

Técnicas de solução de problemas

Objetivos:

- Entender como compor comandos básicos das linguagens de programação para resolver problemas
- Entender os princípios da lógica de programação

Programação estruturada

- Entrada (read, readln)
- Saída (write, writeln)
- Atribuição (:=)
- Comando de repetição (while ... do)
- Desvios condicionais (if ... then ... else)

- Expressões aritméticas
- Expressões booleanas

Compondo comandos

- Os comandos podem ser combinados de várias maneiras
- Vamos ver como algumas combinações são úteis para definir algumas técnicas básicas de programação

Laços com atribuições

- Nós utilizamos nas aulas passada uma composição de atribuições no escopo de um laço para resolver o problema da soma de uma entrada de números
- A técnica utilizada é denominada “Técnica do acumulador”

Técnica do acumulador

Begin

Soma := 0;

Read (n);

While n <> 0 do

Begin

Soma := Soma + n;

Read (n);

End;

Write (Soma);

End.

- A variável “soma” é um acumulador
- Ela é inicializada com zero antes do laço e durante o laço recebe atualizações
- A atribuição está no escopo do while

Laços com desvios

- Nós utilizamos na aula passada uma composição de desvio no escopo de um laço para resolver o problema de qual é o menor de uma entrada de números
- A técnica utilizada é denominada “Chutar e corrigir”

Chutar e corrigir

```
Begin
  Read (n);
  Menor:= n;
  While n <> 0 do
  Begin
    If n < menor then
      Menor:= n;
    Read (n);
  End;
  Write (menor);
End.
```

- À variável “menor” é atribuído um chute
- Durante o laço, testamos se o chute está errado e corrigimos. Ao final temos o menor
- O desvio está no escopo do while

Maneiras de compor laços e desvios

If $n \bmod 2 = 0$ then

While $n > 0$ do

$n := n - 1;$

If $n \bmod 2 = 0$ then

Write (n);

While $n > 0$ do

$n := n - 1;$

While $n > 0$ do

If $n \bmod 2 = 0$ then

$n := n - 1;$

While $n > 0$ do

$n := n + 1;$

If $n \bmod 2 = 0$ do

Write (n);

Maneiras de compor laços e desvios

```
Read (n);  
If n >= 0 then  
  Begin  
    Write (n);  
    While sqrt(n) < 50 do  
      n:= n + 1;  
  End;
```

```
Read (n);  
If n >= 0 then  
  Write (n);  
  While sqrt (n) < 50 do  
    n:= n + 1;
```

Maneiras de compor laços e desvios

```
Read (n);  
While n > 0 do  
Begin  
    n:= n - 1;  
    If n mod 2 = 1 then  
        Write (n);  
End;
```

```
Read (n);  
While n > 0 do  
    n:= n - 1;  
    If n mod 2 = 1 then  
        Write (n);
```

Composição de comandos

- O modo correto de compor laços com desvios depende do problema!
- Cada problema exige um dos 4 tipos estudados
- O importante é saber a diferença semântica entre as várias formas de composição

Compondo comandos

- Estudamos as seguintes composições:
- Laços com atribuições
- Laços com desvios
- Desvios com laços
- Agora estudaremos composição de desvios

Composição de desvios

- Podemos compor, ou aninhar, desvios condicionais.
- São estruturas com if aninhado sob outro if
- Exemplo: Dados 3 números, determinar o menor deles

Problema: O menor de três

- Dados três números inteiros, determinar qual é o menor deles usando apenas comparações

Problema: O menor de três

- Inicialmente, não se sabe nada sobre a ordem dos três números.
- Logo, é preciso compará-los dois a dois

