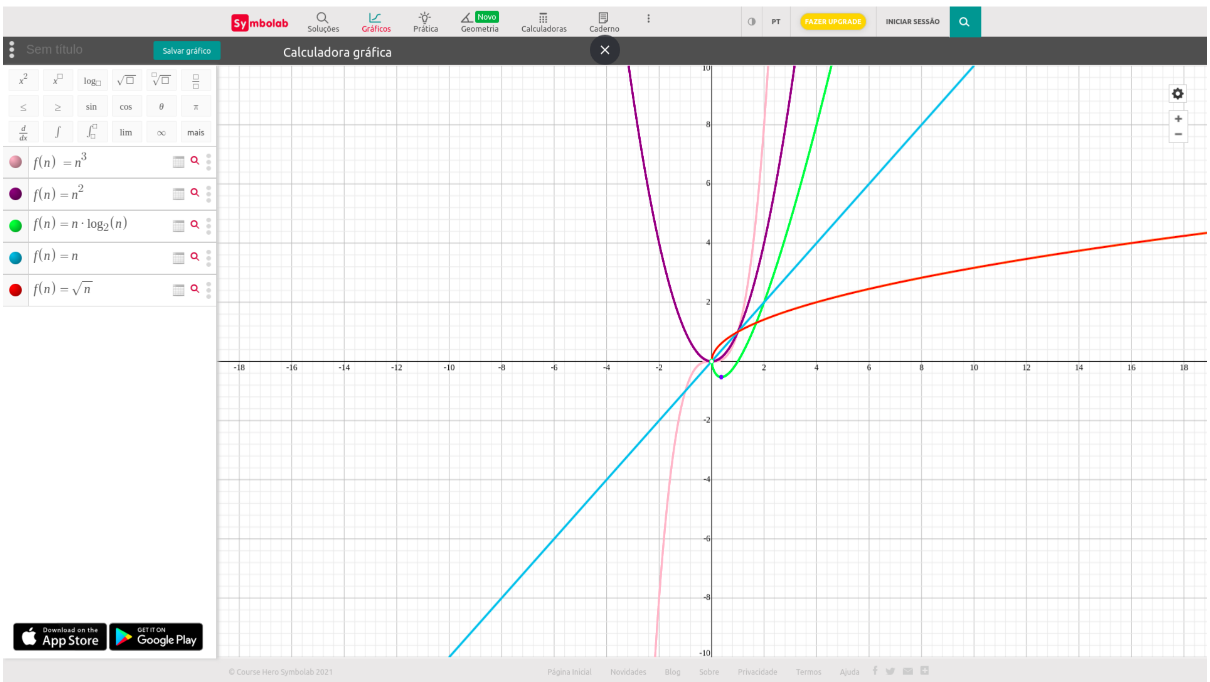
Exercícios da Unidade III: Fundamentos de Análise de Algoritmos

Nome: Bárbara Maria Sampaio Portes

- Exercício Resolvido (1)
  a) 210 = 1024
  b) lg(1024) = 10
  c) lg(17) = 4,08746284125034
  d) (teto) lg(17) = 5
  e) (piso) lg(17) = 4
- Exercício Resolvido (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)



- Exercício Resolvido (3)
  Melhor caso: f(n) = n, logo, O(n), Ω(n) e Θ(n)
  Pior caso: f(n) = 2n, logo, O(n), Ω(n) e Θ(n)
- Exercício Resolvido (4)
  n - 3 subtrações
  Logo, O(n), Ω(n) e Θ(n)
- Exercício Resolvido (5)

Para um valor qualquer de n,
temos lg(n) + 1 multiplicações,
logo, O(lg n), Ω(lg n) e Θ(lg n)

- Exercício Resolvido (6)

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

- Exercício Resolvido (7)

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

- Exercício Resolvido (8)

1º) Qual é a operação relevante?
R: Comparação entre elementos do array
2º) Quantas vezes ela será executada?
R: Se tivermos n elementos: T(n) = n - 1
3º) O nosso T(n) = n – 1 é para qual dos três
casos?
R: Para os três casos
4º) O nosso algoritmo é ótimo? Por que?
R: Sim porque temos que testar
todos os elementos para garantir
nossa resposta

- Exercício Resolvido (9)
  boolean resp = false;
  for (int i = 0; i < n; i++){
  if (array[i] == x){
  resp = true;
  i = n;
  }
  }
- Exercício (1)

```java
// funcao para encontrar o menor valor de um array
// retorna o menor valor
public static int minimo(int array[]) {
    int n = array.length;
    int min = array[0];
    for (int i = 1; i < n; i++) {
        if (min > array[i]) {
            min = array[i];
        }
    }
    return min;
}
// funcao para encontrar o maior valor de um array
// retorna o maior valor
public static int maximo(int array[]) {
    int n = array.length;
    int max = array[0];
    for (int i = 1; i < n; i++) {
        if (max < array[i]) {
            max = array[i];
        }
    }
    return max;
}
```

