

Processamento de Dados aplicado à Geociências

AULA 2: Introdução aos algoritmos

UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSO SUPERIOR DE TECNOLOGIA EM
GEOPROCESSAMENTO

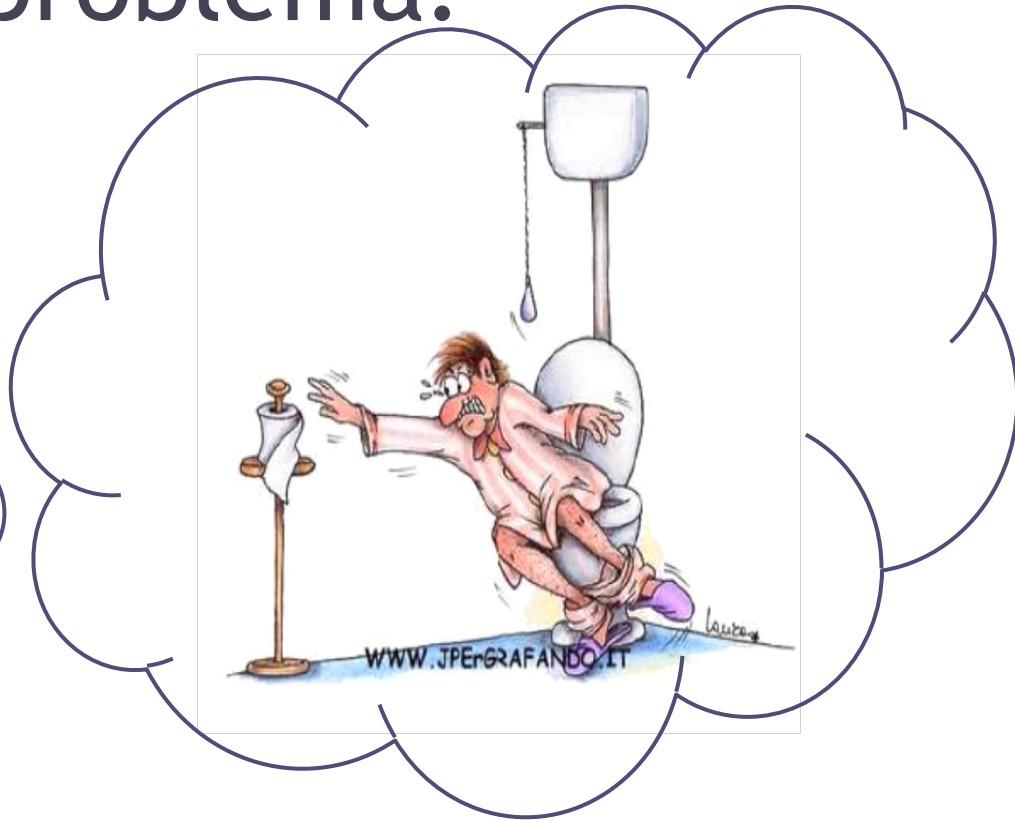
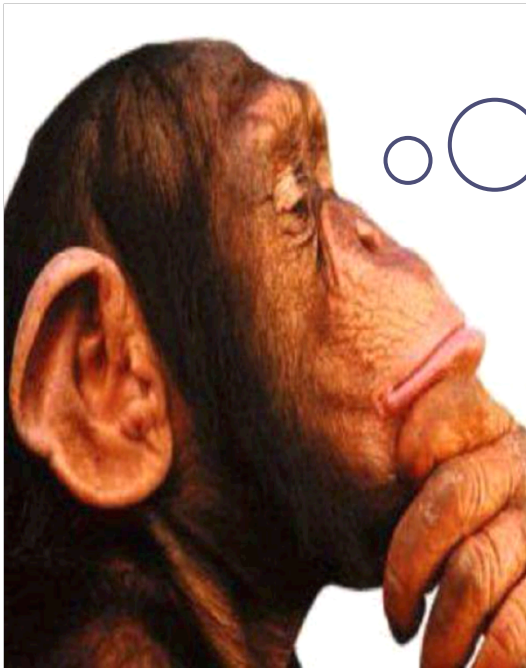
Professor: Guilherme Tomaschewski Netto
guilherme.netto@inf.ufpel.edu.br



Introdução aos algoritmos



O que é um problema?



