Introdução

Patrícia de Siqueira Ramos

UNIFAL-MG, *campus* Varginha

21 de Março de 2018

- Não existe, em geral, uma única solução para um problema
- Algumas soluções são melhores do que outras, sob algum critério
- Alguns problemas são casos particulares de outros similares
- Muitas vezes é melhor resolver um problema mais genérico, resolvendo-se uma classe de problemas e não apenas um

Fases na resolução de problemas

- Análise do problema
- Projeto do algoritmo
- Codificação
- Compilação e execução
- Verificação
- Depuração e manutenção
- Documentação

Exemplos de problemas

1) Contando o número de presentes em uma sala de aula: quantos alunos estão presentes na sala de aula neste momento, supondo capacidade para 50 pessoas?
Uma sugestão de algoritmo narrativo seria:

- Todos os alunos se levantam e se atribuem o número 1
- Todos os alunos se organizam em pares
- Em cada par, primeiro é somado o número de cada um dos dois, um deles guarda esse número e permanece de pé e o outro se assenta
- Os que ficarém de pé repetem o processo até que só exista um de pé

Exemplos de problemas

1) Contando o número de presentes em uma sala de aula:

a)
```
1
1
1
1
1
1
1
1
1
1
1
```

b)
```
1
1
1
1
1
1
1
1
1
1
```

Exemplos de problemas

1) Contando o número de presentes em uma sala de aula:

- Para 1.000 pessoas, o processo termina em 10 passos
- Para 1.000.000 de pessoas, em 20 passos

Exemplos de problemas

2) Trocando o pneu de um carro:

- Verifique a existência do estepe e de um macaco
- Levante o carro com o macaco
- Tire os parafusos da roda
- Tire a roda do eixo
- Coloque a roda com pneu novo no eixo
- Aperte os parafusos
- Abaixe o carro
- Retire o macaco

Exercícios sobre resolução de problemas

Sugira algoritmos narrativos para os seguintes problemas:

- Trocar uma lâmpada
- Sacar dinheiro no caixa eletrônico

Exemplos de problemas

Trocar uma lâmpada:

Exemplos de problemas

Trocar uma lâmpada:

- Pegue uma lâmpada nova
- Pegue uma escada
- Posicione a escada embaixo da lâmpada queimada
- Suba a escada levando a lâmpada nova
- Retire a lâmpada queimada
- Coloque a lâmpada nova
- Desça da escada
- Ligue o interruptor

Exemplos de problemas

Sacar dinheiro no caixa eletrônico:

Exemplos de problemas

Sacar dinheiro no caixa eletrônico:

- Insira o cartão no caixa eletrônico
- Solicite o saldo
- Digite a senha
- Se saldo $\geq$ quantia:
  Saque a quantia
- Senão:
  Saque o valor disponível
- Retire o dinheiro
- Retire o cartão

Algoritmo

- Sequência de ações executáveis para a obtenção de uma solução para um problema (ao ser lida e executada produz o resultado esperado)
- Instruções do tipo “faça”, “divida” etc. na forma sequencial
- Deve haver ordem nas instruções
- Evitar o uso de frases ambíguas ou imprecisas

Programa

- O programa é a codificação do algoritmo em alguma linguagem formal (C, C++, Pascal, Java, Python, R etc.)
- O computador apenas é capaz de executar o programa e não o algoritmo
- Assim, programar é uma habilidade que permite representar um algoritmo em uma notação (um programa) que possa ser executada por um computador

Teste:

Qual a alternativa correta?
Um algoritmo é:
a) Uma solução para um problema que pode ser resolvido por um computador.
b) Uma lista passo a passo de instruções que, caso seguidas exatamente, resolvem o problema sendo considerado.
c) Uma série de instruções implementadas em uma linguagem de programação.
d) Um tipo especial de notação usado por cientistas da computação.

Exemplo de algoritmo e programa:

Seja o seguinte algoritmo para somar duas notas:

```
Início
  Faça nota1 igual a 8
  Faça nota2 igual a 9
  Some nota1 e nota2
  Mostre o resultado da soma
Fim
```

Ou, de uma forma mais esquemática:

```
Início
  nota1 = 8
  nota2 = 9
  soma = nota1 + nota2
  Escreva(soma)
Fim
```

Exemplo de algoritmo e programa:

Em Python, o programa poderia ser:

```
nota1 = 8
nota2 = 9
soma = nota1 + nota2
print('A soma das notas é:', soma)
```

Exemplo de algoritmo e programa:

Em Python, o programa poderia ser:

```
nota1 = 8
nota2 = 9
soma = nota1 + nota2
print('A soma das notas é:', soma)
```

Após a execução, na tela apareceria:

```
A soma das notas é: 17
```

Exercício:

Como você alteraria o algoritmo da soma para que ele retornasse a média das duas notas?

```
Início
  nota1 = 8
  nota2 = 9
  soma = nota1 + nota2
  Escreva(soma)
Fim
```

Linguagens de programação

- As linguagens de programação mais populares são chamadas de linguagem de alto nível (Python, C++, PHP, Java, R etc. etc.)
- Mas o computador só consegue executar programas escritos em linguagens de baixo nível ("linguagens de máquina" ou "linguagens *assembly*")
- Assim, programas escritos em linguagens de alto nível precisam ser processados antes (o que leva algum tempo)
- É muito mais fácil e rápido programar em uma linguagem de alto nível. Os programas ficam mais curtos, mais fáceis de ler e mais simples de alterar
- Além disso, as linguagens de alto nível podem rodar em diferentes tipos de computador, com pouca ou nenhuma modificação
- Programas em baixo nível só podem rodar em um único tipo de computador e precisam ser reescritos para rodar em outro tipo

Interpretadores × compiladores

- Dois tipos de programas processam linguagens de alto nível fazendo a tradução para linguagens de baixo nível: interpretadores e compiladores
- Interpretador: lê um programa escrito em linguagem de alto nível e o executa
- Ele processa o programa um pouco de cada vez, alternadamente: ora lendo algumas linhas, ora realizando computações.

Interpretadores × compiladores

- Compilador: lê o programa e o traduz completamente antes que o programa comece a rodar
- Programa escrito em linguagem de alto nível: código fonte
- Programa traduzido: código objeto ou executável

- Muitas linguagens modernas se utilizam desses dois processos