

AED II - Unidade 03 - Fundamentos de

Luca Ribeiro Schettino Regne

1

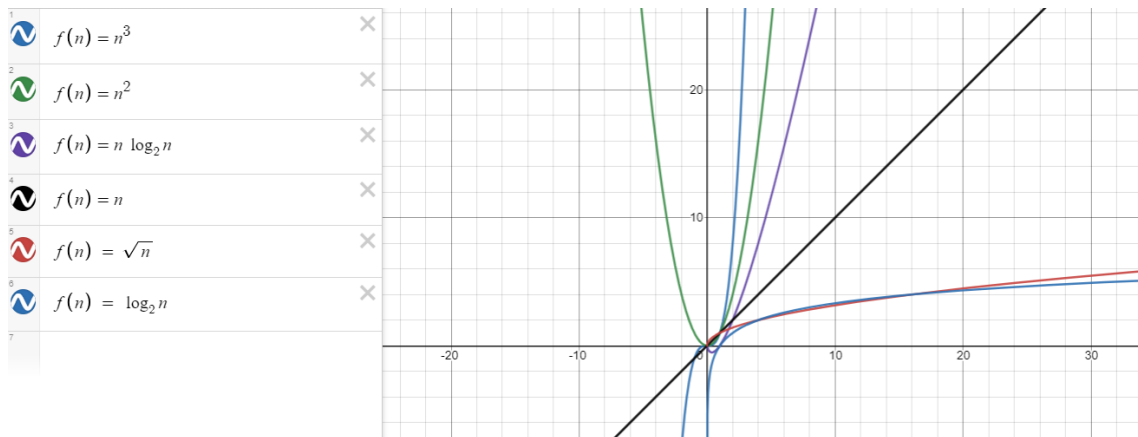
1. Exercícios Resolvidos

1.1. Resolva as equações abaixo:

- a) $2^{10} = 1024$
- b) $\lg(1024) = 10$
- c) $\lg(17) = 4,0875$
- d) $\lceil \lg(17) \rceil = 5$
- e) $\lfloor \lg(17) \rfloor = 4$

1.2. Calcule o número de subtrações que o código abaixo realiza:

```
1 f (i a--; b--; --;  
2 ]Plote um gráfico com todas as funções abaixo) a)  $f(n) = n^3$   
b)  $f(n) = n^2$   
c)  $f(n) = n \lg n$   
d)  $f(n) = n$   
e)  $f(n) = \sqrt{n}$   
f)  $f(n) = \lg n$ 
```



1.3. Calcule o número de subtrações que o código abaixo realiza:

```
1 f (i a--; b--; --;  
Melhor caso:  $f(n) = 2n(O(n), \Theta(n), \Omega(n))$  Pior caso:  $f(n) = 2n(O(n), \Theta(n), \Omega(n))$ 
```

1.4. Calcule o número de subtrações que o código abaixo realiza:

```
1 ...  
2 for (int i = 3; i < n; i++) {  
3     if (i % 2 == 0) {  
4         a--;  
5     }  
6 }
```

Subtrações = $n - 3$.
 $O(n)$, $\Theta(n)$, $\Omega(n)$

1.5. Calcule o número de multiplicações que o código abaixo realiza:

```
1  for(int i = n; i > 2; i /= 2){
2      a *= 2;
3  }
```

Multiplicações = $\lceil \lg(n) \rceil + 1$
 $O(\lg n)$, $\Theta(\lg n)$ e $\Omega(\lg n)$

1.6. Outra forma de compreender o código anterior é executando o mesmo

```
1  class Log {
2      public static void main (String[] args){
3          int[] n =
4          {4,5,6,7,8,9,10,11,12,13,14,15,16,17,31,32,33,63,64,65};
5          int cont;
6          for(int k = 0; k < n.length; k++){
7              System.out.print("\nn[n = " + n[k] + "] => ");
8              cont = 0;
9              for(int i = n[k]; i > 0; i /= 2){
10                 System.out.print(" " + i);
11                 cont++;
12             }
13             System.out.print(" (" + cont + " vezes)");
14         }
15         System.out.print("\n");
16     }
```

1.7. Encontre o menor valor em um array de inteiros

```
1  int min = array[0];
2  for (int i = 1; i < n; i++){
3      if (min > array[i]){
4          min = array[i];
5      }
6  }
```

1. Qual é a operação relevante?

Comparação entre elementos do array

2. Quantas vezes ela será executada?

Se tivermos n elementos: $T(n) = n - 1$

3. O nosso $T(n) = n - 1$ é para qual dos três casos?

Para todos os casos. $O(n)$, $\Theta(n)$, $\Omega(n)$

1.8. Encontrar Mínimo

```
1  int min = array[0];
2  for (int i = 1; i < n; i++){
3      if (min > array[i]){
4          min = array[i];
5      }
6  }
```

1º) Qual é a operação relevante?