Definição de problema

- **Problema (Dicionário Michaelis):**
 - Questão matemática proposta para ser resolvida.
 - Questão difícil, delicada, suscetível a diversas soluções.
 - Qualquer coisa de difícil explicação; mistério, enigma.
 - Dúvida, questão.

Definição de problema

- **Problema (na Ciência da Computação):**
 - É uma descrição geral de um conjunto de dados de entrada e um conjunto de requisitos que uma solução satisfatória deve ter.

Solucionando um problema...

- Para solucionar com precisão problemas computacionais devemos ter em mente os seguintes passos:
 - Interpretar o problema proposto
 - Identificar os dados disponíveis para a sua resolução
 - Definir seu objetivo. Pergunte-se: “Qual o resultado que se deseja alcançar?”
 - Definir passos necessários para atingir o objetivo

O que é um algoritmo?

- Descreve um modo de solucionar um problema.
- Faz uso dos dados disponíveis para executar algumas ações que levam a solução do problema.

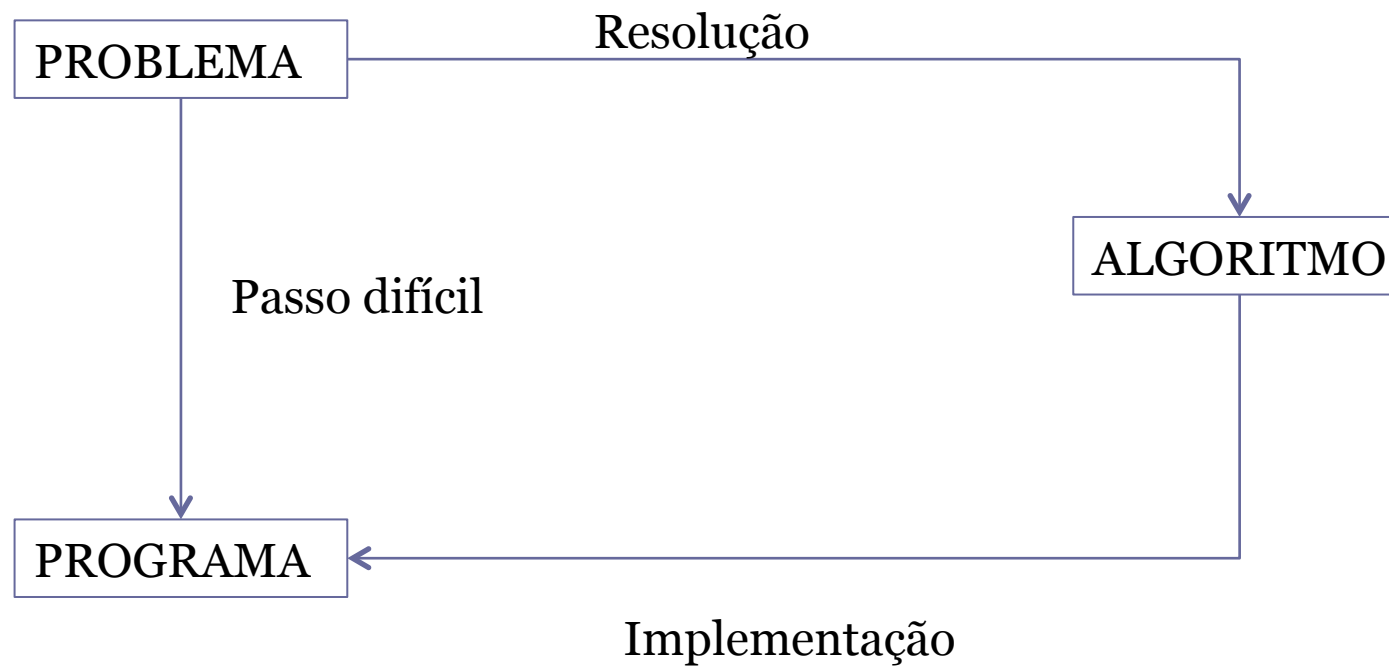
Definição de Algoritmo

- Um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como **entrada** e produz algum valor ou conjunto de valores como **saída**.
- Portanto, um algoritmo é uma sequência de passos computacionais que transformam a entrada na saída.

Algoritmo

- O entendimento da estrutura, do desenvolvimento e da construção de algoritmos é de fundamental importância para o aprendizado das mais diversas linguagens de programação.
- Um algoritmo pode ser visto como um modelo intermediário entre a linguagem humana e a linguagem de programação.

Algoritmo



Algoritmo

- Um algoritmo bem definido deve:
 - 1. Ter início e fim
 - 2. Ser descrito em termos de ações não ambíguas e bem definidas
 - 3. Que as ações sigam uma sequência ordenada

Algoritmo

- Utilizamos constantemente algoritmos em nosso cotidiano:
 - Instruções para se utilizar um aparelho eletrodoméstico
 - Uma receita culinária
 - Cálculo de orçamento

Exemplo

