23/04/2025

Universidade Estadual de Maringá – Departamento de Informática
Prof. Josiane M. Pinheiro

Conceitos Iniciais para a disciplina de Fundamentos de Algoritmos

Texto baseado nos livros:

"Lógica de programação", dos autores Alexandre Cruz Berg e Joice Pavek Figueiró, Editora da Ulbra, 2ª ed., 2001

"Fundamentos da Programação de Computadores", das autoras Ana Fernanda G. Ascencio e Edilene Aparecida V. de Campos, Editora Pearson Prentice Hall, 2002.

Algoritmo

Ao contrário do que se pensa, o conceito de **algoritmo** surgiu bem antes da invenção do computador. A palavra "algoritmo" tem origem no século IX, como matemático Al-Khowarizmi, que desenvolveu um conjunto de regras para efetuar operações aritméticas com números decimais.

Existem várias definições sobre o que é um algoritmo. Por exemplo:

"Algoritmo é uma sequência de passos que visa atingir um objetivo bem definido" [Forbellone, 1999]

"Algoritmo é a descrição de uma sequência de passos que deve ser seguida para realização de uma tarefa" [Ascencio, 1999]

Algoritmos são aplicados às mais diversas áreas e os utilizamos muitas vezes em nossas tarefas do cotidiano. Eles são úteis para descrever tarefas que devem sempre ser feitas da mesma maneira. O exemplo mais comum de algoritmo é uma receita de bolo. Nela são descritos quais **ingredientes** são necessários para a realização da tarefa de fazer um bolo e os **passos necessários** para transformar os ingredientes em um bolo de fato. Isso é muito parecido com um algoritmo, pois, normalmente descrevemos quais dados serão necessários para executar a tarefa e depois, descrevemos quais passos serão necessários para transformar os dados na resposta do problema para o qual o algoritmo está sendo construído. Por outro lado, uma receita de bolo é, na maioria das vezes, um algoritmo impreciso que usa expressões do tipo "sal a gosto" ou "uma pitada de pimenta", fazendo com que as receitas clássicas dificilmente se tornem **algoritmos computacionais** (programas que possam ser executados no computador).

De qualquer forma, muitas outras sequências de passos que nos deparamos no dia-a-dia podem ser consideradas um algoritmo. Por exemplo:

- Trocar o pneu furado de um carro
    - Identificar qual é o pneu furado.
    - Encontrar o estepe e o macaco.
    - Utilizar o macaco para levantar o carro do lado do pneu furado.
    - Soltar as porcas da roda do pneu furado.
    - Substituir o pneu furado pelo estepe.
    - Apertar as porcas.
    - Guardar o pneu furado e o macaco no carro.

- Somar três números
    - Obter os três números.
    - Somar os três números.
    - Mostrar o resultado obtido.

- Sacar dinheiro no banco 24 horas
    - Ir até um banco 24 horas.
    - Colocar o cartão.
    - Digitar a senha.
    - Solicitar a quantia desejada.
    - Se o saldo for maior ou igual à quantia desejada, sacar; caso contrário, saque impossibilitado.
    - Retirar o cartão.
    - Sair do banco 24 horas.

O que podemos observar nos exemplos de algoritmos citados é que **ele não é a solução para um problema, mas sim um caminho para obtê-la**. Existem vários caminhos diferentes para se obter a solução de um problema, o que significa que existem vários algoritmos diferentes para resolver o mesmo problema. Por exemplo, pelo menos dois métodos para multiplicar números com mais de um dígito são bastante conhecidos: o americano e o inglês.

- **Americano**: cada dígito do multiplicador deve multiplicar todo o multiplicando, da direita para a esquerda, escrevendo os resultados de cada dígito em uma linha, sendo que a última casa do resultado intermediário deve coincidir com a o dígito que está sendo usado para fazer a multiplicação. Após todos os dígitos serem multiplicados, todos os resultados intermediários devem ser somados, na ordem em que foram gerados.

- **Inglês**: cada dígito do multiplicador deve multiplicar todo o multiplicando, da esquerda para a direita, escrevendo os resultados de cada dígito em uma linha, sendo que a última casa do resultado intermediário deve coincidir com a o dígito que está sendo usado para fazer a multiplicação. Após todos os dígitos serem multiplicados, todos os resultados intermediários devem ser somados, na ordem em que foram gerados.

```
    981                    981
 X 1234                 X 1234
   3924                   981
  2943                    1962
 1962                      2943
+981_ __                +  3924
1210554                 1210554
```
(Americano) (Inglês)

Mas também existe uma terceira forma de se obter o resultado dessa multiplicação. Trata-se de um algoritmo *à la russe*.

