Exercício Resolvido (1): Resolva as Equações

a) 2^{10} =

b) lg(1024) =

c) lg(17) =

d) $\lceil lg(17) \rceil$ =

e) $\lfloor lg(17) \rfloor$ =

a) 1024
b) 10
c) 4,08746284125034
d) 5
e) 4

Exercício Resolvido (2): Plote os Gráficos

a) f(n) = n^3

b) f(n) = n^2

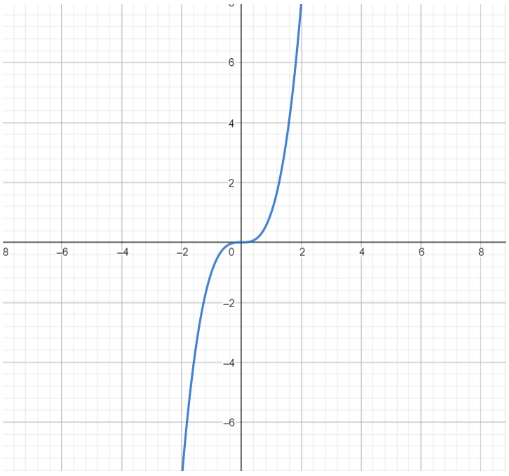
c) f(n) = n x lg(n)

d) f(n) = n

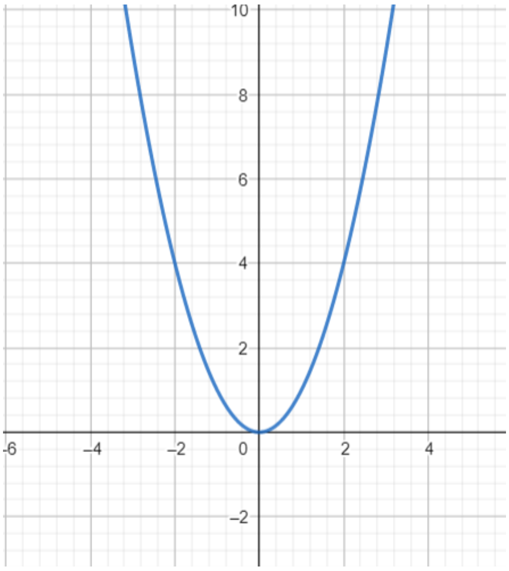
e) f(n) = sqrt(n)

f) f(n) = lg(n)

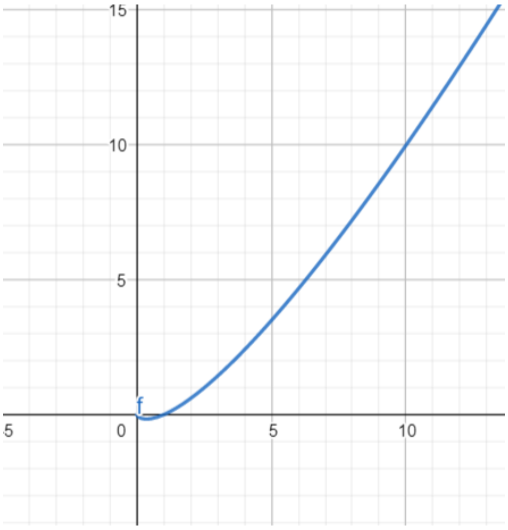
a)



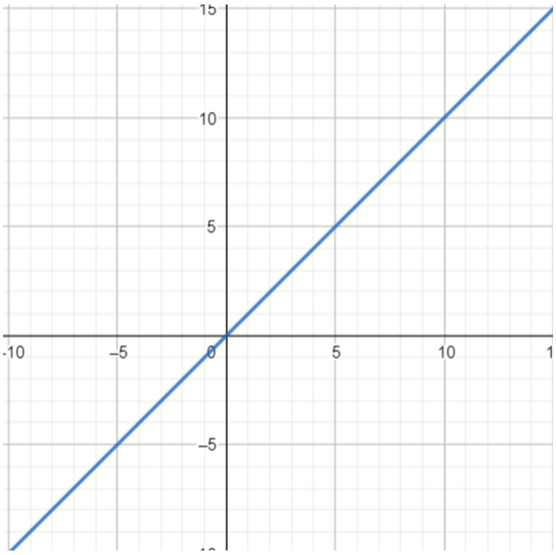
b)