- Algoritmo para fritar um ovo
 - 1. Colocar um ovo na frigideira;
 - 2. Esperar o ovo ficar frito;
 - 3. Tirar o ovo.

Exemplo

- Versão detalhada
 - 1. Retirar um ovo da geladeira;
 - 2. Colocar a frigideira no fogo;
 - 3. Colocar óleo;
 - 4. Esperar até o óleo ficar quente;
 - 5. Quebrar o ovo separando a casca;
 - 6. Colocar o conteúdo na frigideira;
 - 7. Esperar um minuto;
 - 8. Retirar o ovo da frigideira;
 - 9. Apagar o fogo.

Exemplo

- Problema: Troca de uma lâmpada
- Dados: Lâmpada velha, lâmpada nova, escada
- Objetivo: Lâmpada deve acender

Exemplo

- Podemos descrever os seguintes passos para solucionar o problema de trocar uma lâmpada:
 1. Pegar uma escada;
 2. Posicionar a escada embaixo da lâmpada;
 3. Buscar uma lâmpada nova;
 4. Subir na escada;
 5. Retirar a lâmpada velha;
 6. Colocar a lâmpada nova;
 7. Descer da escada;
 8. Guardar a escada.

Exemplo

- Podemos descrever os seguintes passos para solucionar o problema de trocar uma lâmpada:
 1. Pegar uma escada;
 2. Posicionar a escada embaixo da lâmpada;
 3. Buscar uma lâmpada nova;
 4. Subir na escada;
 5. Retirar a lâmpada velha;
 6. Colocar a lâmpada nova;
 7. Descer da escada;
 8. Guardar a escada.
- E se a lâmpada não estiver queimada?

Refinando o Exemplo...

- Troca de lâmpada com teste
 1. Pegar uma escada;
 2. Posicionar a escada embaixo da lâmpada;
 3. Buscar uma lâmpada nova;
 4. Acionar o interruptor;
 5. Se a lâmpada não acender, então
 - a) Subir na escada;
 - b) Retirar a lâmpada queimada;
 - c) Colocar a lâmpada nova;
 - d) Descer da escada;
 6. Guardar a escada.

Refinando o Exemplo...

- Troca de lâmpada com teste
 1. Pegar uma escada;
 2. Posicionar a escada embaixo da lâmpada;
 3. Buscar uma lâmpada nova;
 4. Acionar o interruptor;
 5. Se a lâmpada não acender, então
 - a) Subir na escada;
 - b) Retirar a lâmpada queimada;
 - c) Colocar a lâmpada nova;
 - d) Descer da escada;
 6. Guardar a escada.
- Está correto, mas podemos melhorar... Como?

