



• ALGORITMOS & LÓGICA DE PROGRAMAÇÃO

Cláudio C. Rodrigues

1



Algoritmos & Lógica

- O conceito de Algoritmos
- O delineamento de Algoritmos
 - A teoria da Resolução de Problemas
- A representação de Algoritmos
 - Narrativa
 - Fluxograma
 - Pseudocódigo
 - Primitivas
 - Linguagem de Programação
- Lógica

2



O Conceito de Algoritmo

- Um algoritmo é um conjunto ordenado de passos executáveis não ambíguos, definindo um processo que tem um término.
- Conceitos inter-relacionados:
- **Algoritmo:**
 - abstrato e distinto de sua representação
- **Programa:**
 - É uma das possíveis representações do algoritmo
- **Processo:**
 - A atividade de executar um algoritmo.

3



Um Algoritmo

- É uma sequência de passos (ações) que visam atingir um objetivo bem definido.
- Segundo *Dijkstra*, um algoritmo corresponde a uma descrição de um padrão de comportamento, expresso em termos de um *repertório bem definido e finito de ações “primitivas”*, das quais damos por certo que elas podem ser executadas.
 - Executando a operação **$a + b$** percebemos um padrão de comportamento, mesmo que a operação seja realizada para valores diferentes de **a** e **b** .
- Programar é basicamente construir algoritmos



Estruturas de dados

- Estruturas de dados e algoritmos estão intimamente ligados:
 - não se pode estudar estruturas de dados sem considerar os algoritmos associados a elas,
 - assim como a escolha dos algoritmos em geral depende da representação e da estrutura dos dados.
- Para resolver um problema é necessário escolher uma abstração da realidade, em geral mediante a definição de um conjunto de dados que representa a situação real.
- A seguir, deve ser escolhida a forma de representar esses dados.

5



Programas

- Programar é basicamente estruturar dados e construir algoritmos.
- Programas são formulações concretas de algoritmos abstratos, baseados em representações e estruturas específicas de dados.
- Programas representam uma classe especial de algoritmos capazes de serem seguidos por computadores.
- Um computador só é capaz de seguir programas em linguagem de máquina (sequência de instruções obscuras e desconfortáveis).
- É necessário construir linguagens mais adequadas, que facilitem a tarefa de programar um computador.
- Uma linguagem de programação é uma técnica de notação para programar, com a intenção de servir de veículo tanto para a expressão do raciocínio algorítmico quanto para a execução automática de um algoritmo por um computador.

6



Características do Algoritmo

- **Finitude:** um algoritmo tem de terminar ao fim de um número finito de passos.
- **Definitude:** cada passo do algoritmo tem de ser definido com precisão.
- **Entradas:** um algoritmo pode ter zero ou mais entradas.
- **Saídas:** um algoritmo tem uma ou mais saídas.
- **Eficácia:** todas as operações feitas por um algoritmo têm de ser básicas.



Um algoritmo:

1. Possui um ponto de entrada (passo inicial) e um ponto de saída (passo final) bem definidos.
2. É composto de passos individuais.
3. Cada passo está bem definido, pode ser executado, e o seu resultado é previsível. Isto é, repetidas execuções de um determinado passo produzirão sempre o mesmo resultado.
4. Existe uma ordem (sequência) adequada para a execução dos passos. Essa sequência é fundamental para a obtenção de um resultado final correto. Em geral, depois de executado um determinado passo, a execução prossegue no passo seguinte.
5. O número de passos a executar é finito.
6. Quando executado com um conjunto de dados válido um algoritmo termina garantidamente, produzindo o resultado esperado.



A representação de Algoritmos

- **Primitivas:**
 - Conjunto bem definido de **elementos funcionais básicos** com os quais podem ser construídas representações de algoritmos.
 - Um conjunto de **primitivas**, juntamente com um de **regras**, estabelecendo o modo que as primitivas podem ser combinadas, constitui uma **linguagem de programação**.
- Cada **primitiva** consiste de duas partes:
 - **Sintática:**
 - representação simbólica da primitiva
 - **Semântica:**
 - significado da primitiva

9



Böhm e Jacopini (1966)

- as estruturas têm um papel fundamental quando desejamos descrever um algoritmo que seja computável, isto é, implementável em um computador;
- Böhm e Jacopini demonstraram em 1966 que qualquer programa de computador pode ser decomposto através de três tipos de estruturas de programação:
 1. **estrutura sequencial;**
 2. **estrutura condicional;**
 3. **estrutura repetição;**
- estas estruturas também podem estar embutidas umas nas outras, por exemplo, uma estrutura repetitiva pode repetir uma sequência onde um dos comandos desta sequência seja uma estrutura condicional que por sua vez condiciona uma outra sequência, etc.



