Processamento da Informação

Fabrício Olivetti de França

02 de Fevereiro de 2019

UFABC

CMCC
Centro de Matemática, Computação e Cognição
Center for Mathematics, Computation and Cognition

Topics

1. Processamento da Informação
2. Conceitos de Programação
3. Funções e Programas de Computador

Processamento da Informação

Informação

Do latim, *informatio onis*, conceber ideias.

Informação

Informar é transmitir conhecimento:

- Professor aos alunos
- Médico ao paciente
- Jornais aos leitores
- Usuário ao computador

Processamento da Informação

Organizar, transformar dados de acordo com uma sequência de instruções.

Processamento da Informação

Eu entrego uma informação ao computador e ele me retorna uma nova informação!

Processamento da Informação

Isso é feito com o auxílio de conceitos como **algoritmos**, **funções** e **programas de computador**.

Computador ao nosso serviço

O computador é uma máquina de calcular avançada, temos que tirar proveito disso para nossas próprias tarefas!

Mas para isso precisamos nos comunicar...

Programação de Computadores

Muitos países estão incorporando curso de programação de computadores no ensino básico e médio.

Importância da Computação

Relevante para diversas áreas:

- Carros automatizados
- Detecção de plágio
- Verificação de Fake News
- Encontrar funções biológicas

Objetivo do Curso

Ensinar o **pensamento computacional** para resolver problemas com o auxílio do computador.

Ferramentas

Para isso utilizaremos uma **linguagem de programação** escolhida pelo docente do curso para nos comunicarmos com o computador.

Metodologia

3 horas de aulas teóricas + 2 horas de aulas práticas

Site do Curso (teoria)

https://folivetti.github.io/teaching/2020-summer-teaching-1

Empenho x Desempenho

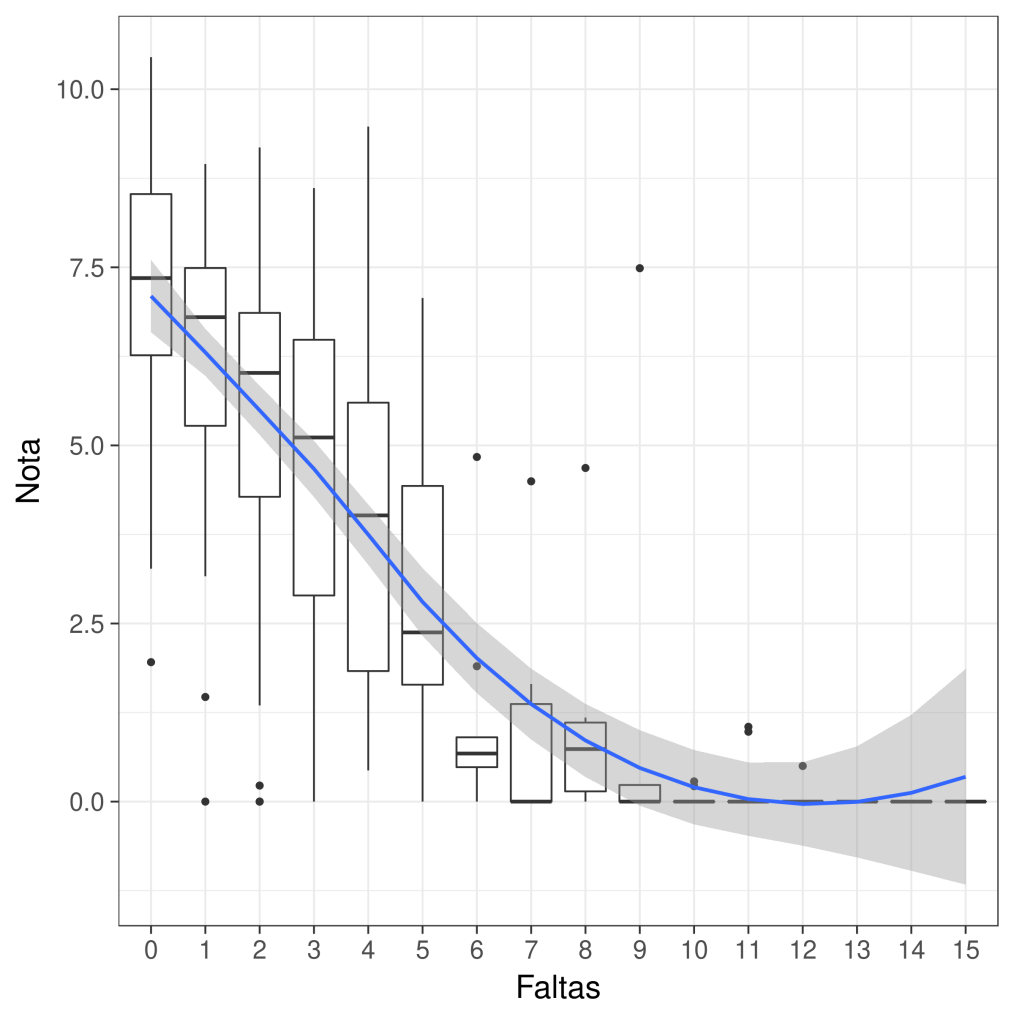

Figura 1: FONTE: prof. Thiago Covões

Conceitos de Programação

Computação

Computação se refere a um cálculo, aritmético ou não, seguindo um modelo bem definido para a solução de um problema.