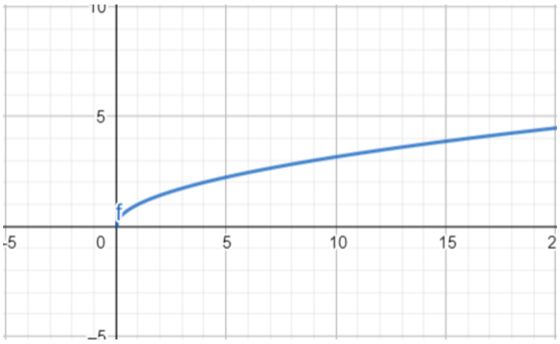
c)



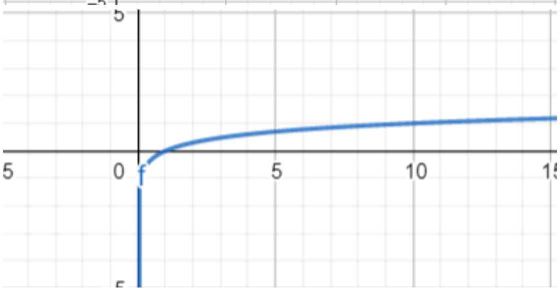
d)



e)



f)



Exercício Resolvido (3)

- Calcule o **número de subtrações** que o código abaixo realiza:

```
...
if (a - 5 < b - 3){
        i--;
        --b;
        a -= 3;
} else {
        j--;
}
```


a) //Melhor caso: f(n) = 3, $\Theta(1)$
   //Pior caso: f(n) = 5, $\Theta(1)$

Exercício Resolvido (4)

- Calcule o **número de subtrações** que o código abaixo realiza:

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


a) //Todos os casos: f(n) = 2n, Θ (n)

Exercício Resolvido (5)

- Calcule o **número de subtrações** que o código abaixo realiza:

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


a) $f(n) = 3n^2$, $\Theta(n^2)$

Exercício Resolvido (6)

- Calcule o **número de multiplicações** que o código abaixo realiza:

```
...
for (int i = n; i > 0; i /= 2){
        a *= 2;
}
```


a) $f(n) = \lfloor lg(n) \rfloor + 1$, $\Theta(lg\ n)$

Exercício Resolvido (7): Pesquisa Sequencial

- Apresente a função de complexidade de tempo (número de comparações entre elementos do *array*) da pesquisa sequencial no melhor e no pior caso

```
boolean resp = false;

for (int i = 0; i < n; i++){
        if (array[i] == x){
                resp = true;
                i = n;
        }
}
```

Este algoritmo é ótimo?

PUC Minas Virtual

a) Melhor caso: elemento desejado na primeira posição
t(n) = 1

Pior caso: elemento desejado não está no array ou está na última posição
t(n) = n

Exercício Resolvido (8)

- 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?

PUC Minas Virtual

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

Exercício Resolvido (9)

- Responda se as afirmações são verdadeiras ou falsas:
  a) $3n^2 + 5n + 1$ é $O(n)$:
  b) $3n^2 + 5n + 1$ é $O(n^2)$:
  c) $3n^2 + 5n + 1$ é $O(n^3)$:
  d) $3n^2 + 5n + 1$ é $\Omega(n)$:
  e) $3n^2 + 5n + 1$ é $\Omega(n^2)$:
  f) $3n^2 + 5n + 1$ é $\Omega(n^3)$:
  g) $3n^2 + 5n + 1$ é $\Theta(n)$:
  h) $3n^2 + 5n + 1$ é $\Theta(n^2)$:
  i) $3n^2 + 5n + 1$ é $\Theta(n^3)$:

PUC Minas Virtual

a) falsa
b) verdadeira
c) verdadeira
d) verdadeira
e) verdadeira
f) falsa
g) falsa
h) verdadeira
i) falsa