Entidades Básicas

- Do ponto de vista da forma como decorre o fluxo de execução num algoritmo (qual a sequência de execução dos passos), pode ser demonstrado que qualquer algoritmo de computador pode ser construído utilizando apenas três tipos de entidade de construção:
 1. **Sequência** - Salvo outra indicação, os passos são executados em sequência, de cima para baixo.
 2. **Decisão Condicional** - Uma forma de decidir entre a execução de duas instruções ou dois conjuntos de instruções.
 3. **Repetição** - Uma forma de repetir, várias vezes, a execução de uma dada instrução ou conjunto de instruções.
- Isto é, qualquer método de representação que permita representar as três entidades acima descritas é suficiente para representar qualquer tipo de algoritmo.

11



O delineamento de Algoritmos

- A teoria da Resolução dos problemas
 - Resolver problemas é uma arte
 - (daí a necessidade de desenvolvermos esta habilidade)
- Modelo de G. Polya (matemático)
 - Fase 1 – Entender o problema
 - Fase 2 – Construir um plano para solucionar o problema
 - Fase 3 – Colocar o plano em funcionamento.
 - Fase 4 – Avaliar a solução quanto à precisão e quanto ao seu potencial como ferramenta para solucionar outros problemas.

12



O delineamento de Algoritmos

- Traduzindo o modelo de *Polya* para o contexto do desenvolvimento de programas, estas fases se tornam:

Fase 1 – Compreender o problema

Fase 2 – Adquirir uma idéia da forma como um procedimento algorítmico poderia resolver o problema

Fase 3 – Formular o algoritmo e representá-lo na forma de um programa.

Fase 4 – Avaliar o programa quanto à precisão e quanto ao seu potencial como ferramenta para solucionar outros problemas.

13



O delineamento de Algoritmos

- O desenvolvimento de um programa consiste de duas atividades – **delinear** o algoritmo subjacente, e **representar** o algoritmo na forma de um programa.
- Delinear um algoritmo é descobrir um método de solucionar um problema. Assim, entender como os algoritmos são delineados é entender o próprio processo de resolução de problemas.

14



Passos para Elaborar Algoritmos

- Compreender o problema
- Identificar os dados de entrada
- Identificar os dados de saída
- Determinar o que é preciso para transformar dados de entrada em dados de saída:
 - usar a estratégia do dividir-para-conquistar
 - observar regras e limitações
 - identificar todas as ações a realizar
 - eliminar ambiguidades
- Construir o algoritmo
- Testar o algoritmo
- Executar o algoritmo

15



Descrição de Algoritmos

- **Narrativa:** é o algoritmo descrito com o uso da linguagem natural. A linguagem natural é abstrata, imprecisa e frequentemente pouco confiável. Isto pode ser observado por pessoas que tentam redigir um contrato ou um documento legal;
- **Fluxograma:** é o algoritmo descrito com o uso de símbolos geométricos (retângulos, losangos, etc.) enfatizando seus passos individuais e suas interconexões;
- **Pseudocódigo:** é o algoritmo descrito com o uso da combinação entre a Narrativa e o Fluxograma, extraindo suas melhores características. É similar a várias linguagens de programação como, por exemplo, Pascal, C/C++, Fortran, Java, etc.

Algoritmo Textual Informal

- Modo de preparo:

- Bata a margarina, as gemas e o açúcar até ficar cremoso Quão cremoso?!?
- Junte o leite, o coco e a farinha e continue batendo Quanto tempo?!?
- Acrescente o fermento e, por último, as claras em neve De uma vez só?!?
- Unte uma forma com manteiga e leve ao forno para assar Quanto tempo?!?

Algoritmo Textual Informal Refinado

- Modo de preparo:

- Bata a margarina, as gemas e o açúcar por 15 minutos
- Junte o leite, o coco e a farinha e continue batendo por mais 15 minutos
- Acrescente 20 g de fermento e, por último, as claras em neve
- Unte uma forma com manteiga e leve ao forno para assar por 30 minutos



Representação de Algoritmos

- Descrição Narrativa - exemplo:
 - Trocar o pneu de um carro
 1. Afrouxar ligeiramente as porcas
 2. Suspende o carro
 3. Retirar as porcas e o pneu
 4. Colocar o Pneu reserva e as porcas
 5. Abaixar o carro
 6. Dar o aperto final nas porcas

19



Representação de Algoritmos

- Descrição Narrativa - exemplo:
 - Tomar banho
 - Entrar no banheiro
 - Tirar a roupa
 - Abrir a torneira
 - Entrar na água
 - Ensaboar-se
 - Sair da água
 - Fechar a torneira
 - Enxugar-se
 - Vestir-se

20



Narrativa

- Sequência para fritar um ovo:
 1. Retirar o 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 do ovo na frigideira
 7. Esperar um minuto
 8. Retirar o ovo da frigideira
 9. Apagar o fogo



Narrativa

- **Sequencia para tomar um ônibus.**

ir até a parada
enquanto ônibus não chega faça
esperar ônibus
fim-enquanto
subir no ônibus
pegar passagem
se não há passagem então
pegar dinheiro
fim-se
pagar o cobrador
troco = dinheiro – passagem
enquanto banco não está vazio faça
ir para o próximo
fim-enquanto
Sentar
...

Exercício



- Ordene e estruture uma sequência para a realização de uma ligação telefônica em um orelhão.

