



Unidade 2 – Exemplo de Aplicação do Modelo de Knuth



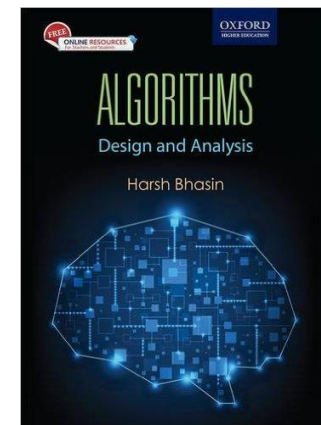
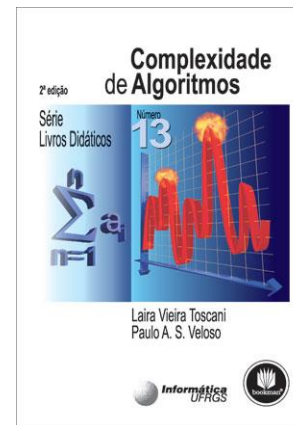
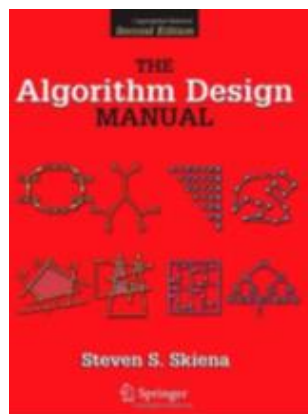
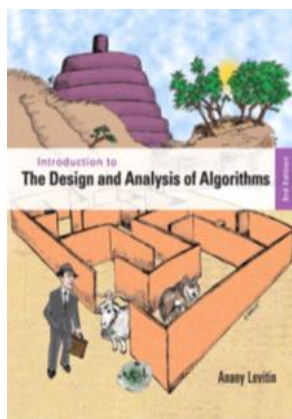
Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUVSP
aparecidovfreitas@gmail.com





Bibliografia

- Algorithm Design and Applications – Michael T. Goodrich, Roberto Tamassia, Wiley, 2015
- Introduction to the Design and Analysis of Algorithms – Anany Levitin, Pearson, 2012
- The Algorithm Design Manual – Steven S. Skiena, Springer, 2008
- Complexidade de Algoritmos – Série Livros Didáticos – UFRGS
- Algorithms – Design and Analysis – Harsh Bhasin – Oxford University Press - 2015





Exemplo

Usando o modelo simplificado, apresente a equação que define o tempo de processamento do algoritmo que calcula a somatória de uma série aritmética simples:

$$\sum_{i=1}^n i$$





Exemplo



O algoritmo pode ser implementado pelo código abaixo:

```
package maua;
import java.util.Scanner;

public class Somatoria {

    public static int Soma (int n){

        int resultado = 0;
        for (int i = 1; i <= n; ++i)
            resultado += i;
        return resultado;
    }

    public static void main(String[] args) {

        Scanner Input = new Scanner (System.in);

        System.out.print("Entre com o valor de n: ");

        int n = Input.nextInt();

        int resposta = Soma(n) ;

        System.out.println("Soma(" + n + ")= " + Soma(n));

    }
}
```





```
public static int Soma (int n){
```

```
1     int resultado = 0;  
2     for (int i = 1; i <= n; ++i)  
3         resultado += i;  
    return resultado;  
}
```

```
public static void main(String[] args) {
```

```
    ...
```

```
    int resposta = Soma(n) ;
```

```
    ...
```

```
}
```





```
public static int Soma (int n){
```

```
1  ──▶ int resultado = 0;  
      for (int i = 1; i <= n; ++i)  
          resultado += i;  
      return resultado;  
}
```

2 operações





```
public static int Soma (int n){
```

2a →

```
    int resultado = 0;  
    for (int i = 1; i <= n; ++i)  
        resultado += i;  
    return resultado;  
}
```

