Exercícios

Aluno: Henrique Castro e Silva **Matricula:** 711260

1 - Encontre a fórmula fechada do somatório $\sum_1^n (3i+5)^2$ e, em seguida, prove a usando indução matemática.

$$\sum_1^n (3i+5)^2 = \sum_1^n 9i^2 + 30i + 25 = 9\sum_1^n i^2 + 30\sum_1^n i + \sum_1^n 25 =$$

1. $\sum_1^n 25 = 25n$
2. $30\sum_1^n i = 30 \times \left(\frac{n(n+1)}{2}\right) = 15n(n+1) = 15n^2 + 15n$
3. $9\sum_1^n i^2 = 9 \times \left(\frac{n(n+1)(2n+1)}{6}\right) = \frac{3}{2} \times n(n+1)(2n+1) = \frac{3}{2}(2n^3 + 3n^2 + n)$

Final: $3n^3 + \frac{9}{2}n^2 + \frac{3}{2}n + 15n^2 + 15n + 25n = \mathbf{3n^3 + 19,5n^2 + 41,5n}$

Indução matemática:

I. $3(1)^3 + 19{,}5(1)^2 + 41{,}5(1) = \mathbf{64}$
$(3(1)+5)^2 = (8)^2 = \mathbf{64}$

II. $S_n = S_{n-1} + a_n$
$= 3(n-1)^3 + 19{,}5(n-1)^2 + 41{,}5(n-1) + (3n+5)^2$
$= 3(n^3 - 3n^2 + 3n + 1) + 19{,}5(n^2 - 2n + 1)^2 + 41{,}5n - 41{,}5 + 9n^2 + 30n + 25$
$= 3n^3 - 9n^2 + 9n - 3 + 19{,}5n^2 - 39n + 19{,}5 + 41{,}5n - 41{,}5 + 9n^2 + 30n + 25$
$= \mathbf{3n^3 + 19,5n^2 + 41,5n}$

2 - Um desafio no projeto de algoritmos é a obtenção de um custo computacional reduzido. Para isso, uma habilidade do projetista é contar o número de operações realizadas pelo algoritmo. Para cada trecho de código abaixo, apresente a função de complexidade *f(n)* para o melhor e para o pior caso considerando a operação de multiplicação. Apresente também a ordem de complexidade desse trecho do algoritmo.

I.

```
for (int i = n; i >= 1; i /= 2) {
    a *= 2;
}
```

$f(n) = \lfloor log_2 n \rfloor + 1$

Complexidade = $\theta(log_2 n)$

II.

```
for (int i = 0; i < 3 ; i++){
    for (int j = 0; j < n - 1; j++){
        l = a * 2 + b * 5;
    }
}
```

$f(n) = 3 \times (n - 1) \times 2 = 6 \times (n - 1)$

Complexidade = $\theta(n)$

3 – Desenvolva as funções para somar, multiplicar e transpor matrizes quadradas de dimensão n x n. Determine a função e a ordem de complexidade de cada uma dessas funções.

Somar Matrizes:

```
    //f(n) = n*n = n^2;
    //Complexidade = teta(n^2)
    public static void somarMatriz(int a[][], int b[][], int n){
        int c[][]=new int[n][n];

        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                c[i][j]=a[i][j]+b[i][j];
                System.out.print(c[i][j]+" ");
            }
            System.out.println();
        }
    }
```

Multiplicar Matrizes:

```
    //f(n) = n*n*n = n^3;
    //Complexidade = teta(n^3)
    public static void multiMatriz(int a[][], int b[][], int n){
        int c[][]=new int[n][n];

        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                c[i][j]=0;
                for(int k=0;k<n;k++)
                {
                    c[i][j]+=a[i][k]*b[k][j];
                }
                System.out.print(c[i][j]+" ");
            }
            System.out.println();
        }
    }
```

Transpor Matrizes:

```
    //f(n) = n*n*n = n^3;
    //Complexidade = teta(n^3)
    public static void transMatriz(int a[][], int b[][], int n){
        int c[][]=new int[n][n];

        for(int i = 0;i<n;i++){
            for(int j = 0;j<n;j++){
                b[i][j] = 0;
                for(int k = 0;k<n;k++){
                    b[i][j]=a[j][i];
                }
                System.out.print(b[i][j]+" ");
            }
            System.out.println();
        }
    }
```

4 - Aplique perturbação da soma para encontrar a fórmula do somatório $S_n = \sum_{i=0}^{n} i^2$.

$$S_n = \sum_{0}^{n} i^2$$

Não funciona a partir desse método:

1. $S_n + (n+1)^2 = 0^2 + \sum_0^n (i+1)^2$
2. $S_n + (n+1)^2 = \sum_0^n i^2 + \sum_0^n 2i + \sum_0^n 1$

Então começar pelo somatório de cubo:

$$S_{cubo} = \sum_{0}^{n} i^3$$

1. $S_{cubo} + (n+1)^3 = 0^3 + \sum_0^n (i+1)^3$
2. $S_{cubo} + (n+1)^3 = \sum_0^n i^3 + \sum_0^n 3i^2 + \sum_0^n 3i + \sum_0^n 1$
3. $(n+1)^3 = 3S_n + \frac{3}{2}n(n+1) + n + 1$
4. $6S_n = 2(n+1)^3 - 3n^2 - 3n - 2n - 2$
5. $6S_n = 2n^3 + 6n^2 + 6n + 2 - 3n^2 - 3n - 2n - 2$
6. $S_n = (2n^3 + 3n^2 + n)/6$