	CURSOS: CIÊNCIA DA COMPUTAÇÃO **1º Bimestre / 1º Semestre de 2022** **DISCIPLINAS:** Complexidade de Algoritmos I	**NOTA:**
Aluno(a):		**R.A.:**
Prof. Dr. Guilherme Pina Cardim		**31 / 03 / 2022**

1) Sejam $T1(n) = 10n + 3\log_2 n + 25$, $T2(n) = 3n + 5n\log_3 n + 15n^2 + 8$ e $T3(n) = 15n^3 + 2n^2 + 9$ as equações que descrevem a complexidade de tempo dos algoritmos Alg1, Alg2 e Alg3, respectivamente, para entradas de tamanho n. A respeito da ordem de complexidade desses algoritmos, pode-se concluir que (0,5 pontos):
   a) As complexidades assintóticas de Alg1, Alg2 e Alg3 estão, respectivamente, em $O(n)$, $O(n\log n)$ e $O(n^3)$.
   b) As complexidades assintóticas de Alg1, Alg2 e Alg3 estão, respectivamente, em $O(n)$, $O(n^2)$ e $O(n^3)$.
   c) As complexidades assintóticas de Alg1, Alg2 e Alg3 estão, respectivamente, em $O(\log n)$, $O(n\log n)$ e $O(n^3)$.
   d) As complexidades assintóticas de Alg1, Alg2 e Alg3 estão, respectivamente, em $O(n)$, $O(n\log n)$ e $O(n^3)$.
   e) Alg1, Alg2 e Alg3 pertencem a mesma classe de complexidade assintótica.

Resposta: b

2) Um algoritmo tem complexidade $O(7m + 3n^3 + n^2 + \log_2 m + 3m^2 + \log_3 n)$. Contudo, podemos simplificar a complexidade apresentada para (0,5 pontos):
   a) $O(m^2 + n^3)$
   b) $O(n^3)$
   c) $O(\log m + \log n)$
   d) $O(m^2 + n^2)$
   e) $O(m^3 + n^2)$

   Resposta: a

3) Analise a afirmação abaixo e diga se ela é sempre verdadeira, sempre falsa ou se depende da situação. Para tanto, considere a função f assintoticamente não negativas. Explique sua resposta. (1,0 pontos)

$$f(n) = O(f(n)^3)$$

Resposta: É sempre verdadeira, pois o tempo para executar $f(n)$ sempre será menor do que $O(f(n)^3)$.

Como $f(n) \leq f(n)^3$ a afirmação é sempre verdadeira.

4) Um método de ordenação possui complexidade $O(n^2 \log n)$ e gasta exatamente 10 milissegundos para ordenar 100 elementos. Supondo que o tempo $T(n)$ para ordenar n desses elementos é diretamente proporcional a $n^2 \log n$, ou seja, $T(n) = c.n^2 \log n$: (1,5 pontos)

a) Estime a constante c utilizando uma base conveniente para o logaritmo (0,75 pontos).
Resposta:

$$T(n) = c.n^2\, log\, n \;\; e \;\; T(n) = 10ms\ para\ n = 100,\ portanto:$$
$$10 = c.100^2\, log_{10} 100,\ como\ log_{10} 100 = 2$$
$$10 = 20000.c$$
$$c = \frac{1}{2000}$$

b) Estime o tempo consumido por esse algoritmo, em segundos, para ordenar 1000 elementos (0,75 pontos).
Resposta:

$$\text{Considerando } c = \frac{1}{2000},\ n = 1000 \text{ e } T(n) = c.n^2\, log\, n$$
$$T(1000) = \frac{1}{2000}.1000^2 \log_{10} 1000,\ como\ log_{10} 1000 = 3$$
$$T(1000) = \frac{1}{2000}.1000.1000.3$$
$$T(1000) = \frac{3000}{2} = 1500\ milisegundos = 1{,}5\ segundos$$

5) 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. (2,0 pontos)