Problema: O menor de três

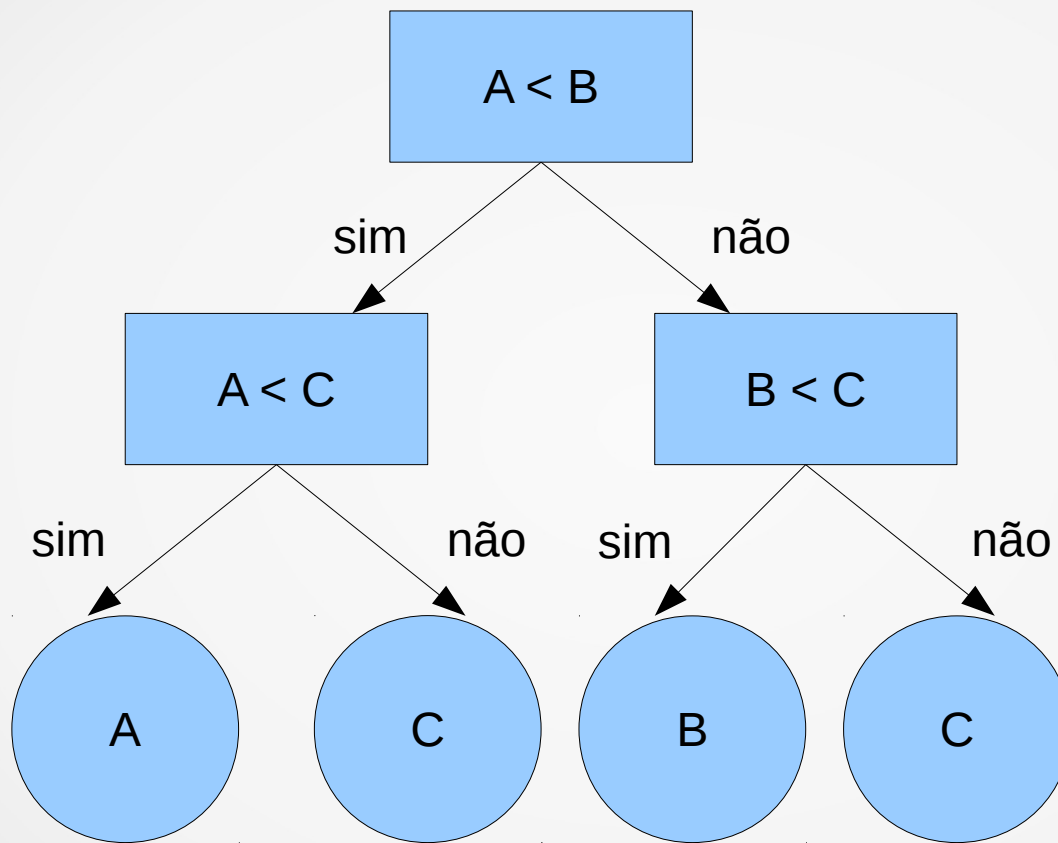
- A pode ser menor, igual ou maior do que B
- Suponhamos que A seja menor do que B
- Do ponto de vista lógico, podemos descartar que B seja o menor dos três
- Portanto, ou o menor é A ou o menor é C

Problema: O menor de três

- Vamos supor que A seja menor do que C
- Como já “sabíamos” que A era menor do que B, e agora “sabemos” que A é menor do que C, concluimos que A é o menor dos três
- O problema é que este raciocínio foi baseado em suposições. Devemos testar todos os casos!

Árvore de decisão

- É uma estrutura que contém todas as possíveis relações de decisão sobre os elementos envolvidos
- No caso do nosso problema, as relações de ordem entre a , b e c



Composição de desvios

If $A < B$ then

 If $A < C$ then

 Write (A)

 Else

 Write (C)

Else

 If $B < C$ then

 Write (B)

 Else

 Write (C)

Exercícios

- Ler trincas de números e para cada uma imprimir o menor deles. Terminar com a trinca 0, 0, 0
- Dado um número inteiro n , imprimí-lo em sua forma binária
- Dados dois números inteiros, imprimir o MDC entre eles