

Complexidade de Algoritmos I – 2022 - ATIVIDADE 4

Nome: Jamir Sui Jun RA: 1082/19

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

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

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

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

Handwritten: $\sim O(1+n) = O(n)$
 $\sim O(1+n) = O(n)$

b)

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

Handwritten: $\sim O(1+n) = O(n)$
 $\sim O(1+n) = O(n)$

c)

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

Handwritten: $\sim O(1+n) = O(n)$
 $\sim O(1+n) = O(n)$

d)

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

Handwritten: $\sim O(1+n \cdot m) = O(n \cdot m)$
 $\sim O(1+n \cdot m) = O(n \cdot m)$

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);
}
```

$2C - 1144$

$O = 42n + n^2$

- 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).

$\Omega(1) O(n)$

- 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]))
```

$\Omega = m^3$ $O = m^3$

- 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 *peessoas* é uma lista e *size()* é um método que retorna o número de elementos dessa lista.

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

$\Omega(1)$

$O(n)$