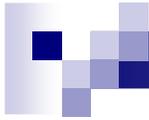


Introdução à Algoritmos

Aula 11

- 
- Um programa de computador é um produto resultante da atividade intelectual. Essa atividade depende de um treinamento **prévio** em abstração e modelagem de problemas, bem como o uso da lógica na verificação das soluções.
 - PROGRAMA == SOFTWARE
 - **Software** é representado por **instruções** e **dados** que alguém definiu e que ao serem executados por alguma **maquina** cumprem algum **objetivo**

- 
- **Ação: *Acontecimento*** que a partir de um ***estado inicial***, após um período de ***tempo finito*** produz um ***estado final previsível*** e bem ***definido***
 - Ex: Três pessoas escolhem um valor numérico L e escrevem os termos de Fibonacci inferiores a L
 - 1. L = 50 1 1 2 3 5 8 13 21 34
 - 2. L = 13 1 1 2 3 5 8
 - 3. L = 1
 - Apenas uma parte do que foi feito é considerado de ação.
 - A seleção de L não é uma ação
 - Escrever os termos da seqüência de Fibonacci inferiores a L é uma ação



- Note que as três ações são diferentes, com três pessoas diferentes com resultados diferentes e valores iniciais diferentes, mas apesar disso pode-se reconhecer um padrão de comportamento, subordinação a uma mesma norma de execução. É como se as três ações tivessem sido executadas em obediência ao comando.
- “Escreva os termos de Fibonacci inferiores a L”
- Essa norma de execução, comando ou padrão de comportamento pode-se definir como **ALGORITMO**



■ **ALGORITMO:** Descrição de um conjunto de comandos que obedecidos resultam numa sucessão de ações.

■ Se destina a resolver um problema, fixa um padrão de comportamento a ser seguido, uma norma de execução a ser trilhada para se atingir como resultado final, a solução de um problema.

■ **Exercícios:**

□ Um algoritmo não pode conter um comando como “Escreva todos os termos da seqüência de Fibonacci”. Por que?

□ Escreva um algoritmo que partindo de diferentes estados iniciais produzem os valores

■ 2 4 6 8 10 12 14

■ 1 3 5 7 9 11 13

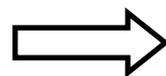


Exemplos de Algoritmos na Vida Cotidiana

- Instruções para se utilizar um aparelho eletrodoméstico
- Uma receita de cozinha
- Uma guia de preenchimento da declaração de imposto de renda
- A regra para a determinação do máximo ou mínimo de funções usando derivadas sucessivas
- A maneira em que as contas de água e luz são calculadas mensalmente

- Um **ALGORITMO** é considerado **COMPLETO** se os seus comandos forem do entendimento do destinatário.
- Num algoritmo um comando que não for entendido deverá ser desdobrado em novos comandos, que constituirão um **REFINAMENTO** do comando inicial.
- Se um algoritmo é formado não apenas por um comando, mas por vários, isso significa que não consideram apenas o estado **inicial** e **final** mas também os **estados intermediários** que delimitam as ações decorrentes de cada comando.
- Exemplo:

1. Escreva os termos de Fibonacci menores do que L



1. Receba o valor de L
2. Processe os dois primeiros termos
3. Processe os termos restantes



■ A tarefa de especificar os algoritmos para representar um programa consiste em detalhar os dados que serão processados pelo programa e as instruções (comandos) que vão a operar sobre esses dados

■ **Formalização de um algoritmo**

- Regras para escrever um algoritmo corretamente (tipos de comandos que podem ser utilizados) expressões e operações dentro desses comandos
- Existem três tipos de estruturas que podem ser usadas para escrever qualquer programa (Estruturas de Controle)
 - Seqüenciais
 - De decisão
 - De repetição

- Exemplo (Estrutura Seqüencial) Após refinamento 1
- Os comandos são executados na mesma ordem em que aparecem