- Escreva o multiplicando e o multiplicador lado a lado.

- Fazer duas colunas, uma para cada operando e repetir a seguinte regra: divida o multiplicando por dois, ignorando qualquer fração e multiplique o multiplicador por dois, até que o multiplicando seja igual a um.
- Some todos os valores do multiplicador, nas quais o multiplicando da mesma linha é ímpar.

981	**1234**	**1234**
490	2468	
245	**4936**	**4936**
122	9872	
61	**19744**	**19744**
30	39488	
15	**78976**	**78976**
7	**157952**	**157952**
3	**315904**	**315904**
1	**631808**	**+631808**
		1210554

Apesar de o método ser bem diferente dos métodos que já conhecemos, este exemplo é interessante para mostrar que podem existir caminhos bem diferentes para a mesma solução. São três formas diferentes de obter o resultado da multiplicação de dois números, três algoritmos diferentes para o mesmo objetivo.

Qualidades essenciais de um bom algoritmo:
- Clareza: as ações do algoritmo devem ser expressas de forma clara e objetiva.
- Eficácia: o algoritmo deve sempre chegar em um resultado.
- Eficiência: o algoritmo deve obter o resultado pretendido com a melhor relação custo/benefício.

Formas de representar um algoritmo

Um algoritmo pode ser representado de forma gráfica ou textual. As formas mais utilizadas são os algoritmos naturais, os fluxogramas, o pseudocódigo e é claro, as linguagens de programação.

Algoritmos Naturais ou Descrição Narrativa

Os algoritmos usados no nosso cotidiano normalmente são descritos em linguagem natural, como todos os algoritmos definidos no início desse material. Por exemplo, uma empresa pode utilizar um algoritmo para descrever um procedimento que deve ser feito por um funcionário sempre da mesma maneira.
Vantagem: não é necessário aprender nenhum conceito novo, pois a linguagem natural é bem conhecida.
Desvantagem: a linguagem natural está propensa a gerar ambiguidades e imprecisões, o que pode gerar dificuldades para transformar esse algoritmo em um programa. Por isso, esta forma de representação é pouco utilizada na prática de programação.

Exemplo 1) Algoritmo para calcular a média de um aluno.
- Obter as notas da primeira e da segunda prova.
- Somar as duas notas e dividir o resultado por dois.
- Se o resultado for maior do que 6 o aluno foi aprovado, caso contrário ele poderá fazer o exame.

Fluxogramas

É uma forma gráfica de representar algoritmos na qual cada forma geométrica representa um tipo de ação que deve ser realizada.
Vantagem: o entendimento dos elementos gráficos é mais simples do que o entendimento de textos.
Desvantagem: É necessário aprender a simbologia dos fluxogramas e o algoritmo resultante não apresenta muitos detalhes, o que pode dificultar a transcrição para um programa.

Mesmo assim, por ser uma representação gráfica e de fácil entendimento, ela é bastante utilizada para ajudar no entendimento da lógica de programação inicial, pois as seleções e repetições podem ser visualizadas mais facilmente do que na linguagem textual. Além disso, os fluxogramas ajudam no entendimento e refinamento do problema.

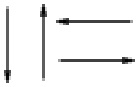
Elipses representam o início e o fim do fluxograma.

Retângulos com o canto superior esquerdo cortado representam operação de entrada de dados.

Retângulos com a base inferior arredondada representam operação de saída de dados.

Retângulos normais representam operações aritméticas e atribuições.

Losangos representam operações de comparações e decisões, dividindo o fluxo de dados sempre em duas partes.



Setas representam a direção do fluxo de processamento.

O Fluxograma da Figura 1 representa o algoritmo do cálculo da média. Um fluxograma tem sempre um ponto de início, que marca o começo do processamento e todos os fluxos de processamento devem terminar em um fim.