Comparação entre elementos do array

2º) Quantas vezes ela será executada?

Se tivermos n elementos: $T(n) = n - 1$

3º) O nosso $T(n) = n - 1$ é para qual dos três casos?

Para os três casos

4º) O nosso algoritmo é ótimo? Por que?

Sim, porque temos que testar todos os elementos para garantir nossa resposta

1.9. Pesquisa Sequencial

```
1  boolean resp = false;
2  for (int i = 0; i < n; i++){
3      if (array[i] == x){
4          resp = true;
5          i = n;
6      }
7  }
```

1º) Qual é a operação relevante?

Comparação entre elementos do array

2º) Quantas vezes ela será executada?

Melhor caso: $f(n) = 1$

Pior caso: $f(n) = n$

Caso médio: $f(n) = (n + 1) / 2$

3º) O nosso algoritmo é ótimo? Por que?

Sim porque temos que testar todos os elementos para garantir nossa resposta.

1.10. Um aluno deve procurar um valor em um array de números reais. Ele tem duas alternativas. Primeiro, executar uma pesquisa sequencial. Segundo, ordenar o array e, em seguida, aplicar uma pesquisa binária. O que fazer?

O aluno deve escolher a primeira opção, pois a pesquisa sequencial tem custo $\Omega(n)$. A segunda opção tem custo $\Omega(n * \lg n)$ para ordenar mais $\Omega(\lg n)$ para a pesquisa binária.

1.11. Responda se as afirmações são verdadeiras ou falsas:

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 $O(n^3)$: Falsa

d) $3n^2 + 5n + 1$ é $\Theta(n)$: Falsa

e) $3n^2 + 5n + 1$ é $\Theta(n^2)$: Verdadeira

f) $3n^2 + 5n + 1$ é $\Theta(n^3)$: Verdadeira

g) $3n^2 + 5n + 1$ é $\Omega(n)$: Falsa

h) $3n^2 + 5n + 1$ é $\Omega(n^2)$: Verdadeira

i) $3n^2 + 5n + 1$ é $\Omega(n^3)$: Falsa

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

```
1 void imprimirMaxMin(int[] array, int n){
2     int maximo, minimo;
3     if (array[0] > array[1]){
4         maximo = array[0];
5         minimo = array[1];
6     }else{
7         maximo = array[1];
8         minimo = array[0];
9     }
10
11     for(int i = 2; i < n; i++){
12         if (array[i] > maximo){
13             maximo = array[i];
14         } else if(array[i] < minimo){
15             minimo = array[i];
16         }
17     }
18 }
```

Função de complexidade - MOV

PIOR: $f(n) = 2 + (n-2)$

MELHOR: $f(n) = 2 + (n-2) \times 0$

Complexidade - MOV

PIOR: $O(n)$, $\Theta(n)$ e $\Omega(n)$

MELHOR: $O(1)$, $\Theta(1)$ e $\Omega(1)$

Função de complexidade - CMP

PIOR: $f(n) = 1 + 2(n-2)$

MELHOR: $f(n) = 1 + (n-2)$

Complexidade - CMP

PIOR: $O(n)$, $\Theta(n)$ e $\Omega(n)$

MELHOR: $O(n)$, $\Theta(n)$ e $\Omega(n)$

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

```
1 i = 0;
2 while (i < n) {
3     i++;
4     a--;
5 }
6
7 if (b > c) {
8     i--;
9 } else {
10    i--;
11    a--;
12 }
```

Função de complexidade

PIOR: $f(n) = n + 2$

MELHOR: $f(n) = n + 1$

Complexidade

PIOR: $O(n)$, $\Theta(n)$ e $\Omega(n)$

MELHOR: $O(n)$, $\Theta(n)$ e $\Omega(n)$

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

```
1  for (i = 0; i < n; i++) {
2      for (j = 0; j < n; j++) {
3          a--;
4          b--;
5      }
6      c--;
7  }
```

Função de complexidade: $f(n) = (2n + 1)n$
Complexidade: $O(n^2)$, $\Theta(n^2)$ e $\Omega(n^2)$