Expressão	Termo(s) Dominante(s)	$O(\ldots)$
$5 + 0.001n^3 + 0.025n$	$0.001n^3$	$O(n^3)$
$100n + 0.01n^2$	$0.01n^2$	$O(n^2)$
$500n + 100n^{1.5} + 2.5$	$100n^{1.5}$	$O(n^{1.5})$
$0.01n + 100n^2$	$100n^2$	$O(n^2)$
$0.3n + 5n^{1.5} + 2.5n^{1.75}$	$2.5n^{1.75}$	$O(n^{1.75})$
$n^2 \log_2(n) + n(\log_2(n))^2$	$n^2 \log_2(n)$	$O(n^2 \log n)$
$2n + n^{0.5} + 0.5$	$2n$	$O(n)$
$n \log_3(n) + n \log_2(n)$	$n \log_3(n)$ / $n \log_2(n)$	$O(n \log n)$
$100n \log_2(n) + n^3 + 100n$	n^3	$O(n^3)$
$5n^3 + n^2 \log n$	$5n^3$	$O(n^3)$

6) Analise o algoritmo abaixo, escrito em C, que recebe um vetor, v, de tamanho igual a n e determine o menor limite assintótico superior para o pior caso em função do parâmetro n. Explique como obteve sua resposta. (1,0 ponto)

```
double funcao(int * v, int n){
    int i, op;
    double res = 1;
    for(i=0;i<n;i++){
        op = *(v+i) %3;
        switch(op){
            case 1:
                for(j=n-1; j>=0; j--)
                    res *= v[j];
                break;
            case 2:
                for(j=n; j>0; j/=2)
                    res /= v[j];
                break;
            case 3:
                for(j=0;j<n*n;j++){
                    if(j<n)
                        res += v[j];
                    else
                        res += v[(j*j)-n];
                }
            default:
                return -1;
        }
    }
    return res;
}
```

$O(n)$

$O(n)$

$O(\log n)$

$O(n^2)$

$O(1)$

$O(n).O(n^2) = O(n^3)$

7) Suponha que ofereçam a você dois pacotes de software, **A** e **B**, para processamento dos dados da sua empresa, que contêm 10^7 registros. Sabendo que o tempo de processamento médio do pacote **A** é $T_A(n) = 500n$ milissegundos, e o tempo médio de **B** é $T_B(n) = 2n^2$ milissegundos, responda: (1,5 pontos)

a) Qual desses pacotes é o mais indicado para processar os dados da empresa? (0,75 pontos)

Resposta:

Calculando $T_A(n)$, com $n = 10^7$

Como $T_A(n) = 500n$, $T_A(10^7) = 500.10^7\ milissegundos$

$$T_A(10^7) = 50.10^8\ milissegundos$$

$$T_A(10^7) = 5.10^9\ milissegundos = 5.10^6 segundos$$

Calculando $T_B(n)$, com $n = 10^7$

Como $T_B(n) = 2n^2$, $T_B(10^7) = 2.(10^7)^2 = 2.10^{14}\ milissegundos$

$$T_B(10^7) = 2.10^{11} segundos$$

Como $T_A(10^7) < T_B(10^7)$, o pacote A é o mais apropriado para a empresa.

b) A partir de quantos registros um dos pacotes passa a ser melhor que o outro? (0,75 pontos)

Resposta:

Nesse caso precisamos encontrar o valor de n, tal que $T_A(n) = T_B(n)$, logo:

$$T_A(n) = T_B(n)$$
$$500.n = 2.n^2$$
$$2n^2 - 500n = 0$$
$$n = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{500 \pm \sqrt{500^2 - 4.2.0}}{2.2} = \frac{500 \pm 500}{4}$$
$$n_1 = \frac{1000}{4} = 250 \text{ e } n_2 = \frac{0}{4} = 0$$

Como o valor 0 não responde a nossa questão, nos sobra apenas o valor 250. Além disso, já vimos que para um valor de n maior do que 250, o pacote A se torna mais eficiente do que o pacote B, então podemos afirmar que o pacote A se torna mais eficiente do que o pacote B a partir do valor 250 para n.