Exercício

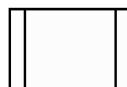
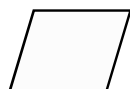
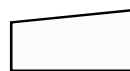
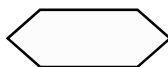
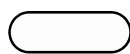
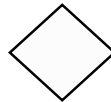
- Resposta I:
 - Tirar o fone do gancho
 - Ouvir sinal de linha
 - Introduzir cartão
 - Teclar o número desejado
 - Conversar
 - Desligar
 - Tirar o cartão

Exercício

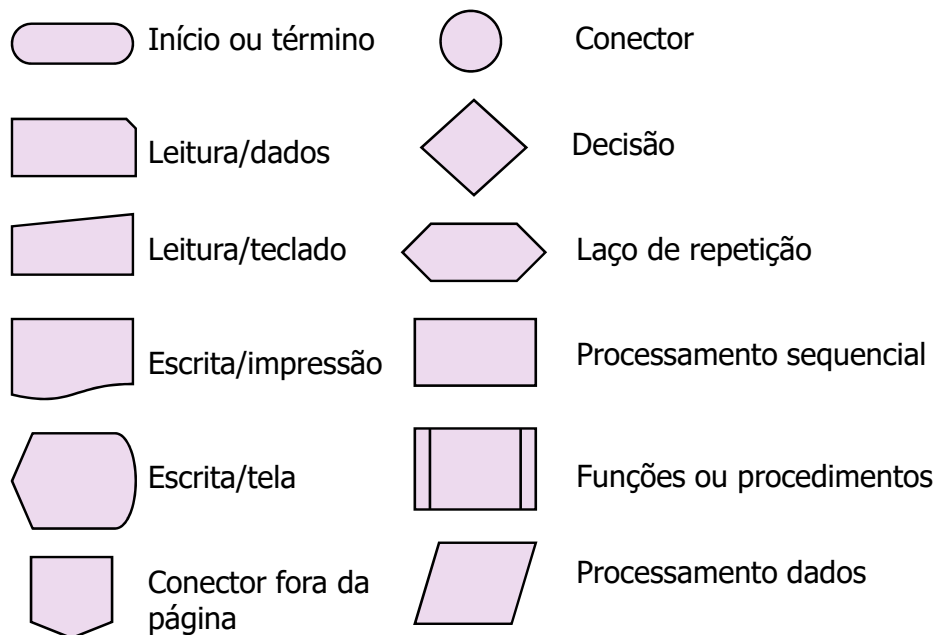
- Resposta 2 (Mais robusta)
 - Tirar o fone do gancho
 - Ouvir sinal de linha
 - Introduzir cartão
 - Teclar o número desejado
 - Se chamar e alguém atender
 - Conversar
 - Desligar
 - Retirar o cartão
 - Senão
 - Desligar
 - Voltar para a instrução I

Representação de Algoritmos

- **Fluxograma**
 - Representação gráfica pelo uso de formas geométricas associadas a ações.

