

## Fontes principais

1. Cormen T. H.; Leiserson C. E.; Rivest R.; Stein C.. *Introduction to Algorithms*, 3<sup>a</sup> edição, MIT Press, 2009
2. Análise de algoritmo - IME/USP (prof. Paulo Feofiloff)  
[http://www.ime.usp.br/~pf/analise\\_de\\_algoritmos](http://www.ime.usp.br/~pf/analise_de_algoritmos)

## Projeto de algoritmos por indução

## Projeto de algoritmos por indução

Usaremos indução para projetar algoritmos que resolvem certos problemas

A formulação do algoritmo será análoga a uma demonstração por indução

## Projeto de algoritmos por indução

Assim para resolver um problema  $P$ :

- mostramos como resolver instâncias pequenas de  $P$  (base) e
- mostramos como obter uma solução de uma instância  $P$  a partir das soluções de instâncias menores de  $P$

## Projeto de algoritmos por indução

O processo indutivo resulta em algoritmos recursivos tais que:

- a base da indução corresponde à solução de casos base de recursão
- a aplicação da hipótese de indução corresponde a uma ou mais chamadas recursivas
- o passo da indução corresponde ao processo de obtenção da resposta para o problema original a partir das respostas devolvidas pelas chamadas recursivas

## Projeto de algoritmos por indução

- um benefício imediato é que o uso (correto) da técnica dá uma prova de corretude do algoritmo
- a complexidade do algoritmo é expressa por uma recorrência
- muitas vezes é imediato a conversão do algoritmo recursivo em um iterativo
- frequentemente o algoritmo é eficiente, embora existam exemplos simples em que isso não acontece

## Exemplo: cálculo do polinômio

### Problema:

**Entrada:** sequência de números reais  $a_n, a_{n-1}, \dots, a_1, a_0$  e  $x \in \mathbb{R}$

**Saída:**  $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

Estamos interessados em projetar um algoritmo que efetue o menor número de operações aritméticas

## Exemplo: cálculo do polinômio

Solução 1:

**Hipótese de indução: (1a. tentativa)**

Dada uma sequência de números reais  $a_{n-1}, a_{n-2}, \dots, a_1, a_0$  e  $x \in \mathbb{R}$ , sabemos calcular  $P_{n-1}(x) = a_{n-1}x^{n-1} + \dots + a_1x + a_0$

**Base:** Para  $n = 0$  a solução é  $a_0$

**Passo:** Para calcular  $P_n(x)$ , basta calcular  $x^n$ , multiplicar o resultado por  $a_n$  e somar com  $P_{n-1}(x)$



## Solução 1

Calculo-Polinomio( $A, n, x$ )

```
1  se  $n = 0$ 
2      então  $P = a_0$ 
3      senão
4           $A' = a_{n-1}, \dots, a_1, a_0$ 
5           $P' = \text{Calculo-Polinomio}(A', n - 1, x)$ 
6           $x_n = x$ 
7          para  $i = 0$  até  $n$  faça
8               $x_n = x_n * x$ 
9               $P = P' + a_n * x_n$ 
10 retorne  $P$ 
```

## Solução 1

Seja  $T(n)$  o número de operações aritméticas realizadas pelo algoritmo

$$T(n) = \begin{cases} 0 & , \text{ se } n = 0 \\ T(n-1) + n \text{ multiplicações} + 1 \text{ adição} & , \text{ se } n > 0 \end{cases}$$

## Solução 1

Não é difícil perceber que:

$$\begin{aligned} T(n) &= \sum_{i=1}^n (i \text{ multiplicações } + 1 \text{ adição}) = \\ &= \frac{n(n+1)}{2} \text{ multiplicações } + n \text{ adições} = O(n^2) \end{aligned}$$

Note que a solução desperdiça tempo recalculando  $x^n$ .

## Solução 2

**Ideia:** Eliminar a computação reduntante movendo o cálculo de  $x^{n-1}$  para dentro da hipótese de indução.

**Hipótese de indução:** (2a tentativa)

Sabemos calcular  $P_{n-1} = a_{n-1}x^{n-1} + \dots + a_1x + a_0$  e também o  $x^{n-1}$

**Base:** Para  $n = 0$  a solução é  $(a_0, 1)$

**Passo:** Primeiro calculamos  $x^n$  multiplicando  $x$  por  $x^{n-1}$ , depois calculamos  $P_n(x)$  multiplicando  $x^n$  por  $a_n$  e somando o valor obtido com  $P_{n-1}(x)$

## Solução 2

Calculo-Polinomio( $A, n, x$ )

```
1  se  $n = 0$ 
2      então  $P = a_0$ 
3           $x_n = 1$ 
4  senão
5       $A' = a_{n-1}, \dots, a_1, a_0$ 
6       $(P', x') = \text{Calculo-Polinomio}(A', n - 1, x)$ 
7       $x_n = x * x'$ 
8       $P = P' + a_n * x_n$ 
9  retorne  $(P, x_n)$ 
```

## Solução 2

$T(n)$  : número de operações aritméticas

$$T(n) = \begin{cases} 0 & , \text{ se } n = 0 \\ T(n-1) + 2 \text{ multiplicações} + 1 \text{ adição} & , \text{ se } n > 0 \end{cases}$$

## Solução 2

A solução da recorrência é:

$$T(n) = \sum_{i=1}^n (2 \text{ multiplicações } + 1 \text{ adição}) =$$

$$= 2n \text{ multiplicações } + n \text{ adições } = O(n)$$

## Solução 3

Considerar o polinômio  $P_{n-1}(x)$  na hipótese de indução não é a única escolha possível.

A hipótese de indução ainda pode ser reforçada para termos ganho de complexidade.



## Solução 3

**Hipótese de indução:** (3a. tentativa) Sabemos calcular o polinômio  $P'_{n-1}(x) = a_n x^{n-1} + a_{n-1} x^{n-2} + \cdots a_2 x + a_1$ .

**Note que:**  $P_n = P'_{n-1}(x) * x + a_0$ , ou seja, com apenas 1 multiplicação e 1 adição calculamos  $P_n(x)$  a partir de  $P'_{n-1}(x)$ .

## Solução 3

Calculo-Polinomio( $A, n, x$ )

```
1  se  $n = 0$ 
2      então  $P = a_0$ 
3      senão
4           $A' = a_n, a_{n-1} \cdots, a_1$ 
5           $P' = \text{Calculo-Polinomio}(A', n - 1, x)$ 
6           $P = x * P' + a_0$ 
7  retorne  $P$ 
```

Note que a base de indução é trivial, pois para  $n = 0$  a solução é  $a_0$ .

## Solução 3

$T(n)$  : número de operações aritméticas

$$T(n) = \begin{cases} 0 & , \text{ se } n = 0 \\ T(n-1) + 1 \text{ multiplicação} + 1 \text{ adição} & , \text{ se } n > 0 \end{cases}$$

## Solução 3

A solução da recorrência é:

$$\begin{aligned} T(n) &= \sum_{i=1}^n (1 \text{ multiplicação} + 1 \text{ adição}) = \\ &= n \text{ multiplicações} + n \text{ adições} = O(n) \end{aligned}$$

Esta última maneira de calcular  $P_n(x)$  é chamada de **regra de Horner**.

Obrigado