Não necessariamente utilizando um computador...

Algoritmo

Algoritmo é a descrição de uma solução de um problema computável.

Seu nome vem de **al-Khwarizmi**, um dos percursores da algebra.

Máximo Divisor Comum

O primeiro algoritmo que se tem conhecimento é o **Algoritmo de Euclides**, utilizado para calcular o **Máximo Divisor Comum**.

Máximo Divisor Comum

Dados $a, b \in \mathbb{N}$:

```
mdc(a, 0) = a
mdc(0, b) = b
mdc(a, b) = mdc(b, a%b)
```

Definição: a, b são **argumentos** ou **entradas** de nosso algoritmo.

Exercício

Utilizando a definição do MDC, calcule:

$mdc(15, 25)$

$mdc(78, 66)$

$mdc(132, 154)$

Propriedades de um Algoritmo

Todo algoritmo deve possuir quatro propriedades para ser definido como tal:

1. Finitude
2. Desambiguidade
3. Conjunto de entrada
4. Conjunto de saída

Finitude

Um algoritmo **SEMPRE** deve terminar em um período finito de tempo.

- Como o segundo argumento do algoritmo de Euclides sempre diminui, e por serem definidos para números naturais, eventualmente esse chegará a zero e terminará.

Desambiguidade

Não pode haver ambiguidade em nenhuma das instruções do algoritmo.

- *Vá até a loja e compre duas caixas de leite, e se tiver ovos, compre seis*
- $1 + 2 * 3 = ??$

Entrada

O algoritmo recebe um conjunto de entradas (argumentos) que pode ser vazio, finito ou infinito.

- No algoritmo de Euclides temos o conjunto de entradas $a, b \in \mathbb{N}^2$.

Saída

O algoritmo deve produzir uma (ou mais) saída como resultado do processamento. Não faz sentido perguntarmos algo que não tenha uma resposta.

- No algoritmo de Euclides temos como resposta o máximo divisor comum $m \in \mathbb{N}$.

Funções e Programas de Computador

Os dois lados do algoritmo

A definição de **algoritmo** não é formalizada na área de Ciência da Computação. Ele é apenas uma abstração do pensamento computacional.

Os dois lados do algoritmo

Por outro lado, temos dois conceitos formais que podem definir um algoritmo: **funções** e **programas de computador**.

Funções

Uma **função** $f : X \to Y$ é um mapa de elementos do conjunto X para elementos do conjunto Y.

$mdc : \mathbb{N}^2 \to \mathbb{N}$

$dobra : \mathbb{N} \to \mathbb{N}$

Tipos

Um **tipo** é um conjunto nomeado de valores que apresentam alguma propriedade comum.

Exemplos: $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \{F, V\}$, *primos*

Tipos

Vamos definir o tipo denominado **L** que representa o estado de uma lâmpada:

- L = {On, Off}

Figura 2: Lâmpadas On e Off

Imagine que em uma sala contendo uma lâmpada temos um *botão* que executa uma função que pode alterar o estado da lâmpada.

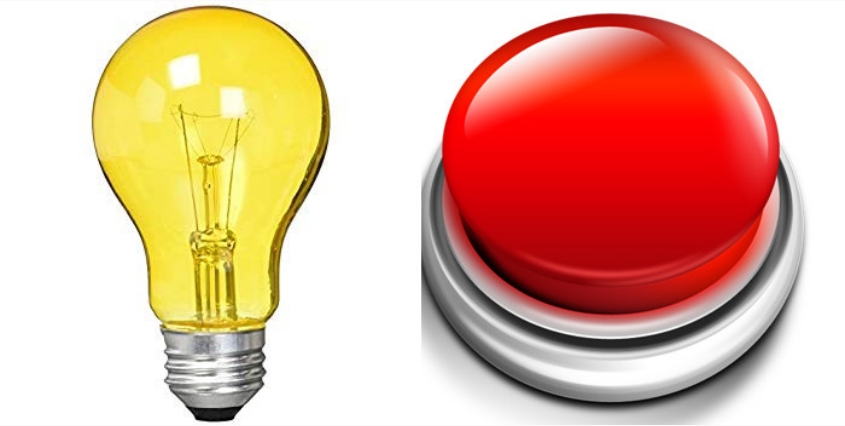

Figura 3: Botão ($f(lamp)$)

Tipos

Podemos formalizar essa função como $f : L \longrightarrow L$.

Como você imagina essa função? Quantas possíveis definições existem?

Tipos

1. $f1(lamp) = On$
2. $f2(lamp) = Off$
3. $f3(lamp) = lamp$
4. $f4(lamp) = \begin{cases} On, & lamp = Off \\ Off, & lamp = On \end{cases}$

Tipos

Se pensarmos nos valores **verdadeiro** e **falso** substituindo os valores On/Off, o que cada função representa?

Funções e Algoritmos

Podemos pensar em uma função como um algoritmo implementado na linguagem matemática.