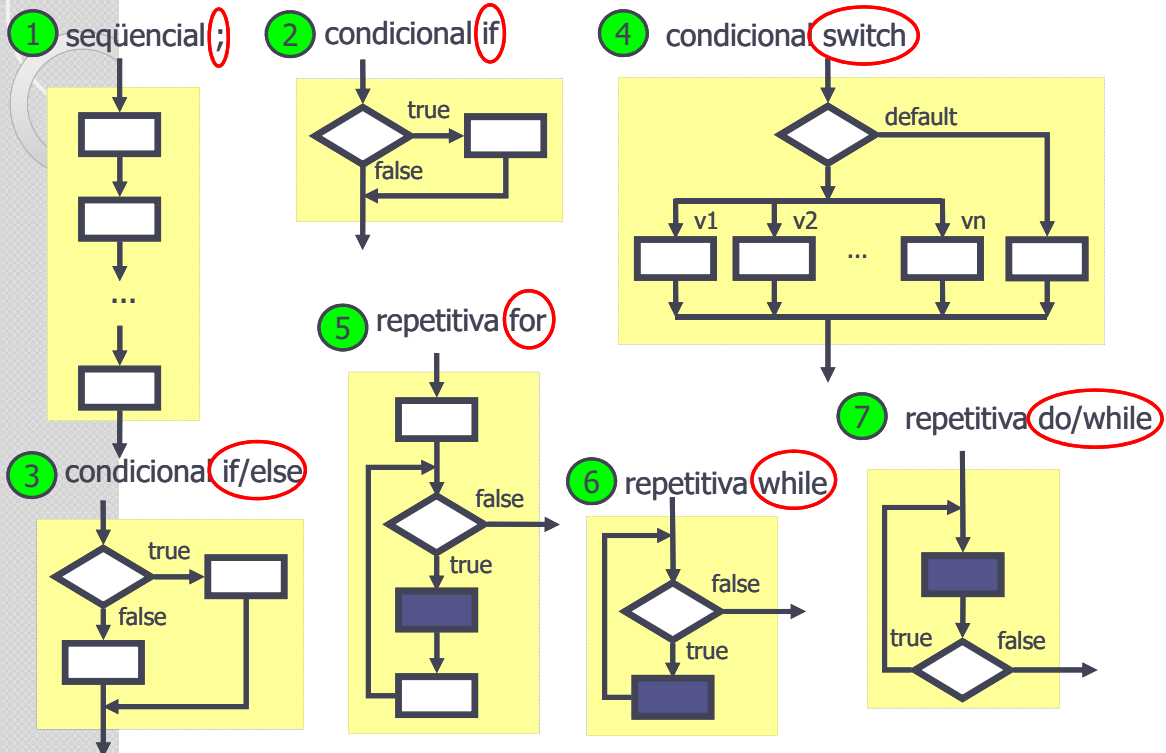


Identificação dos Símbolos



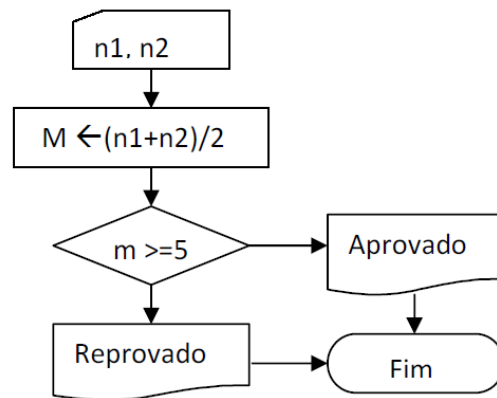
27

Blocos Básicos em C/C++:



Representação de Algoritmos

- Cálculo da Média Final

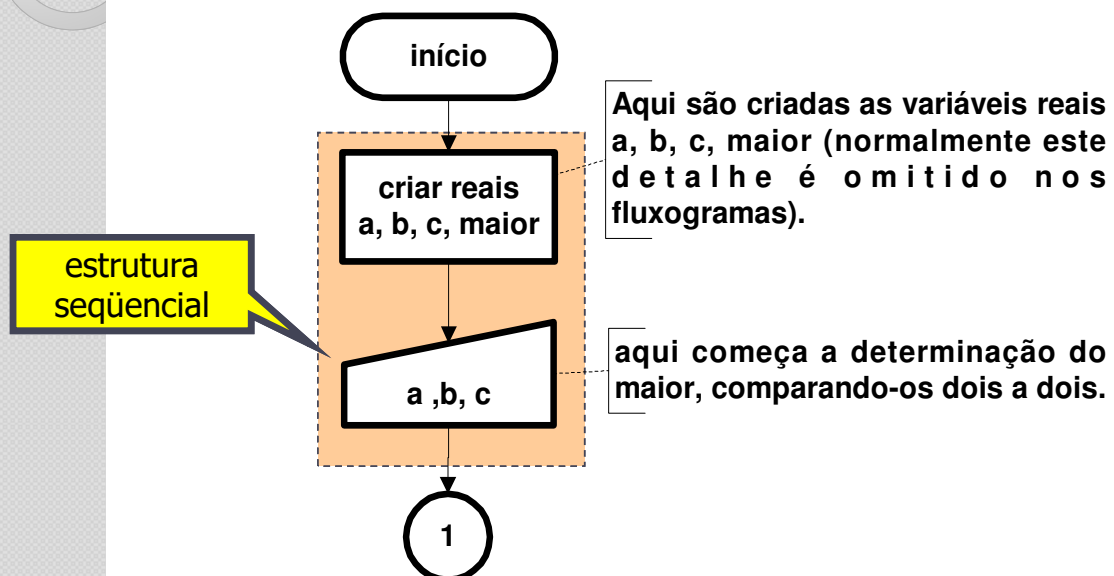


- **Fluxograma Convencional**

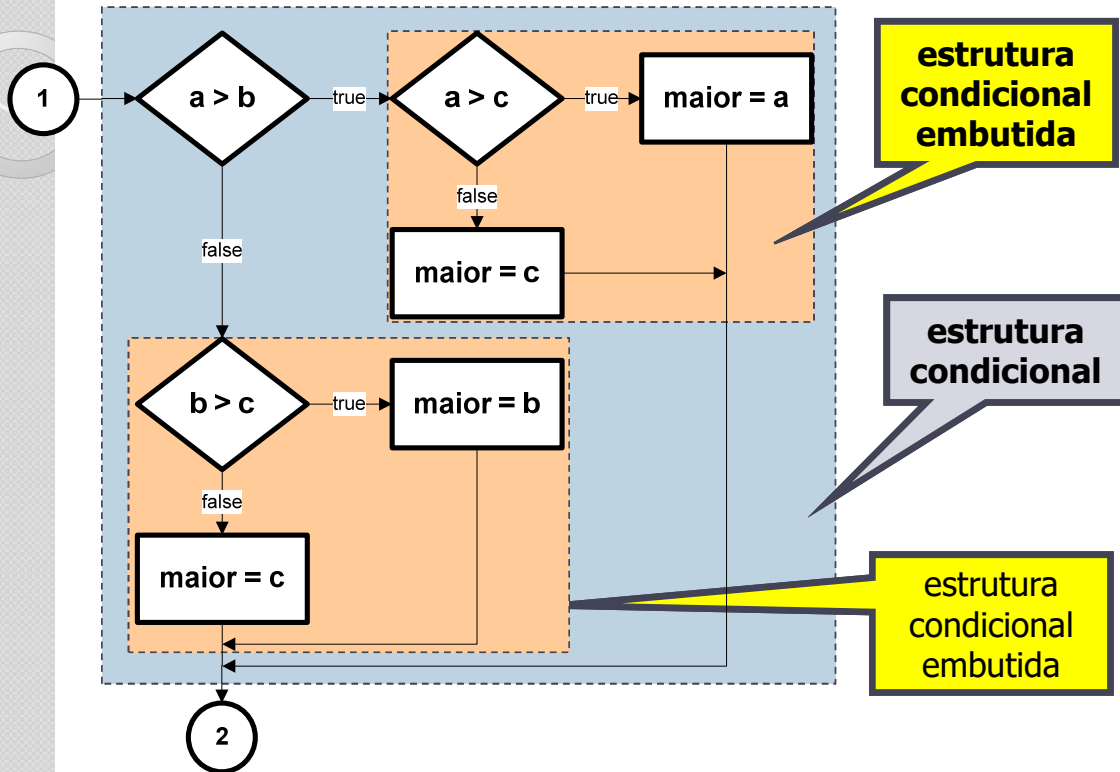
- Tal propriedade facilita o entendimento das idéias contidas nos algoritmos e justifica sua popularidade.
- Esta forma é aproximadamente intermediária à descrição narrativa e ao pseudocódigo (subitem seguinte).

29

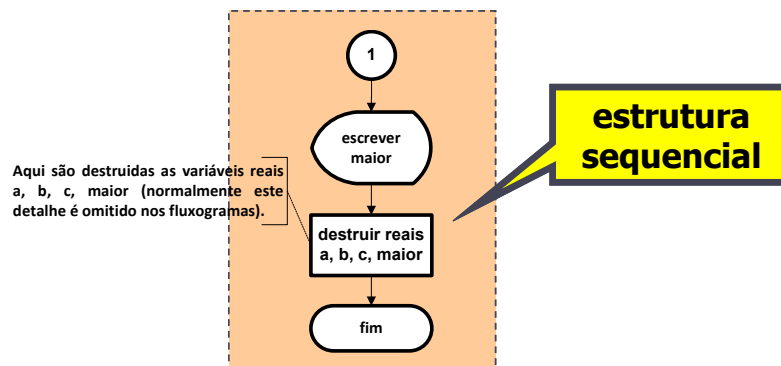
Exercício 01: elabore um fluxograma para determinar o maior valor entre três valores reais.



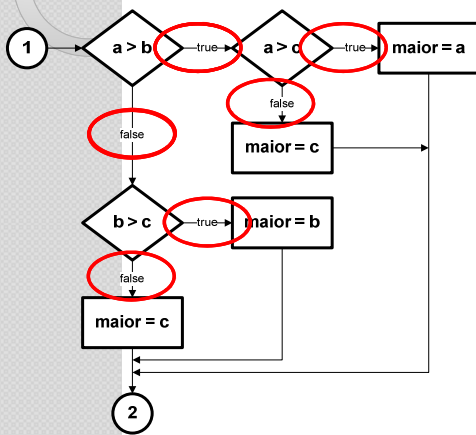
Exercício 01: continuação.



Exercício 01: continuação.

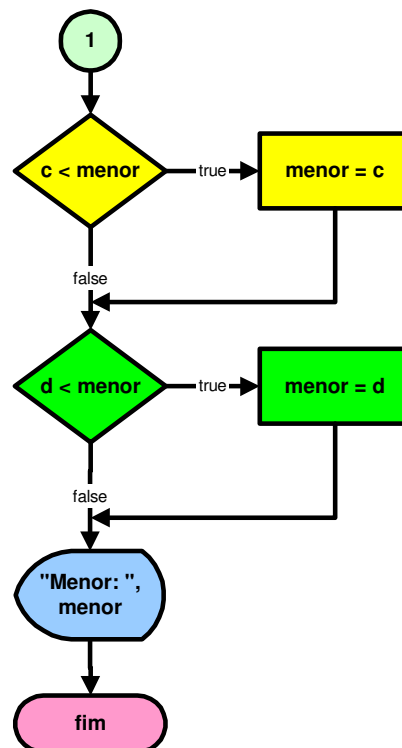
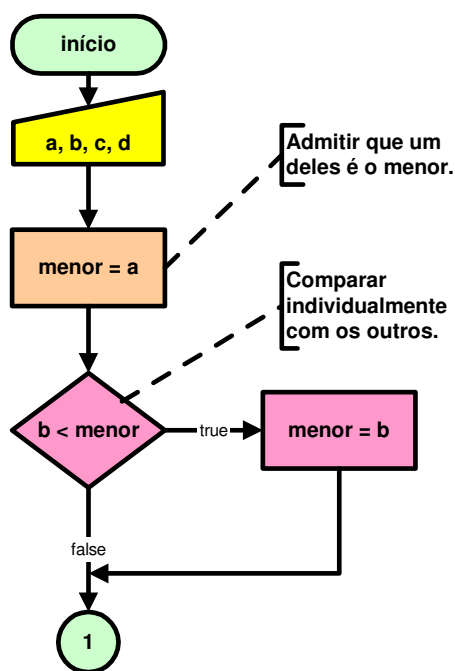