Exercício Resolvido (10)

- Sabendo que o Algoritmo de Seleção faz $\Theta(n^2)$ comparações entre registros, quantas dessas comparações temos no código abaixo? Justifique

```
for (int i = 0; i < n; i++){
        seleção();
}
```

PUC Minas Virtual

a) Neste caso, executamos o Seleção n vezes: n x $\Theta(n^2) = \Theta(n^3)$

Exercício Resolvido (11)

- Dado $f(n) = 3n^2 - 5n - 9$, $g(n) = n.lg(n)$, $l(n) = n.lg^2(n)$ e $h(n) = 99n^8$, qual é a ordem de complexidade das operações abaixo (use a notação Θ):

  a) $h(n) + g(n) - f(n)$

  b) $\Theta(h(n)) + \Theta(g(n)) - \Theta(f(n))$

  c) $f(n)\ x\ g(n)$

  d) $g(n)\ x\ l(n) + h(n)$

  e) $f(n)\ x\ g(n)\ x\ l(n)$

  f) $\Theta(\Theta(\Theta(\Theta(f(n)))))$

a) [99n8] + [n.lg(n)] - [3n2 -5n-9] ⇒ Θ(n^8)
b) Θ(n8) + Θ(n.lg(n)) - $\Theta(n^2)$ ⇒ Θ(n^8)
c) Θ(n2) x Θ(n.lg(n)) ⇒ $\Theta(n^3 .lg(n))$
d) Θ(n.lg(n)) x Θ(n.lg2 (n)) + Θ(n8) ⇒ Θ(n^8)
e) Θ(n2) x Θ(n.lg(n)) x Θ(n.lg2 (n)) ⇒ Θ(n^4 .lg^3 (n))
f) Θ(n)

Exercício Resolvido (12)

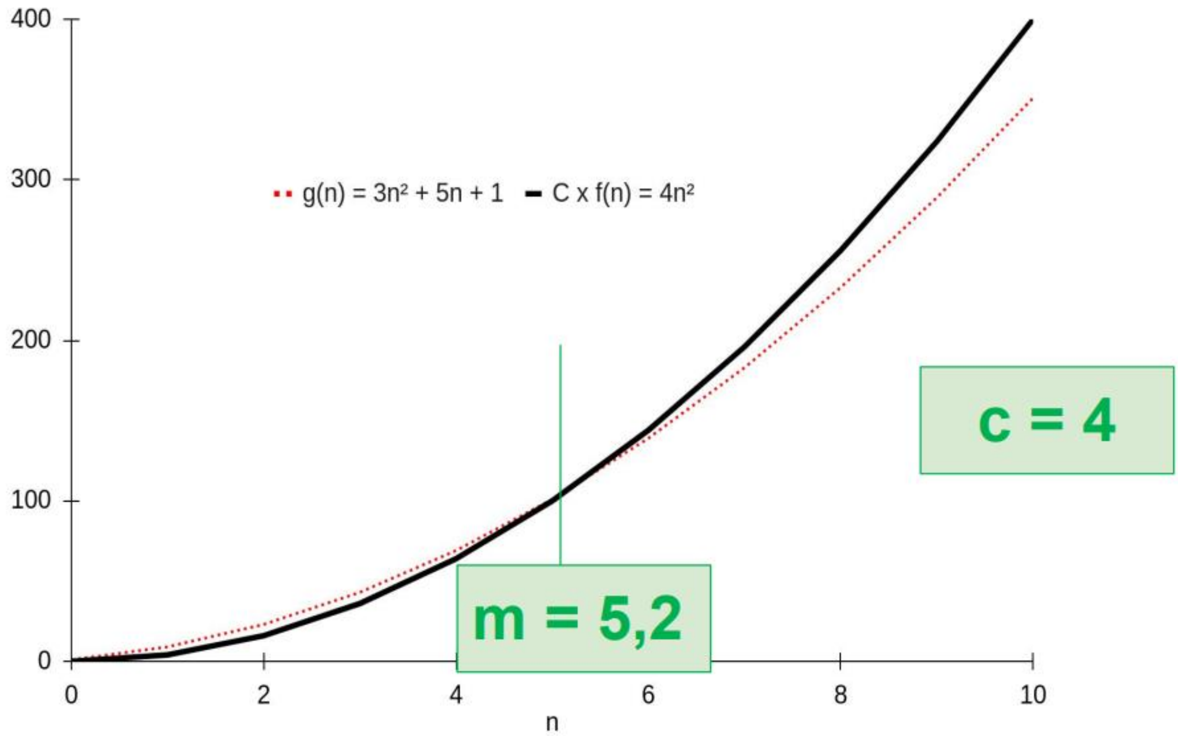
- Dada a definição da notação O:

  a) Mostre os valores de c e m tal que, para $n \geq m$, $|3n^2 + 5n + 1| \leq c \times |n^2|$, provando que $3n^2 + 5n + 1$ é $O(n^2)$

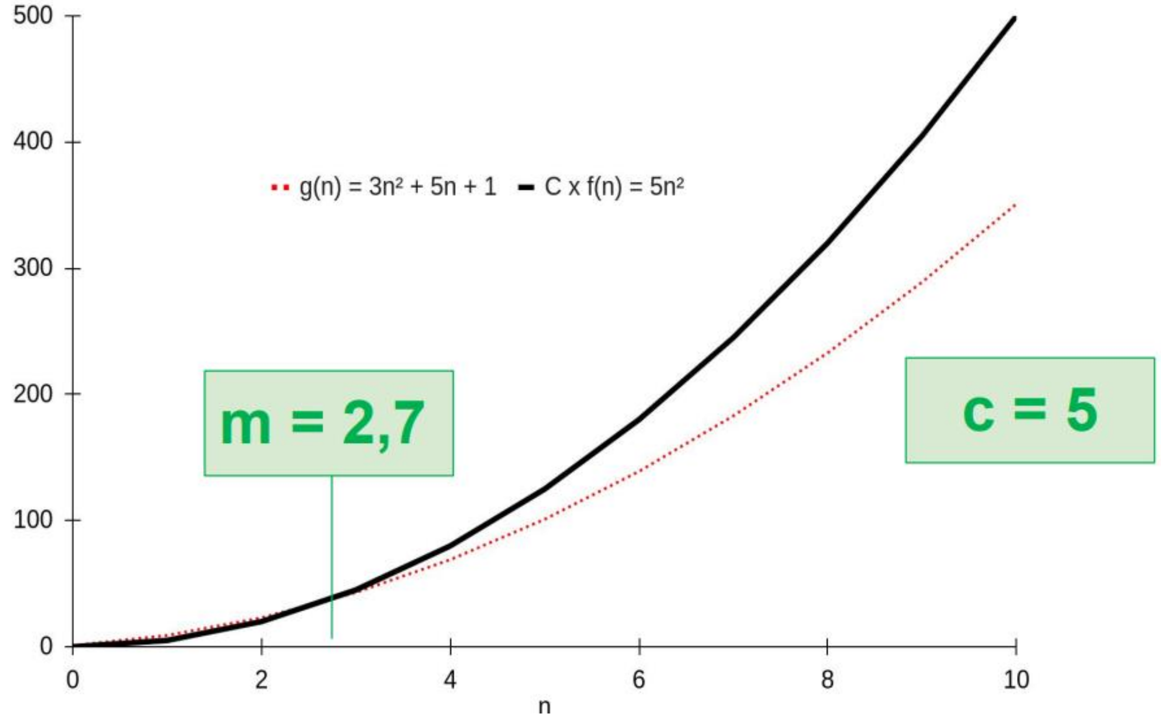
  b) Mostre os valores de c e m tal que, para $n \geq m$, $|3n^2 + 5n + 1| \leq c \times |n^3|$, provando que $3n^2 + 5n + 1$ é $O(n^3)$

  c) Prove que $3n^2 + 5n + 1$ **não é** $O(n)$

a) Para que tal inequação seja verdadeira, c tem que ser maior do que três (e.g., quatro)