1. Receba o valor de L

2. Processe os dos primeiros termos



Comandos vagos (Não se sabe como pode ser feito), passo 2 e 3

3. Processe os termos restantes

Refinamento do passo 2

2.1 Atribuía o valor de 1 ao primeiro termo

2.2 Se ele for menor que L entao escreva-o Fim-se

2.3 Atribuía o valor 1 ao segundo termo

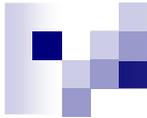
2.4 Se ele for menor que L entao escreva-o Fim-se

- Neste refinamento temos ESTRUTURAS CONDICIONAIS



Estruturas condicionais

- Num algoritmo a execução de estruturas condicionais provocarão a execução ou não de uma ação.
- Estrutura
 - Se condição então comandos Fim-se



Exemplo (Estrutura Condicional) Após refinamento 2

Os comandos são executados se se cumprir uma condição

1. **Receba o valor de L**
2. **Atribua o valor de 1 ao primeiro termo**
3. **Se ele for menor que L então escreva-o Fim-se**
4. **Atribua o valor 1 ao segundo termo**
5. **Se ele for menor que L então escreva-o Fim-se**
6. **Processe os termos restantes** ← Ainda não se sabe como vai ser feito

Refinamento do passo 6

6.1 Repita

6.2 **Calcule novo termo somando os dois anteriores**

6.3 **Se novo termo for maior que L então interrompa Fim-se**

6.4 **escreva o termo**

6.5 Fim-repita

■ Neste refinamento temos ESTRUTURAS DE REPETIÇÃO



Estruturas Repetição

- Nas estruturas de repetição os comandos (ou outras estruturas de controle abrangidas) serão executadas repetidamente até que se verifique a **condição** (Em nosso caso o novo termo seja maior que L)
- A forma de descrição deste algoritmo é mais sintética, elegante e consegue ser executada por qualquer pessoa que tente usar esse algoritmo
- Estrutura
 - repita verificação de condição e comandos Fim-repita



Algoritmo Final

1. Receba o valor de L
2. Atribua o valor de 1 ao primeiro termo
3. Se ele for menor que L entao escreva-o Fim-se
3. Atribua o valor 1 ao segundo termo
5. Se ele for menor que L entao escreva-o Fim-se
6. Repita
7. Calcule novo termo somando os dois anteriores
8. Se novo termo for maior que L entao interrompa Fim-se
9. escreva o termo
10. Fim-repita



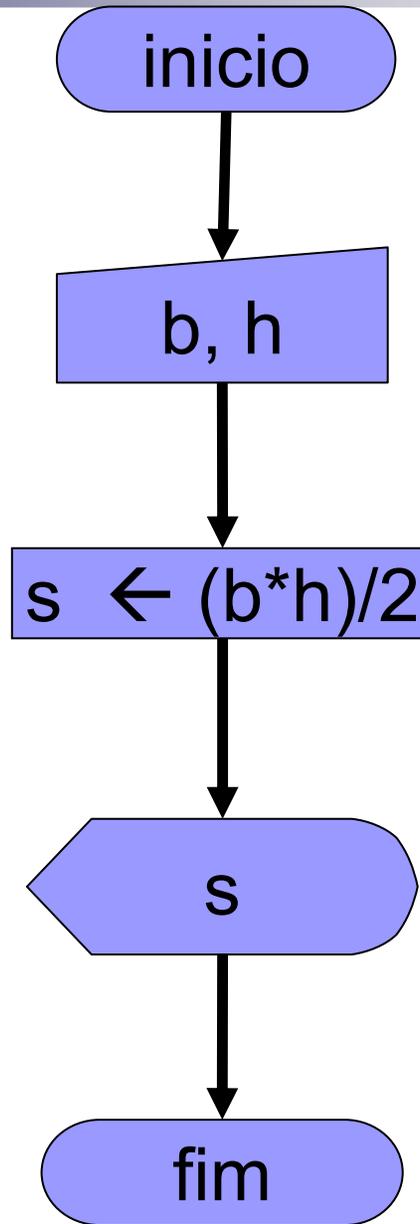
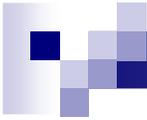
Exercício

- Deseja-se especificar um algoritmo para calcular e exibir na tela a área de um triângulo de base b e altura h , esses valores são fornecidos pelo usuário