Exercício 02: teste o exercício 01 para os valores abaixo. Determine o maior e também o caminho seguido no fluxo. Veja como exemplo o teste 1.



Teste	a	b	c	maior	caminho
1	69	21	7	69	true, true
2	21	69	7	69	false, true
3	21	7	69	69	true, false
4	7	21	69	69	false, false
5	7	7	7	7	false, false

Exercício 03: elabore um fluxograma para determinar o menor valor entre quatro valores reais.



A representação de Algoritmos

- **Pseudocódigo**

- É um sistema notacional no qual as idéias podem ser informalmente expressas durante o processo de desenvolvimento do algoritmo.
- A tática será a de desenvolver uma notação consistente e concisa para representar as estruturas semânticas mais freqüentes, as quais, por sua vez, se tornarão as primitivas de nossa representação.

35

Pseudocódigo

- A forma geral da representação de um algoritmo na forma de pseudocódigo.
- **Algoritmo** é uma palavra que indica o início da definição de um algoritmo em forma de pseudocódigo.
- **<nome_do_algoritmo>** é um nome simbólico dado ao algoritmo com a finalidade de distingui-los dos demais.
- **<declaração_de_variáveis>** consiste em uma porção opcional onde são declaradas as variáveis globais usadas no algoritmo principal e, eventualmente, nos subalgoritmos.
- **<subalgoritmos>** consiste de uma porção opcional do pseudocódigo onde são definidos os subalgoritmos.
- **Início** e **Fim** são respectivamente as palavras que delimitam o início e o término do conjunto de instruções do corpo do algoritmo.

Algoritmo **<nome_do_algoritmo>**

<declaração_de_variáveis>

<subalgoritmos>

Início

<corpo do algoritmo>

Fim

36

Representação de Algoritmos

- Exemplo: Cálculo da média do exemplo anterior.

Algoritmo Media_Notas

Real: n1, n2, media

Início

Escreva("Digite duas notas:")

Leia(n1,n2)

media $\leftarrow (n1+n2)/2$

Se (média ≥ 5) então

Escreva ("APROVADO")

Senão

Escreva ("REPROVADO")

