

Algoritmos

MATA37: Introdução à Lógica de Programação

Prof.: Rafael A. Melo (melo@dcc.ufba.br)

Departamento de Ciência da Computação

Instituto de Matemática

Universidade Federal da Bahia



Quatro princípios de Pólya para resolução de problemas

- Descritos por George Polya (Pólya György)
- **Princípio 1:** Entender o problema
 - Você entendeu realmente o problema?
 - Quais são os dados disponíveis?
 - Pode-se resolver o problema com o que foi dado?
 - Desenhar uma figura pode te ajudar?

Quatro princípios de Pólya para resolução de problemas

- **Princípio 2:** Elaborar um plano
 - Encontrar uma conexão entre os dados e o que se quer encontrar
 - Você já viu este problema antes? Ou este problema de outra forma?
 - Você conhece um problema relacionado o qual você sabe como resolver?

Quatro princípios de Pólya para resolução de problemas

- **Princípio 3:** Colocar o plano em prática
 - Execute o plano
 - Você consegue verificar que cada passo funciona?
- **Princípio 4:** Analisar o seu trabalho
 - Examinar a solução obtida
 - Você pode verificar o resultado?
 - Você pode encontrar a solução de maneira diferente?

Princípios de Pólyak para a construção de um programa

- Definir requisitos do problema (resolver o problema certo)
 - Entradas
 - Cálculos
 - Saída
- Desenvolvimento do algoritmo (resolver certo o problema)
 - Portugol
 - Pseudocódigo
 - Fluxograma
- Codificação do programa
 - Utilizando uma linguagem de programação.
- Teste do programa
 - Defeito na escrita do algoritmo
 - Defeito no algoritmo

Algoritmos: o que são?

- O que é um **Algoritmo**?
 - Definição informal: é uma receita para resolver um problema
 - ou seja, uma sequência de passos que visa atingir um objetivo bem definido.
- Na nossa vida cotidiana utilizamos algoritmos em diversas circunstâncias
 - Receita de bolo;
 - Resolução de uma equação do segundo grau;
 - Montagem de um equipamento com Manual de Instruções;
 - Instruções para uso de um medicamento.

Problema: Preparar bifes à milanesa

- Exemplo:
 - Problema: preparar “bifes à milanesa”
 - Algoritmo: descrição da receita
- 1: Qual é realmente o problema?
- 2: Quais são as entradas?
- 3: Quais os materiais necessários?
- 4: Qual o resultado esperado?

Problema: Preparar bifes à milanesa

- Objetos de “consumo” (entrada):
 - carne
 - farinha
 - ovos
 - alface
- Objetos de “apoio” (atores, executores):
 - faca: instrução de cortar
 - travessa: instrução de armazenar
 - fogão: instrução de esquentar
 - cozinheiro: instrução de preparar

Problema: Preparar bifes à milanesa

- Objetos “produzidos” (saída):
 - bifes à milanesa
- Objeto que “controla” o processo (receita):
 - algoritmo

Algoritmo para preparar um bife à milanesa

BIFE À MILANESA

1. Limpar a peça de carne
2. Fatiar a carne em bifos
3. Colocar farinha de rosca em um prato
4. Bater 2 ovos em outro prato
5. Repetir, para cada bife:
 - 5.1. passar cada lado do bife nos ovos;
 - 5.2. passar cada lado do bife na mistura de farinha;
 - 5.3. levar o bife à frigideira;
 - 5.4. aguardar dourar, virando ambas as faces;
 - 5.5. retirar bife e colocar sobre papel toalha até secar;
 - 5.6. retirar do papel toalha e juntar numa travessa;
6. Decorar a travessa com folhas de alface
7. Servir

Algoritmo

ENTRADA

Ingredientes



ESTADO
INICIAL



SEQUÊNCIA
FINITA DE
AÇÕES BEM
DEFINIDAS

SAÍDA



ESTADO
FINAL



Algoritmo

Um pouco de história

- Um dos primeiros algoritmos:
 - Euclides (300 . . . 400 BC): algoritmo para obter o máximo divisor comum de dois inteiros positivos
- Século IX (800-899 DC), península arábica/Pérsia:
 - Matemático Mohammed al-Khowârizmî
 - Cria regras passo-a-passo para se fazer aritmética com algarismos decimais
- Origem do nome:
 - Al-Khowârizmî, em latim tornou-se **algorismus**
 - Do latim algorismus, derivou-se **algoritmo** em português