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

```
1  for (i = 0; i < n; i++) {
2      for (j = 1; j <= n; j *= 2) {
3          b--;
4      }
5  }
```

Função de complexidade: $f(n) = (\lg(n) + 1) * n = n * \lg(n) + n$
Complexidade: $O(n \lg(n))$, $\Theta(n \lg(n))$ e $\Omega(n \lg(n))$

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

```
1  for (i = 0; i < n; i++) {
2      for (j = 1; j <= n; j *= 2) {
3          b--;
4      }
5  }
```

Função de complexidade: $(\lg(n) + 1) * n = n * \lg(n) + n$
Complexidade: $O(n \lg(n))$, $\Theta(n \lg(n))$ e $\Theta(n \lg(n))$

1.17. Apresente o tipo de crescimento que melhor caracteriza as funções abaixo

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

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

$$\begin{aligned}
 f_6(n) &= 1 \\
 f_2(n) &= n \\
 f_1(n) &= n^2 \\
 f_5(n) &= n^3 \\
 f_4(n) &= (3/2)^n \\
 f_3(n) &= 2n
 \end{aligned}$$

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

$$\begin{aligned}
 f_6(n) &= 64 \\
 f_3(n) &= \log_8(n) \\
 f_2(n) &= \lg(n) \\
 f_9(n) &= 4n \\
 f_1(n) &= n \cdot \log_6(n) \\
 f_5(n) &= n \cdot \lg(n) \\
 f_4(n) &= 8n^2 \\
 f_7(n) &= 6n^3 \\
 f_8(n) &= 8^{2n}
 \end{aligned}$$

1.20. Faça a correspondência entre cada função $f(n)$ com sua $g(n)$ equivalente, em termos de . Essa correspondência acontece quando $f(n) = (g(n))$

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

2. Exercícios

- 2.1. Encontre o maior e menor valores em um array de inteiros e, em seguida, encontre a função de complexidade de tempo para sua solução
- 2.2. Considerando o problema de encontrar o maior e menor valores em um array de inteiros, veja os quatro códigos propostos e analisados no livro do Ziviani
- 2.3. Preencha verdadeiro ou falso na tabela abaixo:

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

2.4. Preencha verdadeiro ou falso na tabela abaixo:

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

2.5. Preencha verdadeiro ou falso na tabela abaixo:

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

2.6. Dado $f(n) = 3n^2 - 5n - 9, g(n) = n * \lg(n), l(n) = n \cdot \lg^2(n)$ e $h(n) = 99n^8$, qual é a ordem de complexidade das operações:

- $f(n) + g(n) - h(n)$
- $O(f(n) + O(g(n)) - O(h(n)))$
- $f(n) \times g(n)$
- $g(n) \times l(n) + h(n)$
- $f(n) \times g(n) \times l(n)$
- $O(O(O(O(f(n))))))$

2.7. Dada a definição da notação O:

- Mostre um valor c e outro m tal que, para $n \geq m$, $|3n^2 + 5n + 1| \leq c \cdot |n^2|$, provando que $3n^2 + 5n + 1$ é $O(n^2)$
- Mostre um valor c e outro m tal que, para $n \geq m$, $|3n^2 + 5n + 1| \leq c \cdot |n^3|$, provando que $3n^2 + 5n + 1$ é $O(n^3)$
- Prove que $3n^2 + 5n + 1$ não é $O(n)$

2.8. Dada a definição da notação Θ :

- Mostre um valor c e outro m tal que, para $n \geq m$, $|3n^2 + 5n + 1| \leq c \cdot |n^2|$, provando que $3n^2 + 5n + 1$ é $\Theta(n^2)$
- Mostre um valor c e outro m tal que, para $n \geq m$, $|3n^2 + 5n + 1| \leq c \cdot |n^3|$, provando que $3n^2 + 5n + 1$ é $\Theta(n^3)$
- Prove que $3n^2 + 5n + 1$ não é $\Theta(n)$

2.9. Dada a definição da notação Ω :

- Mostre um valor c e outro m tal que, para $n \geq m$, $|3n^2 + 5n + 1| \leq c \cdot |n^2|$, provando que $3n^2 + 5n + 1$ é $\Omega(n^2)$
- Mostre um valor c e outro m tal que, para $n \geq m$, $|3n^2 + 5n + 1| \leq c \cdot |n^3|$, provando que $3n^2 + 5n + 1$ é $\Omega(n^3)$
- Prove que $3n^2 + 5n + 1$ não é $\Omega(n)$