- Exercício (2)
  Feito.
- Exercício Resolvido (10)
  O aluno deve escolher a primeira opção, pois a pesquisa sequencial tem custo $\Theta(n)$. A segunda opção tem custo $\Theta(n * \lg n)$ para ordenar mais $\Theta(\lg n)$ para a pesquisa binária

- Exercício Resolvido (11)
  a) 3n2 + 5n + 1 é O(n): falsa
  b) 3n2 + 5n + 1 é O(n2): verdadeira
  c) 3n2 + 5n + 1 é O(n3): verdadeira
  d) 3n2 + 5n + 1 é Ω(n): verdadeira
  e) 3n2 + 5n + 1 é Ω(n2): verdadeira
  f) 3n2 + 5n + 1 é Ω(n3): falsa
  g) 3n2 + 5n + 1 é Θ(n): falsa
  h) 3n2 + 5n + 1 é Θ(n2): verdadeira
  i) 3n2 + 5n + 1 é Θ(n3): falsa
- Exercício (3)

	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	V	V	V	V	V	V	V
f(n) = n . lg(n)	F	F	F	V	V	V	V	V
f(n) = 5n + 1	F	F	V	V	V	V	V	V
$f(n) = 7n^5 - 3n^2$	F	F	F	F	F	F	V	V
$f(n) = 99n^3 - 1000n^2$	F	F	F	F	F	V	V	V
$f(n) = n^5 - 99999n^4$	F	F	F	F	F	F	V	V

- Exercício (4)

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

- Exercício (5)

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

- Exercício Resolvido (12)
  Neste caso, executamos o Seleção n vezes: n x Θ(n2) = Θ(n3)
- Exercício Resolvido (13)
  Neste caso, temos duas etapas e o custo total será a soma das mesmas, logo: Θ(n.lg n) + Θ(lg n) = Θ(n.lg n)
- Exercício Resolvido (14)
  a)h(n) + g(n) - f(n) ⇒ [99n8] + [n.lg(n)] - [3n2-5n-9] ⇒ O(n8), Ω(n8) e Θ(n8)
  b)Θ(h(n)) + Θ(g(n)) - Θ(f(n)) ⇒ Θ(n8) + Θ(n.lg(n)) - Θ(n2) ⇒ O(n8), Ω(n8) e Θ(n8)
  c)f(n) x g(n) ⇒ Θ(n2) x Θ(n.lg(n)) ⇒ O(n3.lg(n)), Ω(n3.lg(n)) e Θ(n3.lg(n))
  d)g(n) x l(n) + h(n) ⇒ Θ(n.lg(n)) x Θ(n.lg2(n)) + Θ(n8) ⇒ O(n8), Ω(n8) e Θ(n8)
  e)f(n) x g(n) x l(n) ⇒ Θ(n2) x Θ(n.lg(n)) x Θ(n.lg2(n)) ⇒ O(n4.lg3(n)), Ω(n4.lg3(n)) e Θ(n4.lg3(n))
  f) Θ(Θ(Θ(Θ(f(n))))) ⇒ O(n2), Ω(n2) e Θ(n2)
- Exercício Resolvido (15)
- Exercício (6)
- Exercício (7)
- Exercício (8)
  Em anexo.

- Exercício Resolvido (16)

função de complexidade

	MOV	CMP
PIOR	$f(n) = 2 + (n - 2)$	$f(n) = 1 + 2(n - 2)$
MELHOR	$f(n) = 2 + \cancel{(n - 2) \times 0}$	$f(n) = 1 + (n - 2)$

complexidade

	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 $\Theta(n)$

- Exercício Resolvido (17)
  PIOR - $f(n) = n + 2$ / $O(n)$, $\Omega(n)$ e $\Theta(n)$
  MELHOR - $f(n) = n + 1$ / $O(n)$, $\Omega(n)$ e $\Theta(n)$
- Exercício Resolvido (18)
  $f(n) = (2n + 1)n$ - $O(n2)$, $\Omega(n2)$ e $\Theta(n2)$
- Exercício Resolvido (19)

$f(n) = (\lg(n) + 1) * n = n * \lg(n) + n$ / $O(n \times \lg(n))$, $\Omega(n \times \lg(n))$ e $\Theta(n \times \lg(n))$

- Exercício (9)
- Exercício (10)
- Exercício Resolvido (20)

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

- Exercício Resolvido (21)
  f6(n) = 1
  f2(n) = n
  f1(n) = n^2
  f5(n) = n^3
  f4(n) = (3/2)^n
  f3(n) = 2^n
- Exercício Resolvido (22)
  f6(n) = 64
  f3(n) = log8(n)
  f2(n) = lg(n)
  f9(n) = 4n
  f1(n) = n.log6(n)
  f5(n) = n.lg(n)
  f4(n) = 8n^2
  f7(n) = 6n^3
  f8(n) = 8^2n
- Exercício Resolvido (23)

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

- Exercício (11)