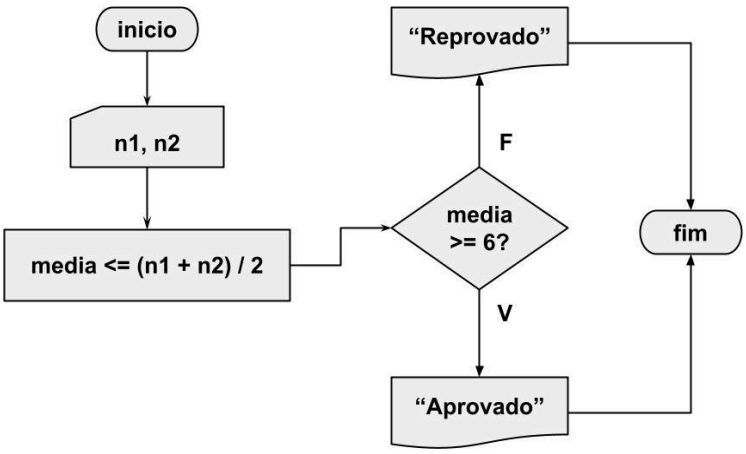


Figura 1: Fluxograma para o cálculo da média de dois números.

Exercício 1) Represente por meio de fluxogramas os seguintes algoritmos:

a) Calcular o valor de delta
   1. Obter os valores dos coeficientes a, b e c
   2. Calcular o valor de delta = $b^2 - 4ac$
   3. Escrever o valor de delta

b) Comparar dois números
   1. Obter os dois números
   2. Comparar os dois números para determinar qual das saídas deve ser executada
   3. Escrever 'os dois números são iguais' ou escrever 'o primeiro número é maior' ou escrever 'o segundo número é maior'

Pseudocódigo ou metalinguagem

É uma forma de representação baseada em linguagem natural, mas que tem regras específicas de sintaxe para se evitar ambiguidades e imprecisões.
Vantagens: a passagem para qualquer linguagem de programação é quase direta.
Desvantagens: é necessário aprender as regras específicas da linguagem.

A forma geral de representação de um algoritmo em Pseudocódigo:

```
algoritmo <<nome do algoritmo>>;
variáveis <<nome das variáveis separados por vírgula>> : <<tipo>>;
início
      <<comandos 1>>;
      <<comandos 2>>;
      ...
      <<comandos n>>;
fim
```

A palavra **"algoritmo"** marca o início do algoritmo. Em seguida deve vir o nome do algoritmo, de preferência um nome que representa o que o algoritmo faz. A palavra **"variáveis"** marca o início da sessão de declaração de variáveis. Em seguida deve vir o nome de uma variável. Se houver mais de uma variável com o mesmo tipo, os nomes devem ser separados por vírgula. As variáveis de mesmo tipo podem ser declaradas juntas. Neste caso, após os nomes das variáveis de mesmo tipo separados por vírgula, a lista é terminada por um sinal de dois pontos, seguido pelo nome do tipo das variáveis listadas e finalizado por um ponto e vírgula. Caso outras variáveis de tipos diferentes precisem ser declaradas, uma nova lista pode ser iniciada da mesma forma. Os tipos primitivos que uma variável pode assumir são: inteiro, real, caracter, cadeia de caracteres e lógico.

As palavras "**início**" e "**fim**" marcam o bloco de comandos do algoritmo em si, ou seja, a transformação dos dados de entrada na saída desejada. Os itens marcados com << e >> variam de acordo com o objetivo do algoritmo. Um comando pode ser um comando de leitura, de escrita, um cálculo matemático (que normalmente envolve a atribuição do valor do cálculo a uma variável), uma atribuição somente, uma

comparação ou um comando de repetição. Cada um desses comandos serão estudados no decorrer da disciplina.

Exemplo de algoritmo para o cálculo da média em pseudocódigo:

```
algoritmo calculo_media;
variáveis n1, n2, media : real;
início
      ler(n1, n2);
      media <- (n1 + n2)/2;
      se (media >= 6)
            então escrever('aprovado')
            senão escrever('exame');
fim.
```

Exemplo de algoritmo para o cálculo da média em Pascal:

```
program calculo_media;
var n1, n2, media : real;
begin
      read(n1, n2);
      media := (n1 + n2)/2;
      if (media >= 6)
            then write('aprovado')
            else write ('exame');
end.
```

Exemplo de algoritmo para o cálculo da média em C++:

```
#include<iostream>
using namespace std;

void main(){
  float n1, n2, media;
  cin >> n1 >> n2;
  media =(n1 + n2)/2;
  if (media >= 6)
    cout << "Aprovado com média: " << media;
    else cout << "Exame com média: " << media;
}
```

Exercício 2) Escreva os algoritmos do Exercício 1 (cálculo de delta e comparação de dois números) em pseudocódigo e também na linguagem que será usada na disciplina. Não se preocupe, esse exercício é só uma tentativa. Todos os comandos necessários para o desenvolvimento desse tipo de algoritmo serão estudados cuidadosamente nas próximas aulas.