

# MAC121 - Algoritmos e Estruturas de Dados I

Universidade de São Paulo

Segundo Semestre de 2020

## Informações Gerais

MAC0110 × MAC0121

Problema – Algoritmo – Programa

Linguagem de Programação C

## MAC0121 – Algoritmos e Estruturas de Dados I

- ▶ Carlos E. Ferreira – [cef@ime.usp.br](mailto:cef@ime.usp.br)
- ▶ Página da disciplina na plataforma [edisciplinas](#)
- ▶ Monitor (será divulgado)
- ▶ Slides + sessões online (com cerca de 1 hora)

## MAC0121 – Algoritmos e Estruturas de Dados I

### Programa:

- ▶ Noções informais de provas de correção e medidas de desempenho de algoritmos;
- ▶ Noções de tipos abstratos de dados: vetores, listas, árvores, etc;
- ▶ Pilhas, filas, filas de prioridade, aplicações;
- ▶ Recursão;
- ▶ Algoritmos de ordenação: elementares, quicksort, mergesort, heapsort, ordenação indireta;
- ▶ tabelas de símbolos;
- ▶ Diversas técnicas algorítmicas como guloso, divisão e conquista, programação dinâmica, etc.

## MAC0121 – Algoritmos e Estruturas de Dados I

Critério de Aprovação:

- ▶ 4 EPs:

$$MEP = \frac{EP_1 + EP_2 + EP_3 + EP_4}{4}$$

- ▶ 4 Listas de exercícios: *ML*: média aritmética das listas



$$MF = \frac{2 * MEP + ML}{3}.$$

## MAC0121 – Algoritmos e Estruturas de Dados I

### Bibliografia:

- ▶ Algoritmos em Linguagem C - P. Feofiloff  
Disponível no drive compartilhado  
utilizando o email USP
- ▶ site do Prof. Feofiloff



## MAC0110 × MAC0121

*Algoritmos devem ser corretos, eficientes e elegantes.*  
*Paulo Feofiloff*

Exemplo:

Dado um inteiro  $n > 0$ , calcule  $1 + 2 + \dots + n$ .

Mais um exemplo:

Dado um inteiro  $n > 0$ , verifique se  $n$  é um número primo.

## MAC0110 × MAC0121

Em MAC0110 apresentamos soluções para problemas como os exemplificados acima.

Os programas deviam estar **corretos**, mas não damos muita importância para eficiência ou elegância.



## Problema – Algoritmo – Programa

Em Computação formulamos soluções **algorítmicas** para **problemas**. Estas soluções, quando implementadas dão origem a **programas**.

Um **problema** é caracterizado por:

- ▶ possíveis dados (entradas);
- ▶ possíveis respostas (saídas);
- ▶ condições que devem ser satisfeitas para que uma determinada saída corresponda a uma dada entrada.

## Problema

Exemplos:

- ▶ Dado um inteiro positivo  $n > 0$ , verificar se  $n$  é primo.
- ▶ Achar as raízes de um polinômio de grau  $n > 0$  de coeficientes reais.
- ▶ Ordenar uma sequência de  $n > 0$  números reais.

Uma **instância** (exemplar) do problema é um dado particular, quando os parâmetros estão fixados.

- ▶ 7139 é primo?
- ▶ Quais são as raízes de  $-7x^{18} + 45x^6 - 3x^3 + 7x + 14$ ?
- ▶ Ordenar (2.0, -5.1, 1.7, 3.0, 7.4, -3.2, 4.5, 9.3).

## Algoritmo

Um **algoritmo** é um processo capaz de, para qualquer instância da entrada de um problema, produzir, em tempo finito e usando uma quantidade finita de recursos, a saída esperada.

- ▶ cada passo deve ser preciso e claro;
- ▶ cada passo deve levar tempo finito e consumir recursos finitos para ser executado;
- ▶ para qualquer instância do problema, o algoritmo executa um número finito de passos.

## Algoritmo

Muhammad ibn Musa al-Khwarizmi (780 - 850): matemático árabe com importantes trabalhos em Álgebra e outras áreas. Em um de seus trabalhos mostra o Algoritmo de Euclides para encontrar o mdc entre dois números.



Um **programa** é uma descrição de um algoritmo usando uma linguagem de programação.

## Correto, eficiente e elegante

- ▶ um algoritmo é **correto** se faz o que se espera, se cumpre o que a documentação promete;
- ▶ É **eficiente** se não desperdiça tempo e espaço. Se conhecemos duas soluções para um problema vamos preferir aquela cuja execução consome menos tempo;
- ▶ É **elegante** se for simples, bonito, limpo e sem enfeites. Não trata casos especiais em separado, não tem código nem variáveis desnecessárias.

Exemplo: Dados inteiros  $a, b > 0$ , calcular o  $mdc(a, b)$ .

## A linguagem C

```
#include <stdio.h>

... /* bibliotecas e definições globais no
arquivo */

/* protótipos das funções locais */

int main () {
    /* declarações */

    /* comandos */

    return (0);
}

/* funções locais */
```

## Alguns exemplos...

- ▶ Dada uma sequência de números inteiros terminada por zero, imprima a soma dos números pares da sequência.
- ▶ Dado um inteiro  $n > 0$ , decide se existem primos  $p$  e  $q$  tais que  $n = p + q$ .
- ▶ Dado  $n > 0$  imprima os coeficientes de  $(a + b)^n$ . Por exemplo, se  $n = 4$  deve imprimir 1, 4, 6, 4, 1.

## Para saber mais...

Material interessante sobre leiaute e documentação de programas:

- ▶ Apêndice A do livro texto;
- ▶ Material sobre leiaute (site do P. Feofiloff) ;
- ▶ Material sobre documentação (site do P. Feofiloff).