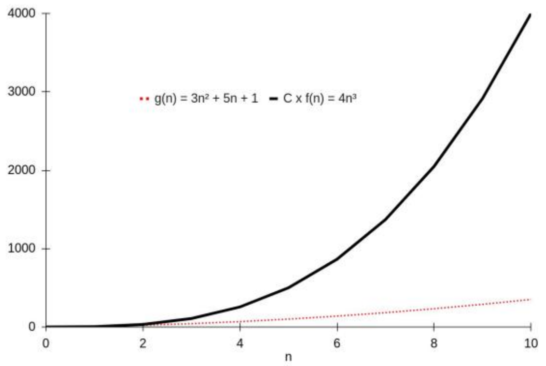


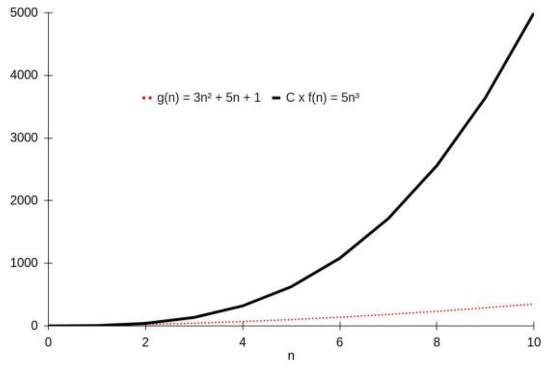
400
300
200
100
0
0
2
4
6
8
10
n
g(n) = 3n² + 5n + 1
C x f(n) = 4n²
c = 4
m = 5,2



500
400
300
200
100
0
0
2
4
6
8
10
n
g(n) = 3n² + 5n + 1
C x f(n) = 5n²
c = 5
m = 2,7

b) Novamente, (c = 4 e m = 5,7) e (c = 5 e m = 2,7)





c) Não existe par (c, m) tal que para n ≥ m, |3n2 + 5n +1| ≤ c x |n| seja verdadeira. Aumentando o valor de c, apenas retardamos o momento em que a curva quadrática supera a linear

c = 100

n	$g(n) = 3n^2 + 5n + 1$	C x f(n) = 100 x n
0	1	**0**
5	101	**500**
10	351	**1000**
15	751	**1500**
20	1301	**2000**
25	2001	**2500**
30	2851	**3000**
35	**3851**	3500
40	**5001**	4000
45	**6301**	4500
50	**7751**	5000

c = 1000

n	$g(n) = 3n^2 + 5n + 1$	C x f(n) = 1000 x n
0	1	**0**
50	7751	**50000**
100	30501	**100000**
150	68251	**150000**
200	121001	**200000**
250	188751	**250000**
300	271501	**300000**
350	**369251**	350000
400	**482001**	400000
450	**609751**	450000
500	**752501**	500000

Exercício Resolvido (13)

- Apresente a função e a ordem de complexidade para o **número de comparações de registros** no **pior** e **melhor** caso

```
void imprimirMaxMin(int [] array, int n){
        int max, min;
        if (array[0] > array[1]){
                max = array[0];
                min = array[1];
        } else {
                max = array[1];
                min = array[0];
        }
        for (int i = 2; i < n; i++){
                if (array[i] > max){
                        max = array[i];
                } else if (array[i] < min){
                        min = array[i];
}       }       }
```


a) Função de Complexidade para Comparações
   Pior: $f(n) = 1 + 2(n - 2)$
   Melhor: $f(n) = 1 + (n - 2)$

   Ordem de Complexidade para Comparações
   Ambos os casos: $O(n)$, $\Omega(n)$ e $\Theta(n)$

Exercício Resolvido (14)

- Apresente a função e a ordem de complexidade para o **número de movimentações de registros** no **pior** e **melhor** caso

```
void imprimirMaxMin(int [] array, int n){
        int max, min;
        if (array[0] > array[1]){
                max = array[0];
                min = array[1];
        } else {
                max = array[1];
                min = array[0];
        }
        for (int i = 2; i < n; i++){
                if (array[i] > max){
                        max = array[i];
                } else if (array[i] < min){
                        min = array[i];
}       }       }
```


a) Função de Complexidade para Movimentações
   Pior: $f(n) = 2 + (n - 2)$
   Melhor: $f(n) = 2 + \cancel{(n - 2) \times 0}$