Início

1. Pedir para o usuário digitar os valores de b e de h .
2. Calcular a área s usando a fórmula $s = (bxh)/2$
3. Exibir o valor de s na tela

Fim





Algoritmo em Portugol

Início

1. Leia(b, h)

2. $s \leftarrow (b * h) / 2$

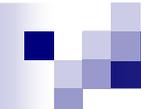
3. Exiba(s)

Fim



Modelagem do problema

- A modelagem é sempre desprezada é a responsável pela facilidade ou dificuldade da solução do problema. O uso da linguagem matemática pode ser fundamental
- Compraram-se 30 canetas iguais que foram pagas com uma nota de 100 reais obtendo-se 67 reais de troco. Quanto custou cada caneta
- Raciocínio matemático
 - quantogastei = $30x$
 - quantogastei + troco = 100
 - $30x + 67 = 100$
 - $30x = 33$
 - $x = 33/30$
 - $x = 1.1$



Algoritmo para solucionar o problema

Inicio

1. Pegar os valores 30, 100 e 67
2. Subtrair 67 de 100 e dividir o resultado por 30
3. Mostre o resultado final

Fim

Algoritmo geral

Inicio

1. Ler os valores de Q, N e T
2. $r \leftarrow (N-T)/Q$
3. Mostrar r

Fim



- **ALgoritmo Correto**

Inicio

Ler os valores de Q, N e T

Se $T > N$ e $Q > 0$ e $N \geq 0$ e $T > 0$ Entao

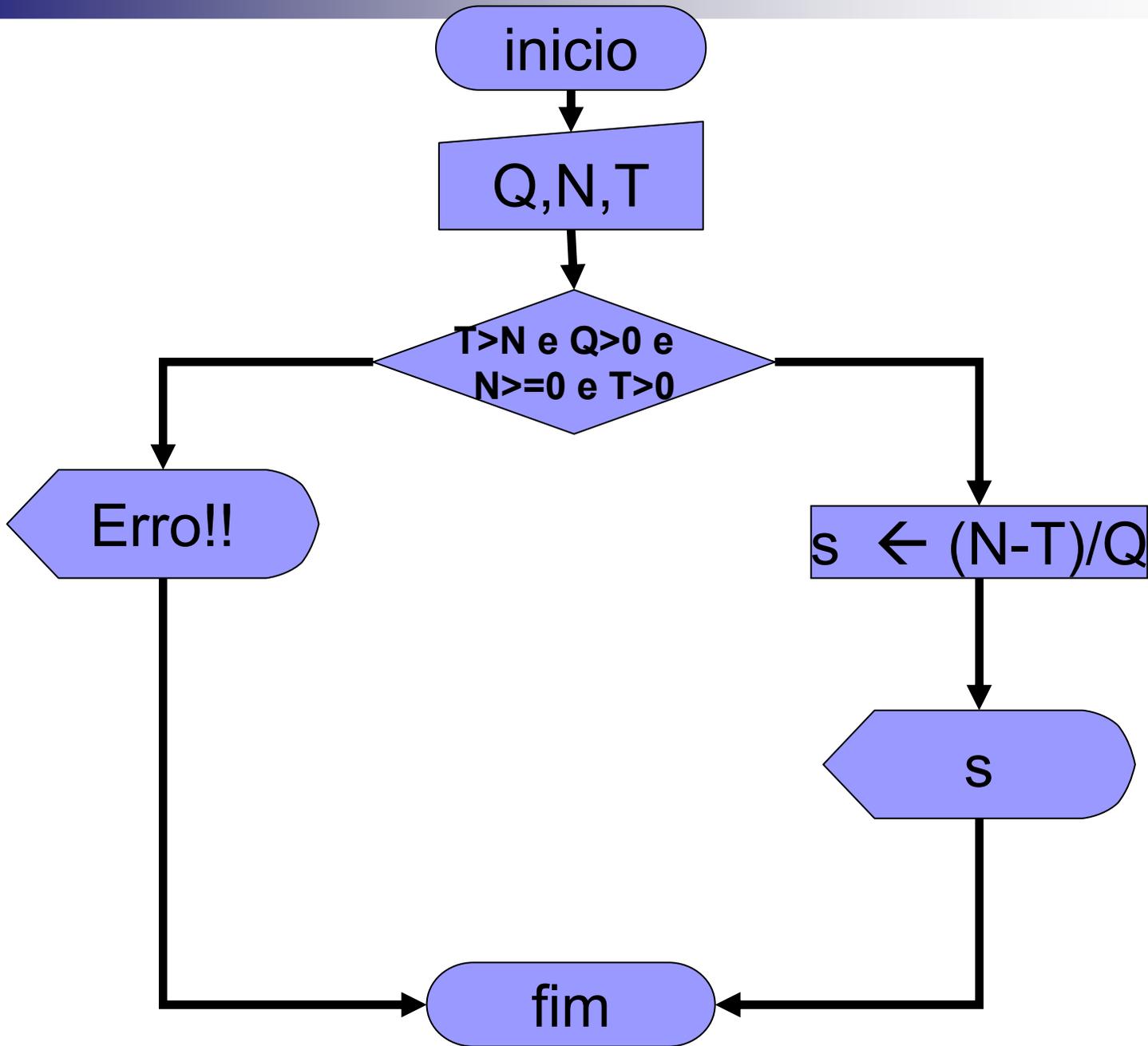
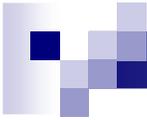
$r \leftarrow (N - T) / Q$

