

9002 — Aula 02

Algoritmos e Programação de Computadores

Instituto de Engenharia – UFMT

Segundo Semestre de 2014

22 de setembro de 2014

Agenda

- 1 Introdução
- 2 Modelo Computacional
- 3 Algoritmos
 - Definição
 - Caracterização
- 4 Algoritmos e Modelos

Resolvendo Problemas

O tópico central deste curso é a **resolução de problemas** computacionais.

Resolvendo Problemas

Algumas questões importantes:

- Quais são os tipos de problemas que podem ser resolvidos computacionalmente? Todos?
- O que significa resolver um problema computacionalmente?

Resolvendo Problemas

Algumas questões importantes:

- Quais são os tipos de problemas que podem ser resolvidos computacionalmente? Todos?
- O que significa resolver um problema computacionalmente?

Resolvendo Problemas

Algumas questões importantes:

- Quais são os tipos de problemas que podem ser resolvidos computacionalmente? Todos?
- O que significa resolver um problema computacionalmente?

Resolvendo Problemas

Algumas questões importantes:

- Quais são os tipos de problemas que podem ser resolvidos computacionalmente? Todos?
- O que significa resolver um problema computacionalmente?

Computador?

O que é um *computador*?



Qual computador?

Computador?

O que é um *computador*?



Qual computador?

Modelagem

Modelos são importantes porque capturam o espírito de fenômenos e objetos estudados.

Modelagem

Modelos

- Capturam a **essência** de um determinado objeto.
- Em geral, não possuem a mesma escala do objeto.
- Omitem alguns **detalhes** do objeto.
- São limitados.

Modelo Computacional

Objeto que:

- recebe dados (entrada);
- armazena e recupera dados (memória);
- realiza operações de acordo com conjuntos de instruções (processamento) e
- produz um resultado (saída).

Modelo Computacional

Como escolher as instruções?

Os tipos de problemas que podem ser resolvidos computacionalmente dependem do conjunto de instruções escolhido?

Modelo Computacional

Como escolher as instruções?

Os tipos de problemas que podem ser resolvidos computacionalmente dependem do conjunto de instruções escolhido?

Modelo Computacional

Para responder estas perguntas, vamos precisar primeiro entender como interagir com este modelo. Esta interação é realizada utilizando um **algoritmo**. Vamos lá!!!

Modelo Computacional

Para responder estas perguntas, vamos precisar primeiro entender como interagir com este modelo. Esta interação é realizada utilizando um **algoritmo**. Vamos lá!!!

Algoritmos: definição

- Uma sequência **finita** de passos, **precisos** e **bem definidos**, para a realização de uma tarefa, em que detalhes e formalismos dependem do 'leitor'.
- **Receita** para resolver um determinado **problema**.

Exemplo de algoritmo

como você descreveria os passos necessários para preparar um macarrão instantâneo?

Algoritmos: definição

- Uma sequência **finita** de passos, **precisos** e **bem definidos**, para a realização de uma tarefa, em que detalhes e formalismos dependem do 'leitor'.
- **Receita** para resolver um determinado **problema**.

Exemplo de algoritmo

como você descreveria os passos necessários para preparar um macarrão instantâneo?

Algoritmo do macarrão instantâneo:

- ➊ ir até a cozinha.
- ➋ procurar pelas embalagens de macarrão.
- ➌ repetir, para cada embalagem:
 - ➊ ferver 2 copos de água em uma panela.
 - ➋ abrir a embalagem.
 - ➌ colocar o macarrão na panela e cozinhar por 3 minutos.
 - ➍ mexer para soltar os fios do macarrão.
 - ➎ retirar do fogo, deixar escorrer a água e juntar em uma travessa.
- ➍ acrescentar azeite e sal na travessa.
- ➎ servir.

Algoritmos: definição

Objetos de entrada:

- macarrão instantâneo.
- azeite.
- sal.

Objetos auxiliares:

- panela.
- travessa.
- fogão.
- cozinheiro.

Objeto de saída:

- macarrão cozido.

