Complexidade de Algoritmos I – 2022 - ATIVIDADE 4

Nome: Vinicius Mesquini de Oliveira RA: 009319

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

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

R: O(1), O(log n), O(n), O(n log n), O(n²), O(n³), 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, considerando o modelo RAM. 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;                          1
for(int i=0; i<n; i++)                 n      Ω(1+1*n)  Ω(n)
    soma = soma + i;                   1      O(1+1*n)  O(n)
```

b)

```
int soma1 = 0;                         1
int soma2 = 0;                         1
for(int i=0; i<n; i++){                n      Ω(2+2*n)  Ω(n)
    soma1 = soma1 + i;                 1      O(2+2*n)  O(n)
    soma2 = soma2 + i;                 1
}
```

c)

```
int soma = 0;                          1
for(int i=0; i<n; i++){                n
    if(vetor[i]%2==0)                  1      Ω(1+1*n)  Ω(n)
        soma = soma + vetor[i];        1      O(1+2*n)  O(n)
    }
}
```

d)

```
int soma = 0;                          1
for(int i=0; i<n; i++){                n
    for(int j=0; j<m; j++){            n      Ω(1+1*n*n)  Ω(n²)
    soma = soma + 1;                   1      O(1+1*n*n)  O(n²)
    }
}
```

e)

```
int menor = MAIOR_INTEIRO;                  1
for(int i=0; i<n; i++){                     n
    if(vetor[i]<menor)                      1
        menor=vetor[i];                     1
}                                           1
if(menor<0){                                n
    for(int i=0; i<n; i++)                  1
        menor=menor*(i+1);
}else{                                      1
    if(menor>0){                            1
        for(int i=0; i<n*n; i++)            n²
            printf("%d\n", menor);          1
    }
    else                                    1
        printf("%d\n",menor);               1
}
```

Ω(1+1*n+1+1+1) Ω(n)
O(1+2*n+1+1+1*n²) O(n²)

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).

Ω(n)
O(n²)

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])
```

Ω(m*m*m*1) Ω(1*m³)
O(m*m*m*1) O(1*m³)

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++){        n
        if(pessoas.get(i).getNome().equals(nome))   1
            return pessoas.get(i);              1
    }
    return null;                                1
}
```

Ω(1*n+1) Ω(n)
O(2*n) O(n)