Complexidade de Algoritmos I – 2022 - ATIVIDADE 3

Nome: VINICIUS MESQUINI DE OLIVEIRA RA: 009319

1) Sejam $T1(n) = 3n + 3n\log_2 n + 25\log_3 n$, $T2(n) = 15n + 3n^2 + 9n^2\log_2 n + 8$ e $T3(n) = 5n^3 + 7n^2 + 2$, apresente as equações que descrevem a ordem de complexidade de tempo dos algoritmos Alg1, Alg2 e Alg3, respectivamente, para entradas de tamanho n.

Alg1: n log n
Alg2: n2 log n
Alg3: n3

2) Um método de ordenação de complexidade $O(\log n)$ gasta exatamente 2 milissegundos para ordenar 10000 elementos. Supondo que o tempo $T(n)$ para ordenar n desses elementos é diretamente proporcional a $\log n$, ou seja, $T(n) = c.\log n$:

a) Estime a constante c utilizando uma base conveniente para o logaritmo.

t(10000) = c.log n

2 = c.log (10000)

2 = c. 4

c = 2/4

c = 1/2 = 0.5

b) Estime o tempo consumido por esse algoritmo, em segundos, para ordenar 1000000 elementos.

t(1000000) = 1/2 * log(1000000)

t(1000000) = 1/2 * 6

t(1000000) = 3 ms

3) Suponha que cada expressão abaixo represente o tempo $T(n)$ consumido por um algoritmo para resolver um problema de tamanho n. Escreva os termos(s) dominante(s) para valores muito grandes de n e especifique o menor limite assintótico superior $O(n)$ possível para cada algoritmo.

Expressão	Termo(s) Dominante(s)	$O(\ldots)$
$5 + 0.01n^2 + 0.52n^4$	0.52n^4	O(n^4)
$100n + 0.01n^3$	$0.01n^3$	$O(n^3)$
$5n^2 + 10n^{1.5} + 5n$	$5n^2$	$O(n^2)$
$13n + 4n^2$	$4n^2$	$O(n^2)$
$0.3n + 5n^{1.5} + 2.5n^{1.75}$	2.5n^1.75	O(n^1.75)
$n^3\log_2(n) + 5n(\log_3(n))^2$	$n^3 \log n$	$O(n^3 \log n)$
$2n + n^{1.5} + 0.5n^2$	$0.5n^2$	$O(n^2)$
$n^2\log_3(n) + n^2\log_2(n)$	$n^2 \log n$	$O(n^2 \log n)$
$5n^2\log_2(n) + 2n^3 + 10n$	$2n^3$	$O(n^3)$
$5n^2 + n^3\log n$	$n^3 \log n$	$O(n^3 \log n)$

4) Analise o algoritmo abaixo, escrito em C, que recebe dois vetores, a e b, de tamanhos iguais n e determine o menor limite assintótico superior para o pior caso em função do parâmetro n.

```
float fc(float *v1, float *v2, int n, int op){
    int i=0;
    float r = 0;
    if(op ==1){
        for(;i<n;i++)
            r += v1[i] + v2[i];
    }
    else{
        r = 1;
        for(;i<n;i++)
            for(int j=i;j<n;j++)
                r *= v1[i]*v2[j];
    }
}
```

```
else{
        r = 1
        for(n){
                for(n){
                }
        }
}
```

n * n

n^2

5) Encontre o menor limite assintótico superior para o algoritmo abaixo, escrito C:

```
int menor(int vetor[], int n){
  int menor = MAX_INT;
  para i=1 ate n faça
    se (vetor[i] < menor)
      menor = vetor[i];
  se menor < 0
    para i=1 ate n faca
      para j=1 ate n faca
        vetor[i] = vetor[i]^(i+j);
  retorna(menor);
}
```

```
n {
  1   —   } n
}

se menor < 0
  n{
    n{
      1  —   } n²
    }
  }

retorna ;
```

$\Omega(n)$

$O(n^2)$

6) Suponha que ofereçam a você dois pacotes de software, **A** e **B**, para processamento dos dados da sua empresa, que contêm 10^6 registros. Sabendo que o tempo de processamento médio do pacote **A** é $T_A(n) = 2n^2$ milissegundos, e o tempo médio de **B** é $T_B(n) = 1000n$ milissegundos, responda:

a) Qual desses pacotes é o mais indicado para processar os dados da empresa?

b) A partir de quantos registros um dos pacotes passa a ser melhor que o outro?

A:

Calculo Ta(n) = 2n² :
Ta(10^6) = 2(10^6)²
Ta(10^6) = 2*10^12 ms
Ta(10^6) = 2*10^9 seg

Calculo Tb(n) = 1000n :
Tb(10^6) = 1000*10^6
Tb(10^6) = 100*10^7
Tb(10^6) = 10*10^8
Tb(10^6) = 1*10^9
Tb(10^6) = 10^9 ms
Tb(10^6) = 10^6 seg

Ta(10^6) > Tb(10^6),
o pacote B é melhor para empresa

B:

Ta(n) = Tb(n)

2.n² = 1000.n = 2n² - 1000n = 0

$$n = \frac{-b \pm \text{raiz}(b^2 - 4ac)}{2a}$$

$$n = \frac{1000 \pm \text{raiz}(1000^2 - 4 * 2 * 0)}{2 * 2}$$

$$n = \frac{1000 \pm 1000}{4} = \quad n1 = \frac{2000}{4} = 500 \qquad n2 = \frac{0}{4} = 0$$

Como valor 0 não corresponde, sobra o valor 500, podemos afirmar que o pacote B passa a ser melhor a partir de n = 500