Refinando o Exemplo...

- Troca de lâmpada com teste no início
 1. Acionar o interruptor;
 2. Se a lâmpada não acender, então
 - a) Pegar um escada;
 - b) Posicionar a escada embaixo da lâmpada;
 - c) Buscar uma lâmpada nova;
 - d) Subir na escada;
 - e) Retirar a lâmpada queimada;
 - f) Colocar a lâmpada nova;
 - g) Descer da escada;
 - h) Guardar a escada.

Refinando o Exemplo...

- Troca de lâmpada com teste no início
 1. Acionar o interruptor;
 2. Se a lâmpada não acender, então
 - a) Pegar um escada;
 - b) Posicionar a escada embaixo da lâmpada;
 - c) Buscar uma lâmpada nova;
 - d) Subir na escada;
 - e) Retirar a lâmpada queimada;
 - f) Colocar a lâmpada nova;
 - g) Descer da escada;
 - h) Guardar a escada.
- E se a lâmpada nova também estiver queimada?

Refinando o Exemplo...

- Troca de lâmpada com teste e repetição
 1. Acionar o interruptor;
 2. Se a lâmpada não acender, então
 - a) Pegar um escada;
 - b) Posicionar a escada embaixo da lâmpada;
 - c) Buscar uma lâmpada nova;
 - d) Subir na escada;
 - e) Retirar a lâmpada queimada;
 - f) Colocar a lâmpada nova;
 - g) Enquanto a lâmpada não acender, faça:
 - Desce da escada;
 - Buscar uma lâmpada nova;
 - Subir na escada;
 - Retirar a lâmpada queimada;
 - Colocar uma lâmpada nova;
 - h) Descer da escada;
 - i) Guardar a escada.

Nível de abstração

- Um algoritmo pode ser detalhado em diferentes níveis de abstração de acordo com a necessidade do problema ou da linguagem de programação em questão.
- Como um algoritmo é uma abstração de um problema real, devem ser consideradas somente as propriedades relevantes à resolução do problema.

Fatores a serem levados em consideração na construção de um algoritmo

- 1.Complexidade
 - A complexidade está na quantidade de situações diferentes que um problema pode apresentar, as quais devem ser previstas na solução.
 - Uma interpretação errônea de um problema pode levar a soluções irrelevantes, gerando algoritmos mais complexos do que o necessário.

Fatores a serem levados em consideração na construção de um algoritmo

- 2. Legibilidade
 - Mede a clareza com que a lógica está exposta, um algoritmo legível é aquele que é compreendido por qualquer observador.
 - Quanto mais legível for um algoritmo, menor será sua complexidade.

Fatores a serem levados em consideração na construção de um algoritmo

- 3. Portabilidade
 - Utiliza-se comumente uma pseudo-linguagem (português estruturado) para que os algoritmos possam ser convertidos facilmente para qualquer linguagem de programação usual, de acordo com a aplicação do problema.

Fatores a serem levados em consideração na construção de um algoritmo

- 4. Técnica do método cartesiano
 - “Dividir para conquistar”, Descartes.
 - Consiste em dividir um problema grande, de difícil solução, em problemas menores, de solução mais fácil, repetindo o método se necessário.
 - 1. Dividir o problema em partes
 - 2. Analisar a divisão e garantir a coerência entre as partes.
 - 3. Reaplicar o método, se necessário

Fatores a serem levados em consideração na construção de um algoritmo

- 5. Planejamento reverso
 - Consiste em, a partir do resultado final, determinar quais são os componentes básicos. Ou seja, a partir da saída desejada, devemos poder determinar, reversamente, quais são os componentes da entrada de dados necessários.