Ordem de Complexidade para Movimentações
Pior: O(n), Ω(n) e Θ(n)
Melhor: O(1), Ω(1) e Θ(1)

Exercício Resolvido (15)

- Apresente a função e a ordem de 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--;
}
```

PUC Minas Virtual

a)

Função de Complexidade	
Pior	f(n) = n + 2
Melhor	f(n) = n +1

Ordem de Complexidade	
Pior	O(n), Ω(n) e Θ(n)
Melhor	

Exercício Resolvido (16)

- Apresente a função e a ordem de 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--;
}
```

PUC Minas Virtual

Função de Complexidade	
Todos os casos	$f(n) = (2n + 1)n$

Ordem de Complexidade	
Todos os casos	$O(n^2)$, $\Omega(n^2)$ e $\Theta(n^2)$

a)

Exercício Resolvido (17)

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

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


Função de Complexidade	
Todos os casos	$f(n) = (\lfloor lg(n) \rfloor + 1) \times n = n \times \lfloor lg(n) \rfloor + n$

Ordem de Complexidade	
Todos os casos	$O(n \times lg(n))$, $\Omega(n \times lg(n))$ e $\Theta(n \times lg(n))$

a)

Exercício Resolvido (18)

- 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				
$(3/2)^n$				

a)

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

Exercício Resolvido (19)

- 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 mais lento para o mais rápido (Khan Academy, adaptado)

PUC Minas Virtual

a)

RESPOSTA:

$f_6(n) = 1 \; < \; f_2(n) = n \; < \; f_1(n) = n^2 \; < \; f_5(n) = n^3 \; < \; f_4(n) = (3/2)^n \; < \; f_3(n) = 2^n$

Exercício Resolvido (20)

- 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 mais lento para o mais rápido (Khan Academy, adaptado)

a)

RESPOSTA:

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

Exercício Resolvido (21)

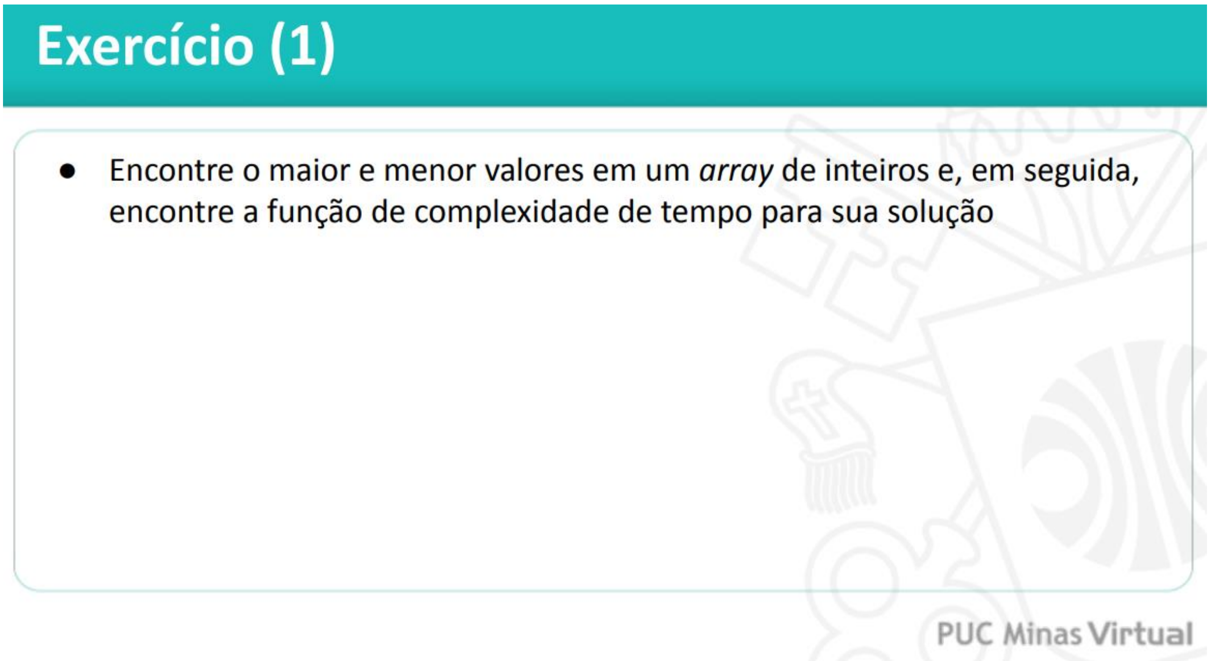
- Faça a correspondência entre cada função f(n) com sua g(n) equivalente, em termos de O. Essa correspondência acontece quando f(n) = O(g(n)) (Khan Academy, adaptado)