Objeto de controle:

- receita (algoritmo).

Algoritmos: definição

Exemplo de algoritmo

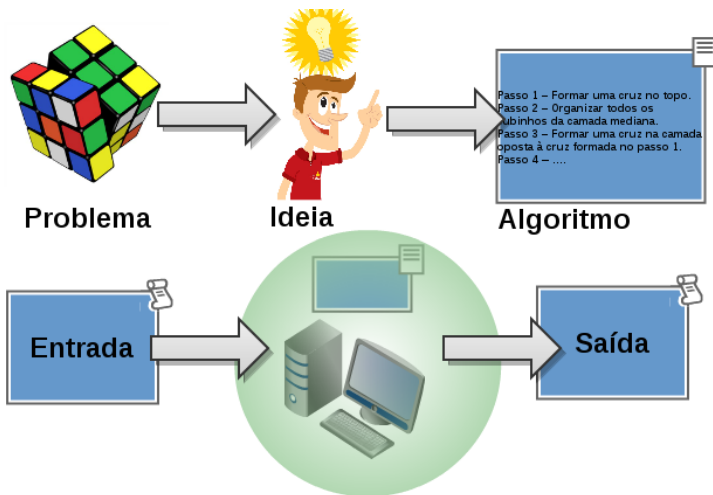
Como você descreveria este mesmo algoritmo para náufrago? Quais são os objetos de entrada, auxiliares e saída, por exemplo?

Algoritmos: definição

Exemplo de algoritmo

Como você descreveria este mesmo algoritmo para náufrago? Quais são os objetos de entrada, auxiliares e saída, por exemplo?

Algoritmos: definição



create and share your own diagrams at gliffy.com



Algoritmos: caracterização

Recapitulando...

- Sequência **finita** de passos, **precisos** e **bem definidos**, para a resolução de um **problema**.

Isto implica que:

- 1 o número de passos é finito.
- 2 o texto deve ser uma receita metódica, passo-a-passo, que inclui um passo inicial, um passo final e o passo seguinte em qualquer instante da execução.
- 3 cada passo contém uma instrução **elementar** para o leitor. Por exemplo,

Algoritmos: caracterização

Recapitulando...

- Sequência **finita** de passos, **precisos** e **bem definidos**, para a resolução de um **problema**.

Isto implica que:

- 1 o número de passos é finito.
- 2 o texto deve ser uma receita metódica, passo-a-passo, que inclui um passo inicial, um passo final e o passo seguinte em qualquer instante da execução.
- 3 cada passo contém uma instrução **elementar** para o leitor. Por exemplo,

“... juntar dois ovos ...” é elementar para um cozinheiro

Algoritmos: caracterização

Recapitulando...

- Sequência **finita** de passos, **precisos** e **bem definidos**, para a resolução de um **problema**.

Isto implica que:

- 1 o número de passos é finito.
- 2 o texto deve ser uma receita metódica, passo-a-passo, que inclui um passo inicial, um passo final e o passo seguinte em qualquer instante da execução.
- 3 cada passo contém uma instrução **elementar** para o leitor. Por exemplo,

“ ... se hoje você puder provar que a cotação do dólar vai subir 10% no próximo mês, compre \$ 1.000,00 ...” não é elementar para muitas pessoas :)

Exercício

Exercício

Considere a seguinte situação hipotética. Há três pequenas caixas lacradas em uma sala. Você sabe que elas estão cheias de ouro em pó, mas não conhece a quantidade exata de cada uma delas.

- Como você faria para ordenar as tais caixas de acordo com sua quantidade de ouro utilizando uma balança de pratos?
- Quantos vezes a balança foi utilizada?

De volta ao computador...

Quais é o conjunto de instruções de nosso *computador*? Sobretudo

- operações aritméticas,
- comparações e
- repetições.

Um exemplo com números...

Problema:

Encontrar máximo divisor comum ou MDC entre dois números inteiros positivos. Lembre-se de que MDC é o maior número inteiro que é fator de ambos os números. Assim, por exemplo, como os divisores comuns entre 12 e 18 são 1,2,3 e 6, o MDC deles é igual à 6.