Código	Tempo
int i = 1	2 operações

- Este tempo corresponde à primeira parte do código for que representa a etapa de inicialização;
- É executado uma única vez antes da primeira iteração do loop.





```
public static int Soma (int n){
```

```
    int resultado = 0;  
2b → for (int i = 1; i <= n; ++i)  
        resultado += i;  
    return resultado;  
}
```

Código	Tempo
i <= n	3 x (n+1) operações

- Este tempo corresponde ao teste de término do loop;
- É executado antes do início de cada iteração do loop;
- O número de vezes em que o teste de término do loop é feito é um a mais que o número de vezes em que o corpo do loop é executado.





```
public static int Soma (int n){
```

```
    int resultado = 0;  
2c → for (int i = 1; i <= n; ++i)  
        resultado += i;  
        return resultado;  
}
```

Código	Tempo
++i	4 x (n) operações

- Este tempo corresponde ao terceiro elemento do for, o passo de incremento do contador do loop. Equivale a $i = i + 1$;
- É executado uma vez a cada iteração do loop. Portanto, n vezes.





```
public static int Soma (int n){  
  
    int resultado = 0;  
    for (int i = 1; i <= n; ++i)  
3  → resultado += i;  
    return resultado;  
}
```

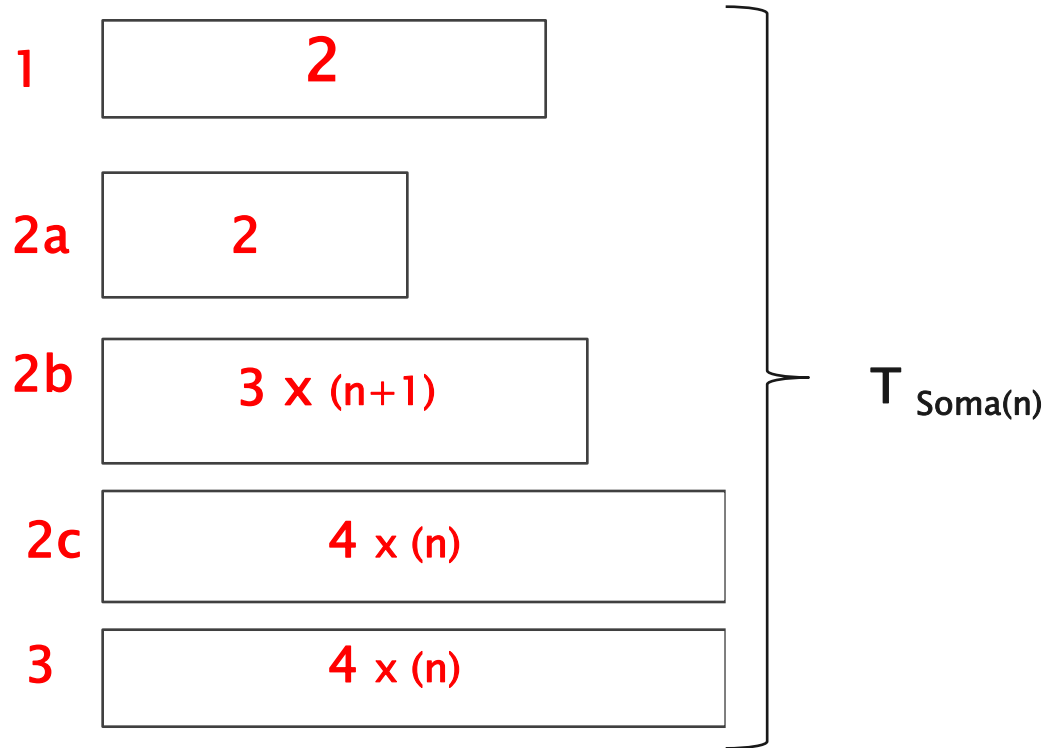
Código	Tempo
resultado += i	4 x (n) operações

- Este tempo corresponde ao corpo do loop;
- É executado n vezes.





Contagem Total de Operações



Contagem Total de Operações