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

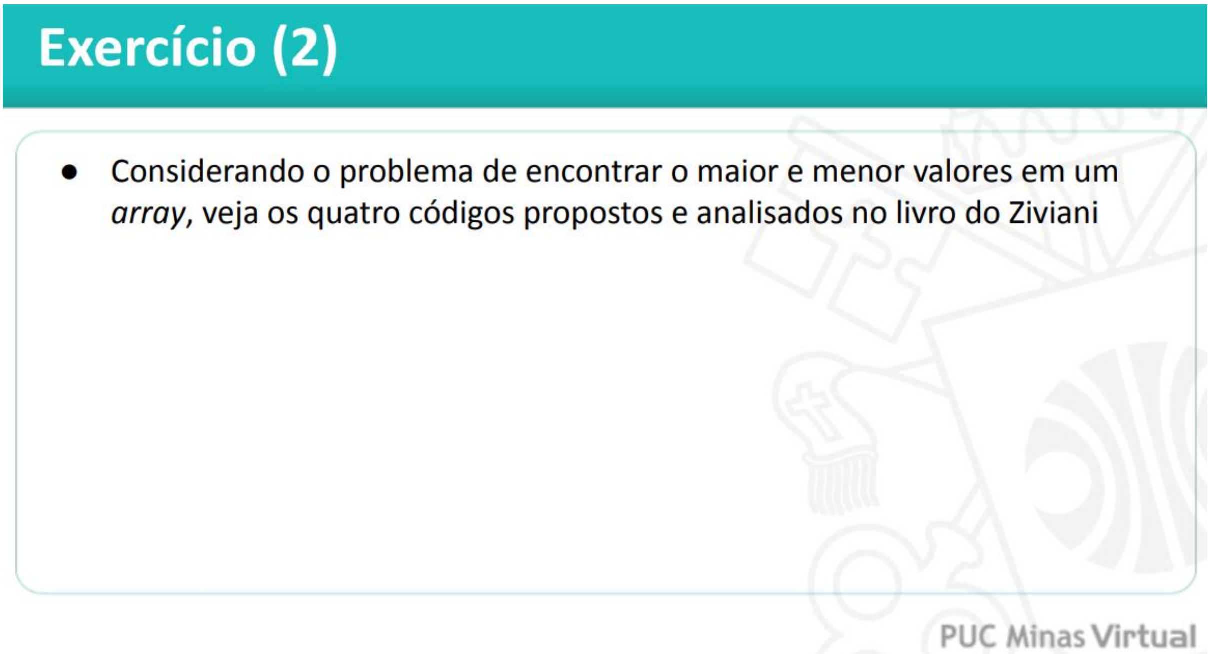
PUC Minas Virtual

R.

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



R. Para encontrar o maior e o menor valor em um array de inteiros, você pode percorrer o array uma vez enquanto mantém o controle do valor máximo e mínimo encontrados até agora.



R. Visto.

Exercício (3)

- Preencha verdadeiro ou falso na tabela abaixo:

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

PUC Minas Virtual

R.

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

Exercício (4)

- Preencha verdadeiro ou falso na tabela abaixo:

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

R.