Ideia:

??????

Exemplo

Um exemplo com números...

Ideia

Utilizar o algoritmo de Euclides, como nos tempos do colégio :)

Mas, antes, uma pergunta:

Quais são os nossos objetos auxiliares, de entrada e saída para este problema?

Algoritmos: definição

Um exemplo com números...

Objetos de entrada:

- dois números inteiros positivos M e N .

Objetos auxiliares:

- lápis, papel, borracha e **você**.

Objeto de saída:

- um número inteiro positivo Z com o resultado do MDC de M e N .

Algoritmos: definição

Um exemplo com números...

Algoritmo de Euclides

- ① Se M é igual N , então MDC, Z , é M (ou N); pare!
- ② Caso contrário:
 - ① se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 1
 - ② se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 1

E aí, funciona?

Como testamos isso?

Simulando o código

Vamos recordar o nosso modelo computacional...

Por que simular?

Simulação

- Eventualmente, simplesmente olhar o código pode não trazer a tona o erro.
- Por isso, utiliza-se uma técnica de simulação do código
 - Pode ser automatizada (utilizando um *debugger*)
 - Pode ser feita manualmente, utilizando papel e caneta (**teste de mesa**).

Por que simular?

Simulação

- Eventualmente, simplesmente olhar o código pode não trazer a tona o erro.
- Por isso, utiliza-se uma técnica de simulação do código
 - ▶ Pode ser automatizada (utilizando um *debugger*)
 - ▶ Pode ser feita manualmente, utilizando papel e caneta (**teste de mesa**).

Por que simular?

Simulação

- Eventualmente, simplesmente olhar o código pode não trazer a tona o erro.
- Por isso, utiliza-se uma técnica de simulação do código
 - ▶ Pode ser automatizada (utilizando um *debugger*)
 - ▶ Pode ser feita manualmente, utilizando papel e caneta (**teste de mesa**).

Simulação Manual

Simulando código

- Bem simples: Existem apenas 2 passos.
 - ▶ “Reservar” os espaços para os nossos objetos
 - ▶ “Executar” em sequência cada um dos passos do algoritmo.

Simulando

Algoritmo de Euclides

- ❶ Defina M igual à 10 e N igual à 6.
- ❷ Se M é igual N , então MDC , Z , é M (ou N); pare!
- ❸ Caso contrário:
 - ❶ se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ❷ se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Simulando

Algoritmo de Euclides

- ① Defina M igual à 10 e N igual à 6.
- ② Se M é igual N , então MDC , Z , é M (ou N); pare!
- ③ Caso contrário:
 - ① se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ② se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Reservando espaço para os objetos

Simulando

Algoritmo de Euclides

- ❶ Defina M igual à 10 e N igual à 6.
- ❷ Se M é igual N , então MDC , Z , é M (ou N); pare!
- ❸ Caso contrário:
 - ❶ se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ❷ se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Reservando espaço para os objetos

Após executar a **linha 1**.

| Tipo | inteiro positivo | inteiro positivo | inteiro positivo |
|-------|------------------|------------------|------------------|
| Nome | M | N | Z |
| Valor | 10 | 6 | ? |

Simulando

Algoritmo de Euclides

- ❶ Defina M igual à 10 e N igual à 6.
- ❷ Se M é igual N , então MDC , Z , é M (ou N); pare!
- ❸ Caso contrário:
 - ❶ se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ❷ se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Reservando espaço para os objetos

Após executar a **linha 3.1**.

| Tipo | inteiro positivo | inteiro positivo | inteiro positivo |
|-------|------------------|------------------|------------------|
| Nome | M | N | Z |
| Valor | 4 | 6 | ? |

Simulando

Algoritmo de Euclides

- ① Defina M igual à 10 e N igual à 6.
- ② Se M é igual N , então MDC , Z , é M (ou N); pare!
- ③ Caso contrário:
 - ① se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ② se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Execução

