Complexidade de Algoritmos I – 2022 - ATIVIDADE 4

Nome:_____ RA: _____

1) Coloque as principais classes de problemas listadas a seguir em ordem crescente.

$$O(n!),\ O(n),\ O(n^3),\ O(1),\ O(2^n),\ O(n\log n),\ O(n^2),\ O(\log n)$$

Resposta: $O(1) < O(\log n) < O(n) < O(n\log n) < O(n^2) < O(n^3) < O(2^n) < O(n!)$

2) Para cada um dos trechos de código abaixo, analise o tempo estimado de execução no melhor e no pior caso. Apresente a função de tempo em relação ao número de instruções executadas. Considere que as variáveis **n**, **m** e **vetor** sejam dados como entrada.

a)

```
int soma = 0;
for(int i=0; i<n; i++)
    soma = soma + i;
```

Pior caso: $T(n) = n + 1$

Melhor caso: $T(n) = n + 1$

b)

```
int soma1 = 0;
int soma2 = 0;
for(int i=0; i<n; i++){
    soma1 = soma1 + i;
    soma2 = soma2 + i;
}
```

Pior caso: $T(n) = 2n + 2$

Melhor caso: $T(n) = 2n + 2$

c)

```
int soma = 0;
for(int i=0; i<n; i++){
    if(vetor[i]%2==0)
        soma = soma + vetor[i];
    }
}
```

Pior caso: $T(n) = 2n + 1$

Melhor caso: $T(n) = n + 1$

d)

```
int soma = 0;
for(int i=0; i<n; i++){
    for(int j=0; j<m; j++){
    soma = soma + 1;
    }
}
```

Pior caso: $T(n) = n.m + 1$

Melhor caso: $T(n) = n.m + 1$

e)

```
int menor = MAIOR_INTEIRO;
for(int i=0; i<n; i++){
    if(vetor[i]<menor)
        menor=vetor[i];
}
if(menor<0){
    for(int i=0; i<n; i++)
        menor=menor*(i+1);
}else{
    if(menor>0){
        for(int i=0; i<n*n; i++)
            printf("%d\n", menor);
    }
    else
        printf("%d\n",menor);
}
```

Pior caso: $T(n) = n^2 + 2n + 3$

Melhor caso: $T(n) = n + 4$

3) Analise novamente os algoritmos do exercício anterior, juntamente com as funções de tempo calculadas para os piores e melhores casos, e apresente em notação assintótica o menor limite superior (notação O) e o maior limite inferior (notação Ω).
   a) $O(n)$ e $\Omega(n)$
   b) $O(n)$ e $\Omega(n)$
   c) $O(n)$ e $\Omega(n)$
   d) $O(n.m)$ e $\Omega(n.m)$
   e) $O(n^2)$ e $\Omega(n)$

4) Analise o algoritmo abaixo e identifique o seu pior e seu melhor caso utilizando a notação assintótica. Explique.

```
exibe_matriz_3D(M)
    for i ← 1 to comprimento_x[M]
        for j ← 1 to comprimento_y[M]
            for k ← 1 to comprimento_z[M]
                do escreva(M[i][j][k]))
```

Pior caso: $T(n) = comprimento\_x * comprimento\_y * comprimento\_z$

Melhor caso: $T(n) = comprimento\_x * comprimento\_y * comprimento\_z$

$O(comp\_x * comp\_y * comp\_z)$ e $\Omega(comp\_x * comp\_y * comp\_z)$

5) Apresente uma análise da complexidade do subprograma apresentado abaixo, apresentando o seu pior e o seu melhor caso utilizando a notação assintótica. Note que ***pessoas*** é uma lista e ***size()*** é um método que retorna o número de elementos dessa lista.

```
Pessao busca(String nome){
    for(int i=0; i<pessoas.size(); i++){
        if(pessoas.get(i).getNome().equals(nome))
            return pessoas.get(i);
    }
    return null;
}
```

Considerando que o método ***size()*** retorne uma variável **n**:

Pior caso: $T(n) = n + 1$

Melhor caso: $T(n) = 3$

$O(n)$ e $\Omega(1)$