Fimse

FimAlgoritmo

37

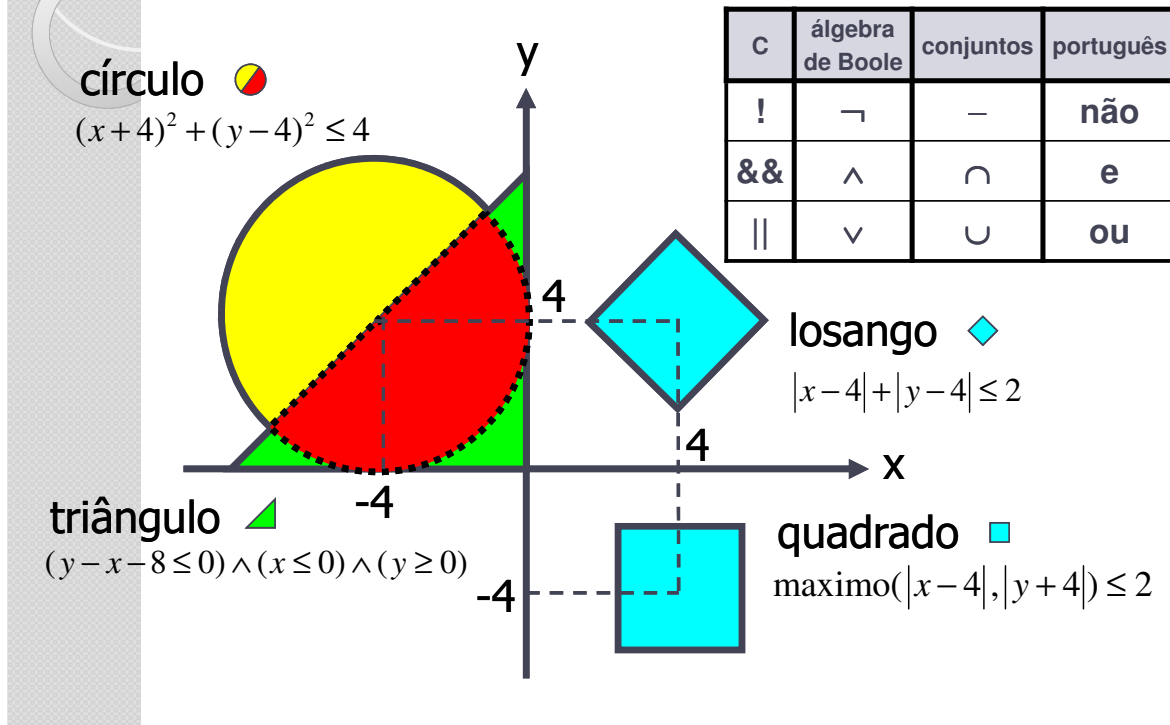
Resolver Equação do 2º Grau

♦ Algoritmo Resolver_equação

- Conhecer os coeficientes da equação de 2º grau:
no formato $Ax^2 + Bx + C = 0$
- Se o coeficiente de x^2 for igual a zero:
não é equação de 2º grau
- Calcular o valor de Delta = $B^2 - 4AC$
- Se o valor de Delta for negativo não,
existem raízes reais. Sair do programa.
- Calcular a primeira raiz.
Calcular a segunda raiz.

♦ Fim_do_algoritmo.

Exercício 04: elabore em pseudocódigo para descobrir qual é a cor do ponto de coordenadas cartesianas x e y dadas.



Exercício 04: solução.

```
int main(int argc, char *argv[])
{
    cout << "Coordenadas x e y?...";
    double x, y;
    cin >> x >> y;
    double a = abs(x - 4.0);
    double b = abs(y + 4.0);
    double maximo;
    if(a > b) maximo = a;
    else maximo = b;

    if(maximo <= 2.0 || a + abs(y - 4.0) <= 2.0)
        cout << "Azul" << endl;
    else{
        double a = x + 4.0;
        double b = y - 4.0;
        double q = a * a + b * b;
        double r = y - x - 8.0;
        if(q <= 4.0 && r > 0.0) cout << "Amarelo" << endl;
        else if(q <= 4.0 && r <= 0.0) cout << "Vermelho" << endl;
        else if(r <= 0.0 && x <= 0.0 && y >= 0.0)
            cout << "Verde" << endl;
        else cout << "Branco" << endl;
    }
    return 0;
}
```



Lógica

- É a forma de ordenar o pensamento.
- É a arte de bem pensar.
- Podemos dizer que a lógica tem em vista a “ordem da razão”.
 - Correção/validação do pensamento.
 - Encadeamento/ordem de ideias.
- Existe lógica no dia-a-dia?
 - A gaveta está fechada. A caneta está dentro da gaveta. Precisamos primeiro abrir a gaveta para depois pegar a caneta.



Lógica: Problema Ilustrativo I

- O indivíduo **A** é encarregado de determinar a idade dos três filhos do indivíduo **B**. **B** conta a **A** que o produto das idades das crianças é **36**. Depois de levar em conta esta informação, **A** responde que precisa de mais informação, e então **B** conta a **A** a soma das idades das crianças. Novamente, **A** diz que necessita de mais informação, e assim **B** conta que a criança mais velha toca piano. Depois de ouvir isto, **A** responde para **B** a idade das três crianças.
- Qual é a idade de cada uma das três crianças?

Resolvendo as Identidades

- Triplas cujo Produto é 36

(1,1,36) (1,6,6)

(1,2,18) (2,2,9)

(1,3,12) (2,3,6)

(1,4,9) (3,3,4)

- Somas das Triplas

$$1+1+36=38$$

$$1+6+6=13$$

$$1+2+18=21$$

$$2+2+9=13$$

$$1+3+12=16$$

$$2+3+6=11$$

$$1+4+9=14$$

$$3+3+4=10$$

43

Lógica: Problema Ilustrativo 2

- Antes de A, B, C e D participarem de uma corrida, eles fizeram as seguintes previsões:
 - A previu que B ganharia
 - B previu que D seria o último
 - C previu que A seria o terceiro
 - D previu que a previsão de A estaria correta
- Apenas uma destas previsões deu certo, e esta foi feita pelo vencedor da corrida. Em que ordem A, B, C e D terminaram a corrida?

44



Lógica: Problema Ilustrativo 3

- Dispomos de **duas vasilhas** com capacidades de 9 e 4 litros respectivamente. As vasilhas não tem nenhum tipo de marcação, de modo que não é possível ter medidas como metade ou um terço.
- Mostre uma seqüência de passos, que usando as vasilhas de 9 e 4 litros encha uma **terceira vasilha** de medida desconhecida com **seis** litros de água.

45



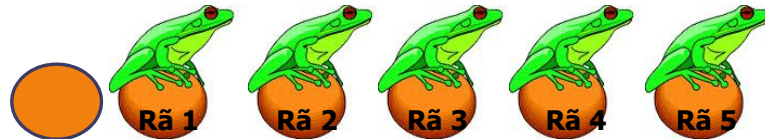
Lógica: Problema Ilustrativo 3

- Uma possível solução:
 - Encha a vasilha de 9 litros;
 - Usando a vasilha de 9 litros, encha a vasilha de 4 litros;
 - Despeje o que sobrou na vasilha de 9 litros (5 litros) na terceira vasilha. Observe que falta um litro para completar os seis litros;
 - Esvazie a vasilha de 4 litros;
 - Torne a encher a vasilha de 9 litros;
 - Usando a vasilha de 9 litros encha a vasilha de 4 litros;
 - Esvazie a de 4 litros;
 - Usando o que restou na vasilha de 9 litros (5 litros), encha novamente a vasilha de quatro litros;
 - Despeje o que sobrou na vasilha de 9 litros (1 litro) na terceira vasilha, que agora tem 6 litros.

