- Lista de Exercícios Adicionais (Análise de Algoritmos) –

Exercício 1:

Seja o algoritmo abaixo:

```
DETERMINA_K (n)
  k ← 0
  para i variando de 1 até n faça ← loop externo
    para j variando de 1 até i faça ← loop interno
      k ← k + 1
  retorne k
```

Analise o algoritmo e determine quanto tempo ele consome para executar em termos de número de operações primitivas, em função do tamanho da entrada n . Para simplificar a análise, suponha que cada linha, quando executada, consome tempo exatamente igual a 1.

Exercício 2:

Conforme se sabe, podemos obter uma boa aproximação para o cálculo de $\exp(x)$ (isto é, e^x) através da série de somas parciais:

$$S_n = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Na verdade, trata-se apenas de uma aproximação, pois

$$\lim_{n \rightarrow \infty} S_n = \exp(x)$$

Veja um algoritmo para calcular $\exp(x)$, com um limite de precisão que interrompe o cálculo após o n -ésimo termo:

```
EXP (x,n)
  S ← 1
  para i de 1 até n faça /* calcula n termos da série */
    num ← 1
    den ← 1
    para j de 1 ate i faça /* calcula numerador e o denominador */
      num ← num * x /* num ← x^i */
      den ← den * j /* den ← i! */
    S ← S + num / den
  retorne S
```

Determine o tempo de execução do algoritmo, $T(n)$, assumindo que cada linha, quando executada, consome tempo exatamente igual a 1.

Exercício 3:

É possível melhorar a eficiência do cálculo de $\exp(x)$. Vamos rever a série:

$$S_n = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Observamos que o cálculo da série com n termos pode ser escrita na forma:

$$S_n = 1 + T_1 + T_2 + \dots + T_n$$

onde cada T_i é um termo da série, e pode ser determinado a partir de seu anterior T_{i-1} :

$$T_1 = x$$

$$T_i = T_{i-1} \cdot \frac{x}{i}, \quad \forall i > 1$$

Vejam os então uma nova versão do algoritmo:

```
EXP (x,n)
  T ← 1
  S ← T
  para i de 1 até n faça      /* calcula n termos da série */
    T ← T * x / i
    S ← S + T
  retorne S
```

Determine o tempo de execução do novo algoritmo, $T(n)$.

Exercício 4:

Prove que são verdadeiras ou que são falsas as afirmações abaixo:

- a) n é $\mathcal{O}(n^2)$
- b) n^2 é $\mathcal{O}(n)$
- c) n^2+3n-5 é $\mathcal{O}(n^2)$
- d) n^2 é $\mathcal{O}(n^3)$
- e) n^3 é $\mathcal{O}(n^2)$
- f) n^2+3n-5 é $\mathcal{O}(n^3)$
- g) n^5 é $\mathcal{O}(2^n)$
- h) $n!$ é $\mathcal{O}(n^n)$
- i) $10n$ é $\Omega(n)$
- j) $\log n$ é $\Omega(n)$
- k) $1/2n^2-3n$ é $\Theta(n^2)$