Algoritmo

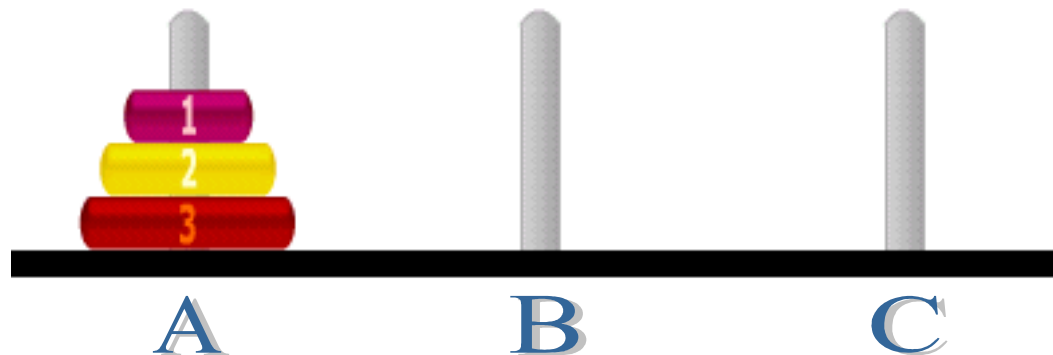
- Definição mais formal de Algoritmo: conjunto de regras e operações **bem definidas** e **ordenadas**, destinadas à solução de um problema ou de uma **classe de problemas**, em um **número finito de etapas**.
- Segundo Dijkstra, um algoritmo corresponde a uma descrição de um **padrão de comportamento**, expresso em termos de um conjunto finito de ações.
 - Considere um algoritmo para somar dois números a e b .
 - percebe-se um padrão de comportamento que se repete, independente de quais sejam esses números.
 - O algoritmo é destinado então à **classe de problemas** “adição de dois números”.

Propriedades de um algoritmo

- Formado por um texto finito;
- Instruções elementares
 - Ações simples, claras e bem definidas (não ambíguas) que possam ser realizadas;
- Texto metódico (sequência ordenada de ações)
 - Passo inicial
 - Passo final
 - Dado o passo atual, sabe-se qual o próximo passo
- É correto
 - Partindo de dados válidos, deve sempre terminar.
 - Partindo de dados inválidos, não necessariamente termina.

Exercício 1: Torre de Hanói

- Escreva um algoritmo para transferir os discos da haste A para a haste C usando a haste B como auxiliar.
- Restrições:
 - Apenas um disco pode ser movido por vez;
 - Um disco não pode ser colocado sobre outro disco menor do que ele.








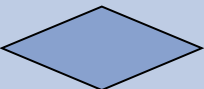
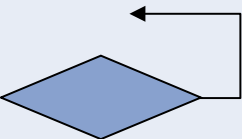
Representação de Algoritmos

- Descrição Narrativa
 - Especificação textual dos passos em linguagem natural.
 - Desvantagens: a linguagem natural é prolixa e imprecisa e freqüentemente pouco confiável como um veículo de transferir informação.
- Fluxograma
 - Um fluxograma mostra, de forma gráfica, a lógica de um algoritmo, enfatizando passos individuais e o fluxo de execução.
 - Desvantagem: pouco escalável.

