

- I.** Suponha que cada expressão abaixo represente o tempo  $T(n)$  consumido por um algoritmo para resolver um problema de tamanho  $n$ . Escreva os termo(s) dominante(s) para valores muito grandes de  $n$  e especifique o tempo de execução em notação  $\Theta$

Expressão	Termo(s) Dominante(s)	$\Theta$ (...)
$5 + 0.001n^3 + 0.025n$		
$500n + 100n^{1.5} + 50n\log_{10}(n)$		
$0.3n + 5n^{1.5} + 2.5n^{1.75}$		
$n^2\log_2(n) + n(\log_2(n))^2$		
$n\log_3(n) + n\log_2(n)$		
$3\log_8(n) + \log_2(\log_2(\log_2(n)))$		
$100n + 0.01n^2$		
$0.01n + 100n^2$		
$2n + n^{0.5} + 0.5n^{1,25}$		
$0.01n\log_2(n) + n(\log_2(n))^2$		
$100n\log_3(n) + n^3 + 100n$		
$0.003\log_4(n) + \log_2(\log_2(n))$		

2. Analise o algoritmo abaixo, escrito em C, que recebe dois vetores,  $a$  e  $b$ , de tamanhos iguais  $n$ . Determine:

```
float f(float* a, float* b, int n) {  
    int i, j;  
    float s = 0.0;  
    for (i=1; i<n; i++) {  
        if (a[i]>600) {  
            for (j=n-1; j>=0; j--) {  
                s += a[i]*b[j];  
            }  
        } else if (a[i]<300) {  
            for (j=n; j<n*n; j+=5) {  
                s += a[i]*b[j];  
            }  
        } else {  
            for (j=1; j<n; j=3*j) {  
                s += a[i]*b[j];  
            }  
        }  
    }  
    return s;  
}
```

- a) qual o tempo de execução do melhor caso em notação  $\Theta$ .  
b) qual o tempo de execução do pior caso em notação  $\Theta$

c) quais as condições que o vetor  $a$  deve satisfazer para caracterizar o melhor caso.

3. Qual o tempo de execução para o pior caso em notação  $\Theta$ , para o algoritmo abaixo escrito em linguagem C.

```
int f(int n) {  
    int i, j, k, sum = 0;  
    for ( i=1; i < n; i *= 2 ) {  
        for ( j = n; j > 0; j /= 2 ) {  
            for ( k = j; k < n; k += 2 ) {  
                sum += (i + j * k);  
            }  
        }  
    }  
}
```

4. Suponha que o vetor  $a$  contenha  $n$  valores. Suponha também que a função *randomValue* necessite de um número constante de processamentos para retornar cada valor, e que a função *goodSort* leve um número de etapas computacionais proporcional a  $n \log n$  para ordenar o vetor. Qual o tempo de execução para o pior caso em notação  $\Theta$ , para o seguinte fragmento de código, escrito em linguagem C.

```
for ( i = 0; i < n; i++ ) {  
    for ( j = 0; j < n; j++ ) {  
        a[ j ] = randomValue( i );  
    }  
    goodSort( a );  
}
```

5. Utilize uma das técnicas conhecidas de análise de algoritmos recursivos e forneça um limite assintótico  $\theta()$  para cada algoritmo abaixo, escrito em C:

- a) `int SomaInteiros(int A[], int n){  
    if (n<0) return 0;  
    else return A[n] + SomaInteiros(A, n-1);  
}`
- b) `int easyQuestion(int* A, int n) {  
    int i;  
    if (n < 2) return (A[0]);  
    for (i=n/2; i<(n/2)+8; i++)  
        return A[i] + easyQuestion(A, 3*n/4);  
}`
- c) `int youWontGuessThisOne(int* A, int n){  
    if (n < 50) return (A[n]);  
    int x, j;  
    x = youWontGuessThisOne(A, n/4);  
    for (j=0; j<n/3; j++) A[j] = A[n-j] - A[j];  
    x += youWontGuessThisOne(A, n/4);  
    return x;  
}`