	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 (5)

- Preencha verdadeiro ou falso na tabela abaixo:

	Ω(1)	Ω(lg n)	Ω(n)	Ω(n.lg(n))	Ω(n^2)	Ω(n^3)	Ω(n^5)	Ω(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) = $n^5 - 99999n^4$								

R.

	Ω(1)	Ω(lg n)	Ω(n)	Ω(n.lg(n))	Ω(n^2)	Ω(n^3)	Ω(n^5)	Ω(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 (6)

- Dada a definição da notação Ω:

  a) Mostre os valores de c e m tal que, para $n \geq m$, $|g(n)| \geq c \times |f(n)|$, provando que $3n^2 + 5n + 1$ é $\Omega(n^2)$

  b) Mostre os valores de c e m tal que, para $n \geq m$, $|g(n)| \geq c \times |f(n)|$, provando que $3n^2 + 5n + 1$ é $\Omega(n)$

  c) Prove que $3n^2 + 5n + 1$ **não é** $\Omega(n^3)$

a) $cxn^2 <= 3n^2+5n+1$ ->(dividindo ambos os lados por n^2) $c <= 3+((5n+1)/n^2)$, ou simplificando $c <= 3$($c=2$, $m=0$)
b) $cxn <= 3n^2+5n+1$ ->(dividindo ambos os lados por n) $c <= 3n+5+(1/n)$, ou simplificando $c > 0$($c=2$, $m=0$)
c) Não existe par (c, m) tal que para $n \geq m$, $|3n2 + 5n +1| <= c \times |n^3|$ seja verdadeira. Diminuindo o valor de c, apenas retardamos o momento em que a curva cúbica supera a quadrática

Exercício (7)

- Dada a definição da notação Θ:

  a) Mostre um valor para 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 $\Theta(n^2)$

  b) Prove que $3n^2 + 5n +1$ **não** é $\Theta(n)$

  c) Prove que $3n^2 + 5n +1$ **não** é $\Theta(n^3)$

a) $c1xn^2 <= 3n^2+5n+1 <= c2xn^2$ ->(dividindo ambos os lados por n^2) $c1 <= 3+((5n+1)/n^2) <= c2$, ou simplificando $c1<3$ e $c2>3$
b) $c1xn <= 3n^2+5n+1 <= c2xn$ ->(dividindo ambos os lados por n) $c1 <= 3n+5+(1/n) <= c2$, é impossível ter um c2 maior que $3n+5+(1/n)$, portanto é falso

c) $c_1 x n^3 <= 3n^2+5n+1 <= c_2 x n^3$ ->(dividindo ambos os lados por n^3) $c_1 <= ((3n^2+5n+1)/n^3) <= c_2$, é impossível ter um c1 menor que $(3n^2+5n+1)/n^3$, portanto é falso

Exercício (8)

- 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.

```
void sistemaMonitoramento() {
        alarme(((telefone() == true && luz() == true)) ? 0 : 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);
} } }
```

PUC Minas Virtual

a. Pior caso: 1+(n-2)
   Melhor caso: 1
b. .
   i. Telefone: 1
   ii. Luz: 1
   iii. Sensor: Melhor caso n-2, pior caso n-2
   iv. Câmera: Melhor caso 0, pior caso n-2

Exercício (9)

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

PUC Minas Virtual

a.
```
int j = 0;
```

```java
    int l = 0;
    int m = 0;
    for (int i = 0; i < 10; i++){
        System.out.println(i);
        if(i % 2 == 1){
            j++;
        }else{
            m++;
            l++;
        }
    }
}
```

Exercício (10)

- Anteriormente, 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 $\Theta(n)$. Nesse caso, o custo de ordenar o *array* e, em seguida, aplicar uma pesquisa binária é mais elevado, $\Theta(n \times \lg(n)) + \Theta(\lg(n)) = \Theta(n \times \lg(n))$. Agora, supondo que desejamos efetuar ***n*** pesquisas, responda qual das duas soluções é mais eficiente

PUC Minas Virtual

R. Usar apenas a pesquisa padrão, pois em quantidades muito altas, esta se torna mais eficiente.