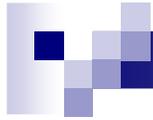
mostrar r

Senão Exibir mensagem “Erro nos valores”

Fim

- **Em portugol???**





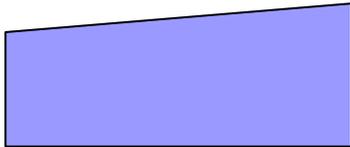
Terminador: Significa saída para ou entrada do ambiente externo



Processo: Algum processo, uso de uma função grupo de operações entre as variáveis.



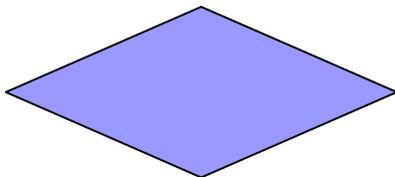
Linha básica: Direção de fluxo e dados, pode ser usada sólidas ou abertas, com ou sem setas



Entrada manual: Representa os dados externos de qualquer tipo de mídia que sejam fornecidos manualmente durante o processamento por teclado, online, caneta óptica, leitor de código de barras



Exibição: Dados mostrados em qualquer tipo de mídia resultantes do processamento



Decisão: Representa uma decisão ou desvio



- Exercício Represente o algoritmo de fibonacci com FLUXOGRAMAS

- Exercício: Que faz o seguinte algoritmo

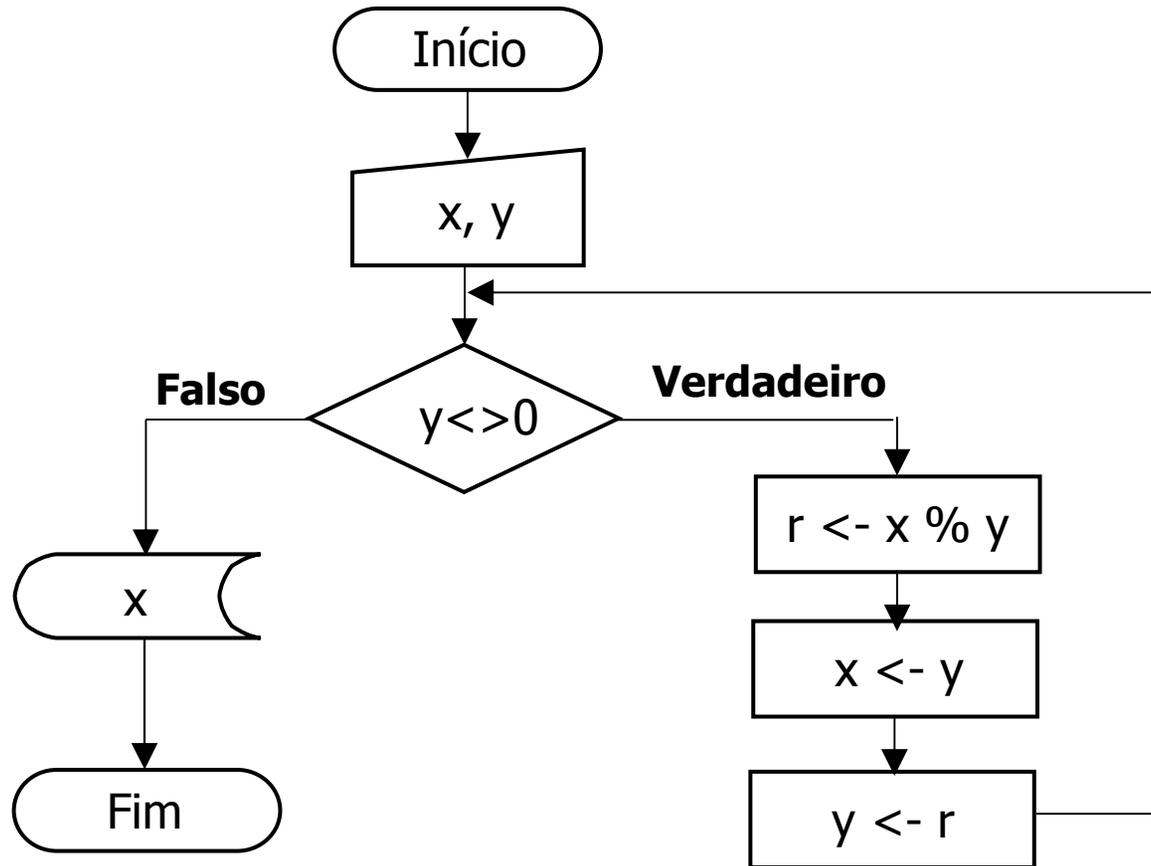
Inicio

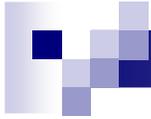
1. **Leia** (x, y)
2. **Enquanto** ($y \neq 0$) **Faça**
3. $r \leftarrow x \% y$
4. $x \leftarrow y$
5. $y \leftarrow r$
6. **Fim Enquanto**
7. **Exiba** (x)

Fim

- Considere x e y valores inteiros. Elabore o fluxograma do algoritmo anterior.

• Exercício... Fluxograma





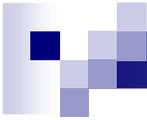
- **Exercício:** Crie um algoritmo que calcule quantas notas de 50, 10, 5 e 1 são necessárias para pagar uma conta cujo valor é fornecido. Considere valores inteiros. Utilize as técnicas de representação de algoritmos estudadas (fluxograma e portugol).



