

CI1055: Algoritmos e Estruturas de Dados I

Prof. Dr. Marcos Castilho

Departamento de Informática/UFPR

9 de julho de 2020

Resumo

Sobre algoritmos, programas e linguagens de programação

Objetivos da aula

- apresentar o conceito de algoritmos
- apresentar o conceito de programas
- apresentar um panorama de linguagens de programação

Dos problemas às soluções

- a computação só existe para resolver problemas
- para um certo problema, podem haver várias soluções
- algumas soluções são melhores em algum caso particular
- as soluções são escritas em forma de algoritmos e depois na forma de programas
- os programas são escritos em linguagens de programação

O que é um algoritmo?

É uma sequência suficientemente precisa de instruções que detalha o processo de solução de um problema e que, quando executada por outra pessoa ou por uma máquina a partir dos dados de entrada, leva à solução do problema

Para que serve um algoritmo?

- para registrar o processo de solução de um problema
- para que outra pessoa possa resolver o mesmo problema, sem ter que reinventar a roda

Problema: como fazer um bolo

Bata em uma batedeira a manteiga e o açúcar. Junte as gemas uma a uma até obter um creme homogêneo. Adicione o leite aos poucos. Desligue a batedeira e adicione a farinha de trigo, o chocolate em pó, o fermento e reserve. Bata as claras em neve e junte-as à massa de chocolate misturando delicadamente. Unte uma forma retangular pequena com manteiga e farinha e leve para assar em forno médio pré-aquecido por aproximadamente 30 minutos. Desenforme o bolo ainda quente e reserve. . .

Características da receita de bolo

- outra pessoa pode fazer o bolo
- as frases estão no imperativo
- existe uma ordem para executar as ações
- algumas ações devem esperar outras terminarem
- algumas ações podem ter a ordem invertida, outras não

Problemas da receita de bolo

- algumas ações são omitidas, quem as escreveu acha que são óbvias (ex.: separar as gemas das claras)
- quem escreveu a receita espera que a outra que vai fazer o bolo tenha um mínimo de conhecimento culinário
- algumas instruções são obscuras:
 - até obter um creme homogêneo
 - misturando delicadamente
 - forno médio pré-aquecido

Pedir para quem escreveu a receita explicar melhor...

- como se faz claras em neve?
- como se separa a gema do ovo?
- como se quebra um ovo?
- onde achar um ovo?
- uso ovo marrom ou branco?
- pode ser ovo de codorna?

Quando se pode parar de detalhar ???

Nível máximo de detalhamento

- é preciso se convencionar o nível máximo de detalhamento
- assim, pode-se escrever a receita de bolo neste nível
- por exemplo: considera-se que quem vai fazer o bolo já fez estágio em alguma doceria por pelo menos 2 anos
- agora o algoritmo (receita) pode permanecer do jeito que está

O que é um programa?

- é um algoritmo escrito com o nível de detalhamento pré-definido para a máquina ou pessoa que vai executá-lo
- ele deve ser escrito de maneira extremamente precisa e rigorosa, sem ambiguidades, e deve produzir o mesmo resultado a cada execução
- a diferença para um algoritmo é que este pode ser escrito em mais alto nível e em linguagem menos formal (português, pseudo-linguagem, . . .)
- o programa deve ser escrito em uma linguagem de programação

Qual é o nível máximo de detalhamento?

- os computadores modernos usam o nível máximo de detalhamento conhecido como *o modelo von Neumann* estabelecido em 1946
- veremos isso na próxima aula

Pascal

by Niklaus Wirth, 1970

Por quê *Pascal*?

Linguagens e paradigmas de programação

- estruturado: *Pascal*, *C*, ...
- funcionais: *Lisp*, *Haskell*, ...
- lógicas: *Prolog*
- orientadas à objetos: *Java*, *SmallTalk*, ...
- ...

Para que serve um paradigma de programação?

Exemplos de linguagens e paradigmas

- vamos considerar um problema bem simples
- pois sem problema não há computação!

Imprimir o texto Alo mamae na tela

Paradigma estruturado

Pascal

```
program alomamae;  
begin  
    writeln ('Alo mamae');  
end.
```

C

```
#include <stdio.h>  
  
int main() {  
    printf ("Alo mamae\n");  
    return 0;  
}
```


Lisp

```
(defun hello ()  
  (format t "Hello,  
           World!~%" ))
```

Haskell

```
main :: IO ()  
main = putStrLn "Alo mame"
```

Prolog

```
?- write('Alo mamae'), nl.
```

Paradigma orientado à objetos

Java

```
public class Alomamae {  
  
    public static void main(  
        String[] args) {  
        System.out.println("Alo mamae");  
    }  
}
```

Smalltalk

```
object Alo {  
    def main(args: Array[  
        String]) = {  
        println("Alo mamae")  
    }  
}
```

Exemplos em linguagens variadas

Basic

```
10 PRINT "ALO MAMAE"  
20 END
```

Python

```
print "Alo mamae"
```

- todas as linguagens de programação seguem uma gramática
- elas variam em detalhes sintáticos e semânticos
- em todas elas o básico é comum

Enfim: porquê *Pascal*

- ela é simples
- é fortemente tipada (importante para calouros)
- compilador “amigável”
- possui compilador para diversos sistemas operacionais
- está presente em vasta literatura sobre algoritmos em geral
- e . . .

Enfim: porquê *Pascal*

- provavelmente você nunca mais vai usar!
- é importante aprender uma segunda linguagem de programação

- fazer os exercícios 2 e 3 do capítulo 3 do livro [1]

http://www.inf.ufpr.br/cursos/ci055/livro_alg1.pdf

- o conteúdo da parte importante desta aula está no livro no capítulo 3
- a parte sobre linguagens de programação não é relevante e pode ser facilmente encontrada na Internet e na literatura da área
- na próxima aula veremos o modelo Von Neumann

- Slides feitos em \LaTeX usando beamer
- Licença

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>