Simulando

Algoritmo de Euclides

- ❶ Defina M igual à 10 e N igual à 6.
- ❷ Se M é igual N , então MDC , Z , é M (ou N); pare!
- ❸ Caso contrário:
 - ❶ se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ❷ se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Execução

Após executar a **linha 3.2**.

| Tipo | inteiro positivo | inteiro positivo | inteiro positivo |
|-------|------------------|------------------|------------------|
| Nome | M | N | Z |
| Valor | 4 | 2 | ? |

Simulando

Algoritmo de Euclides

- ❶ Defina M igual à 10 e N igual à 6.
- ❷ Se M é igual N , então MDC , Z , é M (ou N); pare!
- ❸ Caso contrário:
 - ❶ se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ❷ se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Execução

Após executar a **linha 3.1**.

| Tipo | inteiro positivo | inteiro positivo | inteiro positivo |
|-------|------------------|------------------|------------------|
| Nome | M | N | Z |
| Valor | 2 | 2 | ? |

Simulando

Algoritmo de Euclides

- ❶ Defina M igual à 10 e N igual à 6.
- ❷ Se M é igual N , então MDC , Z , é M (ou N); pare!
- ❸ Caso contrário:
 - ❶ se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ❷ se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Execução

Após executar a **linha 2**.

| Tipo | inteiro positivo | inteiro positivo | inteiro positivo |
|-------|------------------|------------------|------------------|
| Nome | M | N | Z |
| Valor | 2 | 2 | 2 |

Simulando

Algoritmo de Euclides

- ❶ Defina M igual à 10 e N igual à 6.
- ❷ Se M é igual N , então MDC , Z , é M (ou N); pare!
- ❸ Caso contrário:
 - ❶ se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ❷ se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Execução

Opa! Terminamos...

| Tipo | inteiro positivo | inteiro positivo | inteiro positivo |
|-------|------------------|------------------|------------------|
| Nome | M | N | Z |
| Valor | 2 | 2 | 2 |

Simulação Manual

Simulando código

Simule o algoritmo para M igual à 36 e N igual à 21.

Algoritmos

A parte difícil de garantir

- Partindo de objetos de entrada **válidos**, o algoritmo **termina** com a resultado **correto**.
- Partindo de objetos de entrada **inválidos**, o algoritmo pode produzir lixo e não **terminar**.

Exemplo

O algoritmo para o problema do MDC sempre **termina**

Termina?

Algoritmo de Euclides

- ① Se M é igual N , então MDC , Z , é M (ou N); pare!
- ② Caso contrário:
 - ① se M é maior que N , então substitua M por $(M-N)$ e repita a partir do passo 2
 - ② se N é maior que M , então substitua N por $(N-M)$ e repita a partir do passo 2

Exemplo:

Quando M e N são positivos, temos que:

- ① a cada execução M ou N **sempre** diminuem.
- ② M e N sempre são positivos, pois
 - ▶ $M - N$ é maior que 1, se M é maior que N ;
 - ▶ $N - M$ é maior que 1, se N é maior que M ;
- ③ não podemos passar de M igual à N igual à 1.

Logo, o algoritmo **termina!!**

Contra-exemplo

O que acontece quando os valores dos objetos não são válidos (quando M e/ou N não são positivos)?

Considere como contra-exemplo para simulação o caso em que M é igual à 5 e N é igual à -1 e mostre o que acontece.

A parte difícil de garantir - Outro exemplo

- Imagine que você criou um algoritmo que recebe um número inteiro N e procura os N primeiros números perfeitos.
- Atualmente, conhecemos apenas 48 números perfeitos, logo, não sabemos se o algoritmo termina quando $N = 49$, por exemplo.

Definição: Número perfeito

Um número inteiro é dito perfeito se a soma de todos os seus divisores positivos próprios (excluindo ele mesmo) é igual ao próprio número. Por exemplo, 6 é um número perfeito, pois é igual à $1 + 2 + 3$.

Nas próximas aulas...

- Veremos como codificar este algoritmos em C...

FIM

FIM!!!