

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA <i>CAMPUS</i> CAMPINA GRANDE	
Curso: Engenharia de Computação	Disciplina: Análise e Técnicas de Algoritmos	
Professor: Emanuel Dantas Filho		
Aluno: Pedro Macêdo Luna		

Atividade 04

01) Pesquise exemplos de algoritmos que tem complexidade assintótica igual a:

a) $O(1)$

```
n = int(input())
if n > 0:
    print("Valor positivo")
elif n < 0:
    print("Valor negativo")
else:
    print("Valor nulo")
```

b) $O(n)$

```
n = int(input())
for i in range(n):
    print(i)
```

c) $O(\log n)$

```
def busca_binaria(array, alvo):
    menor = 0
    maior = len(array) - 1
    while menor <= maior:
        medio = (menor + maior) // 2
        if array[medio] == alvo:
            return alvo
        elif array[medio] < alvo:
            maior = medio + 1
        else:
            menor = medio - 1
```

d) O (n*n)

```
def matriz_quadrada(tamanho):  
    matriz = []  
    for i in range(tamanho):  
        matriz.append([])  
        for j in range(tamanho):  
            matriz[i].append([ i,j ])  
    return matriz
```

02) Escreva a função matemática correspondente e a complexidade assintótica dos algoritmos a seguir:

```
matrizQuadratica (int X , int Z){  
    int valor[ ][ ];  
    for (a: 1 até X):  
        for (b: 1 até Z)  
            valor[a][b] = X*Z  
    If (X > Z)  
        return 0;  
    else  
        return 1;  
}
```

$\Rightarrow O(1)$
 $\Rightarrow O(n)$
 $\Rightarrow O(n)$
 $\Rightarrow O(1)$
 $\Rightarrow O(1)$

$$f(n) = O(1) + O(n) * O(n) + O(1) + O(1) = O(n^2) + 3*O(1)$$

1º Passo - Ignorar as constantes:

$$\Rightarrow f(n) = O(n^2) + 3 * O(1)$$

$$\Rightarrow f(n) = O(n^2) + O(1)$$

2º Passo - Ignorar os expoentes de menor magnitude:

$$\Rightarrow f(n) = O(n^2) + \cancel{O(1)}$$

$$\Rightarrow f(n) = O(n^2)$$

Portanto, a complexidade assintótica do algoritmo “matrizQuadratica” é $O(n^2)$, ou seja, quadrática .

```
calculaIMC (double peso, double altura){  
    double IMC = 0;  
    if (peso > 0 AND altura > 0)  
        return peso/ altura*altura;  
    else  
        return 0;  
}
```

$\Rightarrow O(1)$
 $\Rightarrow O(1)$

$$f(n) = O(1) + O(1) + O(1) = 3 * O(1)$$

1º Passo - Ignorar as constantes:

$$\Rightarrow f(n) = 3 * O(1)$$

$$\Rightarrow f(n) = O(1)$$

Portanto, a complexidade assintótica do algoritmo “calculaIMC” é $O(1)$, ou seja, constante.

```
visualizarTabuada (int valor){
    for (i: 1 até valor)            $\Rightarrow O(n)$ 
        print i*valor;
}
```

$$f(n) = O(n)$$

Portanto, a complexidade assintótica do algoritmo “visualizarTabuada” é $O(n)$, ou seja, Linear.

03) Considere as seguintes funções e coloque as funções em ordem de crescimento assintótico.

$O(\log n)$, $O(2^{**}n)$, $O(n \log n)$, $O(n*n)$, $O(2*n)$

1. $O(\log n)$
2. $O(2^{**}n)$
3. $O(n \log n)$
4. $O(n*n)$
5. $O(2*n)$

04) Simplifique as expressões e defina a complexidade assintótica:

a) $500 + n*n + 2^{**}n + 1000*n$

1º Passo - Ignorar as constantes:

$$\Rightarrow \cancel{500} + n * n + \cancel{2^{**}} n + \cancel{1000} * n$$

$$\Rightarrow n^2 + n + n$$

$$\Rightarrow n^2 + 2 * n$$

$$\Rightarrow n^2 + n$$

2º Passo - Ignorar os expoentes de menor magnitude:

$$\Rightarrow n^2 + \cancel{n}$$

Portanto, a complexidade assintótica da expressão é $O(n^2)$, ou seja,

quadrática .

b) $\text{Log}(n) + n^2 + 100n + 10\text{Log}(n)$

1º Passo - Ignorar as constantes:

⇒ **$\text{Log}(n) + n^2 + \cancel{100} * n + \cancel{10} \text{Log}(n)$**

⇒ **$\text{Log}(n) + n^2 + n + \text{Log}(n)$**

⇒ **$2 * \text{Log}(n) + n^2 + n$**

⇒ **$\text{Log}(n) + n^2 + n$**

2º Passo - Ignorar os expoentes de menor magnitude:

⇒ **$\cancel{\text{Log}(n)} + n^2 + n$**

Portanto, a complexidade assintótica da expressão é $O(n^2)$, ou seja, quadrática