46

Lógica: Problema Ilustrativo 4

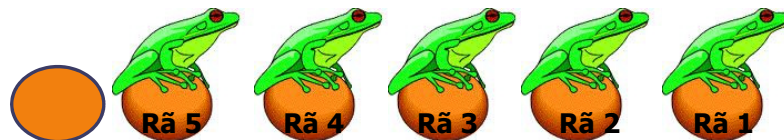
- Considere cinco rãs estão posicionadas em seis casas da seguinte maneira:



As rãs foram treinadas para trocar de casas, mas sempre obedecendo as seguintes regras:

- elas podem pular para a casa vizinha (frente ou trás), se ela estiver vazia;
- elas podem pular sobre a rã vizinha para uma casa livre (frente ou trás).

Mostre como as rãs podem chegar a seguinte posição final:



47

Problema Ilustrativo 4

	rã 1	rã 2	rã 3	rã 4	rã 5
rã 2	rã 1		rã 3	rã 4	rã 5
rã 2		rã 1	rã 3	rã 4	rã 5
rã 2	rã 3	rã 1		rã 4	rã 5
rã 2	rã 3		rã 1	rã 4	rã 5
rã 2	rã 3	rã 4	rã 1		rã 5
rã 2	rã 3	rã 4		rã 1	rã 5
rã 2	rã 3	rã 4	rã 5	rã 1	
rã 2	rã 3	rã 4	rã 5		rã 1
rã 2	rã 3	rã 4		rã 5	rã 1
rã 2	rã 3		rã 4	rã 5	rã 1
rã 2		rã 3	rã 4	rã 5	rã 1
	rã 2	rã 3	rã 4	rã 5	rã 1
rã 3	rã 2		rã 4	rã 5	rã 1
rã 3		rã 2	rã 4	rã 5	rã 1
rã 3	rã 4	rã 2		rã 5	rã 1
rã 3	rã 4		rã 2	rã 5	rã 1
rã 3	rã 4	rã 5	rã 2		rã 1
rã 3	rã 4			rã 2	rã 1
rã 3	rã 4		rã 5	rã 2	rã 1
rã 3		rã 4	rã 5	rã 2	rã 1
	rã 3	rã 4	rã 5	rã 2	rã 1
rã 4	rã 3		rã 5	rã 2	rã 1
rã 4	rã 5	rã 3		rã 2	rã 1
rã 4	rã 5		rã 3	rã 2	rã 1
rã 5	rã 4	rã 3	rã 2	rã 1	

- Este é um problema de colocar em ordem um conjunto de dados, no caso ordenação decrescente, tarefa muito comum em computação.
- Uma possível solução para este problema esta mostrada na Figura ao lado. Observe que em negrito estão as rãs que estão no momento sendo dirigidas para o seu destino.
- A aparente desordem esconde alguma ordem. Observe que no problema das rãs, elas seguiram um plano básico, que era fazer com que cada uma delas, uma por vez, achasse a sua posição final. A primeira a ir para sua posição final foi a rã 1, em seguida a rã 2 e assim por diante.

48

Critérios para Avaliação do Algoritmo

- Pode existir mais de uma solução para o mesmo problema
- Exemplo
 - Encontrar o máximo divisor comum de dois números “m” e “n” inteiros e positivos
 - Seja “r” o resto da divisão de “m” por “n” e $0 \leq r < n$

49

Soluções...

Solução 1

1. Seja “r” o resto da divisão de “m” por “n” e $0 \leq r < n$
2. Se $r = 0$ então escrever “n” como resposta e terminar.
3. $m = n$
4. $n = r$
5. Voltar ao passo 1.

Solução 2

1. Se “m” \geq “n” então fazer “m” igual ao resto de “m” dividido por “n” e repetir este mesmo passo
2. Trocar os valores de “m” e “n” entre si.
3. Se “n” = 0 então escrever “m” como resposta e terminar.
4. Voltar ao passo inicial.

50



Soluções...

Solução 3

1. Dividir “m” por “n” e fazer “r” ser o resto.
2. Se $r = 0$ então escrever “n” como resposta e terminar.
3. Dividir “n” por “r” e fazer “m” ser o resto.
4. Dividir “r” por “m” e fazer “n” ser o resto.
5. Se $m = 0$ escrever “m” como resposta e terminar.
6. Se $n = 0$ escrever “n” como resposta e terminar.
7. Voltar ao passo inicial.

51



Bibliografia

- Ciência da Computação: uma visão abrangente
– J. Glenn Brookshear – Bookman, 5a. edição

52