **n** pesquisas, responda qual das duas soluções é mais eficiente

R.: No caso apresentado a solução mais eficiente é ordenar o array e aplicar a pesquisa binária, pois ao realizar as **n** pesquisas sequenciais, a complexidade seria de $\Theta(n^2)$. Já a complexidade ao ordenar e pesquisar binariamente seria de Θ(n * lg(n));

Referências

Nivio Ziviani. *PROJETO DE ALGORITMOS COM IMPLEMENTAÇÕES EM JAVA E C++*. Cengage Learning Edições Ltda., October 2006. ISBN 978-8522105250. book.