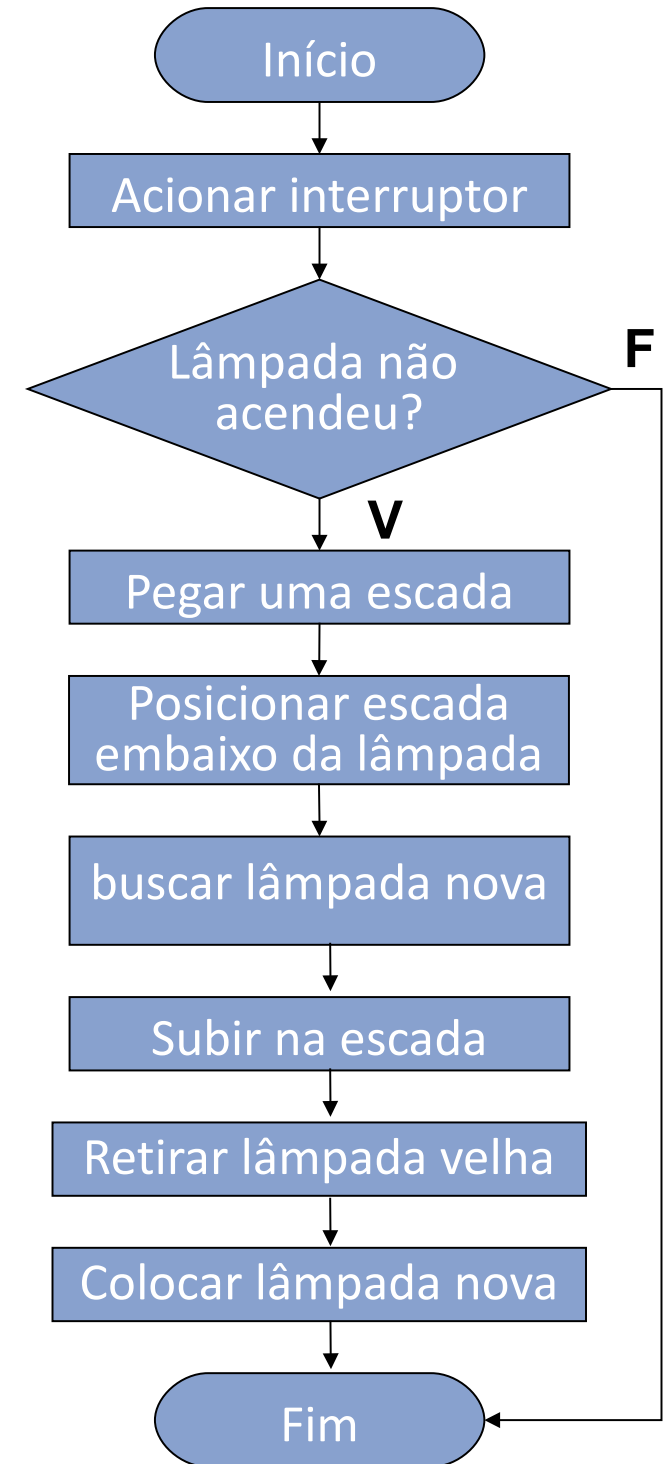
Representação de Algoritmos

- Pseudocódigo ou Linguagem Algorítmica ou Pseudolinguagem
 - Linguagem especial para expressão de algoritmos.
 - Funciona como uma “linguagem simplificada de programação”.
 - Utiliza expressões concisas e pré-definidas.
 - Descrição textual, estruturada e regida por regras.
 - Utiliza-se palavras-chaves, endentação, apenas um passo por linha.
- Vamos usar o Portugol
 - Estrutura similar à do Pascal.

Fluxograma

Símbolo	Nome	Função
	Terminador	Representar a saída para ou entrada do ambiente externo, por exemplo, início ou final de programa.
	Processo	Representar qualquer tipo de processo, geralmente utilizado para definir cada passo sequencial do algoritmo.
	Linha básica	Representar o fluxo dos dados ou controle. Pode-se usar pontas de setas para indicar a direção do fluxo.
	Entrada manual	Representar os dados que sejam fornecidos manualmente em tempo de processamento.
	Exibicao	Representar dados que devem ser exibidos para uso humano, como em um monitor ou impressora.
	Decisao	Representar uma decisão ou um desvio de fluxo.
	Repeticao	Uma decisão combinada com um fluxo (linha básica) de retorno.

Fluxograma – Algoritmo troca de lâmpada



Pseudocódigo – Algoritmo troca de lâmpada

Início

```
    Ir até o interruptor do primeiro soquete;  
    enquanto <houver soquetes> faça  
        acionar interruptor;  
        se <lâmpada não acender> então  
            pegar uma escada;  
            posicionar a escada embaixo da lâmpada;  
            buscar uma lâmpada nova;  
            subir na escada;  
            retirar a lâmpada velha;  
            colocar a lâmpada nova;  
            ir até interruptor do próximo soquete;  
        fim se  
    fim enquanto
```

Fim

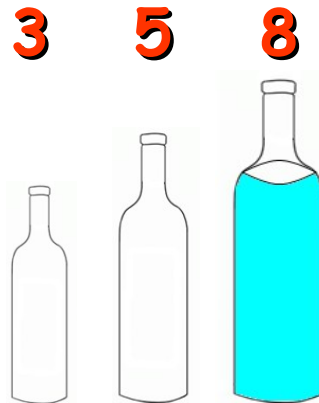
Exercício 2: A raposa, as galinhas e o milho

- Um senhor está numa das margens de um rio com uma raposa, algumas galinhas e um saco de milho. Ele pretende atravessar o rio com suas cargas, num barco que só comporta ele mesmo e uma das cargas. Evidentemente, ele não pode deixar em uma das margens, sozinhos, a raposa e a galinha, nem a galinha e o milho. Escreva um algoritmo que oriente o senhor a realizar o seu intento.



Exercício 3: Problema das garrafas

- Escreva um algoritmo para separar o líquido de três garrafas com formatos diferentes e não milimetradas em duas quantidades iguais, sendo que:
 - Uma garrafa está cheia até a boca, com 8 litros
 - Uma está vazia, com capacidade de 5 litros
 - Uma está vazia, com capacidade de 3 litros



Exercício 4: Problema da moeda falsa

- Escreva um algoritmo para descobrir a moeda falsa (mais leve) de um total de 5 moedas usando uma balança analítica
- Dica: é possível resolver com somente duas pesagens



Algoritmos

MATA37: Introdução à Lógica de Programação

Prof.: Rafael A. Melo (melo@dcc.ufba.br)

Departamento de Ciência da Computação

Instituto de Matemática

Universidade Federal da Bahia