DICAS

1. Ao se deparar com um problema novo, tente entendê-lo:

- O que se deve descobrir ou calcular? (Objetivo)
- Quais são os dados disponíveis? São suficientes?
- Quais as condições necessárias e suficientes para resolver o problema?
- Se possível, modele o problema de forma matemática.



DICAS

2. Crie um plano com a solução:

- Consulte sua memória e verifique se você já resolveu algum problema similar (analogia, generalização, especialização)
- Verifique se é necessário introduzir algum elemento novo no problema, como um problema auxiliar.
- Se o problema for muito complicado, tente quebrá-lo em partes menores e solucionar essas partes.



DICAS

3. Formalize a solução:

- Crie um algoritmo informal com os passos que resolvam o problema.
- Verifique se cada passo do algoritmo esta correto.
- Escreva um algoritmo formalizado (fluxograma ou portugol)



DICAS

4. Exame dos resultados:

- Teste o algoritmo com diversos dados e verifique os resultados (teste de mesa)
- Se o algoritmo não gerou resultado algum. Volte e tente encontrar o erro.
- Se o algoritmo gerou resultados, estes estão corretos?
- Se não estão corretos, alguma condição, operação ou a ordem, estão incorretas. Volte e tente encontre o erro.



DICAS

5.Otimização da solução:

- É possível melhorar o algoritmo?
- É possível reduzir o número de passos ou dados?
- É possível conseguir uma solução ótima?