Algoritmo

- Algoritmo não é a solução do problema, pois, se assim fosse, cada problema teria um único algoritmo
- Algoritmo é um caminho para a solução de um problema:
 - Em geral, os caminhos que levam a uma solução são muitos

Método para construir um algoritmo

- 1. Ler atentamente o enunciado
 - Deve-se reler o enunciado de um exercício quantas vezes for necessário, até compreendê-lo completamente. A maior parte da resolução de um exercício consiste na compreensão completa do enunciado.

Método para construir um algoritmo

- 2. Retirar a relação das entradas de dados do enunciado
 - Deve-se descobrir através do enunciado, quais são os dados que devem ser fornecidos ao programa, via leitura, a partir dos quais são desenvolvidos o problema.
 - Pode haver algum algoritmo que não necessite da entrada de dados (pouco comum).

Método para construir um algoritmo

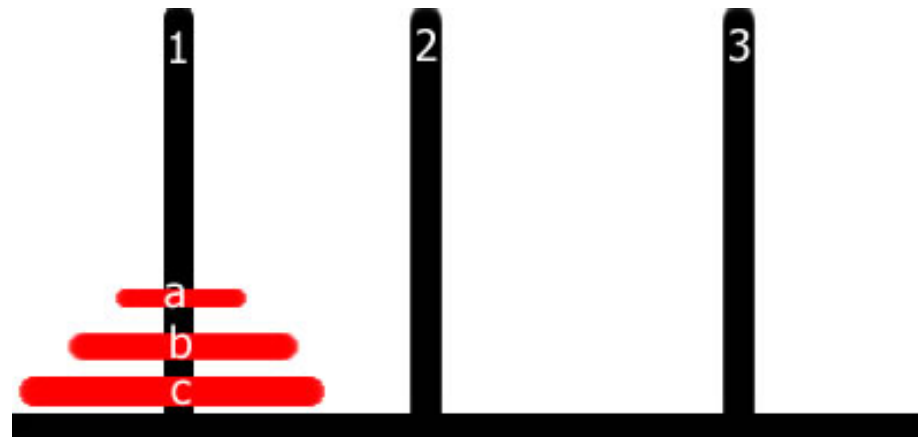
- 3. Retirar do enunciado, a relação das saídas das informações
 - Através do enunciado deve-se descobrir quais são as informações que devem ser mostradas para compor o resultado final, objetivo do algoritmo.

Método para construir um algoritmo

- 4. Determinar o que deve ser feito para transformar as entradas nas saídas especificadas
 - Nessa fase há a construção do Algoritmo propriamente dito. Deve-se determinar qual sequência de passos ou ações é capaz de transformar um conjunto de dados nas informações de resultado.
 - Para isso, recomenda-se a utilização dos fatores descritos anteriormente, tais como legibilidade, portabilidade, método cartesiano e planejamento reverso, para finalmente construir o algoritmo.

Exercício 1

- Elabore um algoritmo que mova três discos de uma Torre de Hanói, que consiste em três hastes (1-2-3), uma das quais serve de suporte para três discos de tamanhos diferentes (a-b-c), os menores sobre os maiores. Pode-se mover um disco de cada vez para qualquer haste, contanto que nunca seja colocado um disco maior sobre um menor. O objetivo é transferir os três discos para outra haste.



Exercício 2

- Um homem precisa atravessar um rio com um barco que possui capacidade apenas para carregar ele mesmo e mais uma de suas três cargas, que são: um lobo, uma ovelha e uma caixa de repolho. Lembrando que o lobo come a ovelha e a ovelha come o repolho, o homem não poderá deixar a sós em uma margem nenhuma das duplas lobo&ovelha ou ovelha&repolho. O que o homem deve fazer para conseguir atravessar o rio sem perder suas cargas? Escreva um algoritmo mostrando a resposta, ou seja, indicando todas as ações necessárias para efetuar uma travessia segura.
 - <http://www.plastelina.net/games/game1.html>