

## PROGRAMAÇÃO II – SISTEMAS DE INFORMAÇÃO

### Lista de exercícios 11 – Análise de Complexidade

**Prof. Jean Eduardo Glazar**

- 1) Suponha dois algoritmos distintos  $A_1$  e  $A_2$  que desempenham a mesma tarefa, com tempos de execução expressados pelas funções  $T_1(n) = 8n + 3$  e  $T_2(n) = 2n^3 + n + 5$ , onde  $n$  representa o tamanho da entrada. Qual dos algoritmos é mais eficiente para executar a tarefa? Desenhe os gráficos correspondentes a cada função e dê a justificativa de sua afirmação a partir dos gráficos.
- 2) Dois algoritmos A e B possuem complexidade  $n^5$  e  $2^n$ , respectivamente. Você utilizaria o algoritmo B ao invés do A. Em qual caso? Exemplifique. Desenhe o gráfico para auxiliá-lo.
- 3) Dê a complexidade assintótica (notação O) de cada um dos algoritmos correspondentes aos programas abaixo:

```
a) void misterio1(int n) {
    int i, j;

    for (i = 1; i <= n; i++) {
        j = 0;
        while(j < 3) {
            printf("Este programa eh ridiculo!!!\n");
            j++;
        }
    }
}

b) void misterio2(int n) {
    float m[10][20];
    int i, j;

    /* Abaixo temos uma suposta funcao para preencher
       a matriz m. Considere que ela é O(1). */
    incializaMatriz(m);

    for (i = 0; i <=n; i++) {
        for (j = 0; j <20; j++) {
            printf("%f ", m[i][j] * n);
        }
        printf("\n");
    }
}
```

```
c) void misterio3(float* v, int n) {
    if( n < 0 ) { return; }
    float multiplicador;
    for(int i = 0; i <= n; i++) {
        multiplicador = v[i];
        for(int j = 2; j <= 5; j++) {
            v[i] = v[i] * multiplicador;
        }
    }
}

d) int misterio4(char* s, char c) {
    int n;
    n = strlen(s);

    if (!n) { return -1; }

    int i = 0;
    while((i < n) && (s[i] != c)) {
        i++;
    }
    if (i < n)
        return i;
    else
        return -1;
}

e) void misterio5() {
    int i, j;
    for( i = 1; i <= n; i++) {
        for( j = 1; j <= 2i; j++) {
            // trecho com O(1)
        }
    }
}

f) void misterio6() {
    int i, j;
    for( i = 1; i <= n; i++) {
        for( j = 1; j <= 2i; j++) {
            // trecho com O(j2)
        }
    }
}
```

```
g) void misterio7() {
    int i, j;
    for( i = 1; i <= n; i++) {
        for( j = 1; j <= n; j++) {
            // trecho com  $O(2^i)$ 
        }
    }
}

h) void misterio8() {
    int i, j;
    for( i = 1; i <= n; i++) {
        for( j = 1; j <= i; j++) {
            // trecho com  $O(2^j)$ 
        }
    }
}

i) void misterio9() {
    int i, j;
    for( i = 1; i <= n; i++) {
        for( j = i; j <= n; j++) {
            // trecho com  $O(2^j)$ 
        }
    }
}

j) float misterio10(int n, float A[], float x){
    int k;
    float y = 0.0;
    for (k = n; k >= 0; k--) {
        y = A[k] + y * x;
    }
    return y;
}

k) int misterio11(int n) {
    int i, j, x, soma = 0;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            for (x = 0; x < n; x++) {
                soma += n;
            }
        }
    }
    return soma;
}
```

```
1) void misterio12 (int* A, int n){
    int i, j, aux;
    for( j = 2; j <= n; j++){
        aux = A[j];
        i = j - 1;
        while (i > 0 && A[i] > aux){
            A[i + 1] = A[i];
            i = i - 1;
        }
        A[i + 1] = aux;
    }
}
```