O algoritmo *mdc* é uma função!

Composição de Funções

Um conceito importante para a programação de computadores é a composição de funções, que permite reutilização de definições anteriores:

$f(x) = 2 * x, g(x) = x + 1$

$f(g(x)) = 2 * (x + 1)$

$g(f(x)) = 2 * x + 1$

Exercício

Dadas as funções $f(x) = 2 * x$, $g(x) = x + 1$, qual o resultado de:

- $f(g(2))$
- $g(f(2))$
- $f(g(5))$
- $g(f(5))$

Programa de Computador

Um programa de computador é um conjunto de instruções de máquina que implementam um algoritmo.

Passo a passo de como o computador deve processar os dados.

Programa de Computador

Internamente ele é definido por sequências de bits.

Cada sequência de bit é mapeada para uma instrução do processador.

Programa de Computador

001000	00001	0000000101011110
Código OP	Endereço	Valor
add	eax	360

Algoritmo MDC em Linguagem de Máquina

```
mov esi, 68     # m = 68
mov ebx, 119    # n = 119
jmp .L2         # vai para o passo 2

.L3:
    mov ebx, edx    # n = r
.L2:
    mov eax, ebx
    idiv esi        # EAX = m / n (EAX), EDX = r
    mov esi, ebx    # m = n
    test  edx, edx  # verifica se o resto é zero
    jne .L3         # se teste anterior não zero,
                    # vai para L3
```

Linguagem de Máquina

- Difícil de ler, escrever e entender.
- Requer uma lógica de programação imperativa.

Linguagem de Alto Nível

Para resolver esses problemas, foram criadas linguagens de programação que serviriam como intermediários entre a linguagem de máquina e o programador.

Linguagem de Alto Nível

- Possui um conjunto de instruções próximas da linguagem natural.
- Minimiza o número de instruções para tarefas frequentes.
- Não requer completo entendimento do funcionamento do computador.

Linguagem de Alto Nível

Um arquivo texto contendo as instruções escritas em uma linguagem de programação é chamado de **código-fonte**.

Compilador x Interpretador

- **Compilador:** o código-fonte é traduzido para o código de máquina e escrito em forma de um arquivo executável.
- **Interpretador:** o código-fonte é traduzido para instruções de máquina durante a execução do programa.

Linguagem Python

Nesse curso utilizaremos a linguagem de programação **Python**:

- Tem uma sintaxe simples.
- Filosofia de códigos pequenos e legíveis.
- Declarativo (ou quase).

Funções no Python

A função *dobra* pode ser definida em Python como:

```
def dobra(x):
    return 2*x
```

Funções no Python

A palavra-chave *def*:

```
def nome_da_funcao(argumentos):
    codigo
```

Funções no Python

O nome de uma função e de seus argumentos devem seguir certas regras:

- Devem começar com uma letra ou um underscore (_).
- Os caracteres restantes podem conter letras, números e underscore (_).
- O Python diferencia entre letras minúsculas ou maiúsculas.

Escolhendo nomes

Para seu programa de computador ser legível, é importante escolher nomes adequados para as funções e variáveis.

Escolhendo nomes

- Os nomes de funções devem ser ações ou o nome de um algoritmo conhecido:

```
dobra, ordena, mdc, newton
```

Escolhendo nomes

- Os nomes de argumentos devem ser descritivos com sua função:

```
vetor, somatorio, total_aprovados
```

Escolhendo nomes

- Nomes compostos por múltiplas palavras devem ser separadas por underscore:

```
alunos_aprovados, soma_salarios
```

Escolhendo nomes

- Funções matemáticas podem utilizar variáveis com nomes x, y, n, sendo senso comum.

Funções no Python (cont.)

Após a linha em que definimos um nome para a função, escrevemos as instruções, uma em cada linha, alinhados com 4 espaços a direita de *def*:

```
def f(x):
    instrucao1
    instrucao2
    instrucao3
```

Funções no Python (cont.)

A instrução **return** indica a saída do algoritmo.

Exercício

Crie as seguintes funções em Python:

- `dobra`
- `quadruplica`
- soma_dois_numeros

Resposta

```
def dobra(x):
    return 2*x

def quadruplica(x):
    return 4*x
```

```
def quadruplica(x):
    return dobra(dobra(x))
```

```
def soma_dois_numeros(x, y):
    return x+y
```

Exercício

Crie as 4 possíveis definições da função botão para uma lâmpada.

Resposta

```
def f1(lampada):
    return "Off"

def f2(lampada):
    return "On"

def f3(lampada):
    return lampada

def f4(lampada):
    if lampada=="On":
        return "Off"
    if lampada=="Off":
        return "On"
```

Algoritmo MDC em Python

O algoritmo MDC em Python pode ser escrito como:

```
def mdc(x, y):
    if x==0:
        return y
    if y==0:
        return x
    return mdc(y, x%y)
```

Nas aulas seguintes aprenderemos mais detalhes de sintaxe.

Próxima aula

Aprenderemos um pouco mais sobre a sintaxe do Python, seus **tipos padrões**, uso de **variáveis auxiliares** e **condicionais**.