Noção de Computabilidade

Procedimento X Algoritmo

- **Procedimento**: sequência *finita* de *instruções*, que são operações *claramente descritas*, e que podem ser executadas *mecanicamente*, em tempo *finito*.

  - *claramente descritas:* não há dúvidas sobre o que deve ser feito;
  - *mecanicamente:* por "quem não pensa";
  - *em tempo finito:* não há dúvidas de que a tarefa correspondente à instrução pode, em qualquer caso, ser levada até sua conclusão.

- Exemplo 1:

  1. **se i = 0, pare e diga "sim".**
  2. **diminua o valor de i de duas unidades.**
  3. **vá para 1.**

Note que se i for originalmente ímpar ou negativo, o procedimento não para, embora cada instrução seja finita, conforme a definição. Esse procedimento não poderia servir de instrução para outro procedimento, pois não seria garantido seu término.

Procedimento X Algoritmo

- Exemplo 2:

1. se i = 0, pare e diga “sim”.
2. se i < 0, pare e diga “não”.
3. diminua o valor de i de duas unidades.
4. vá para 1.

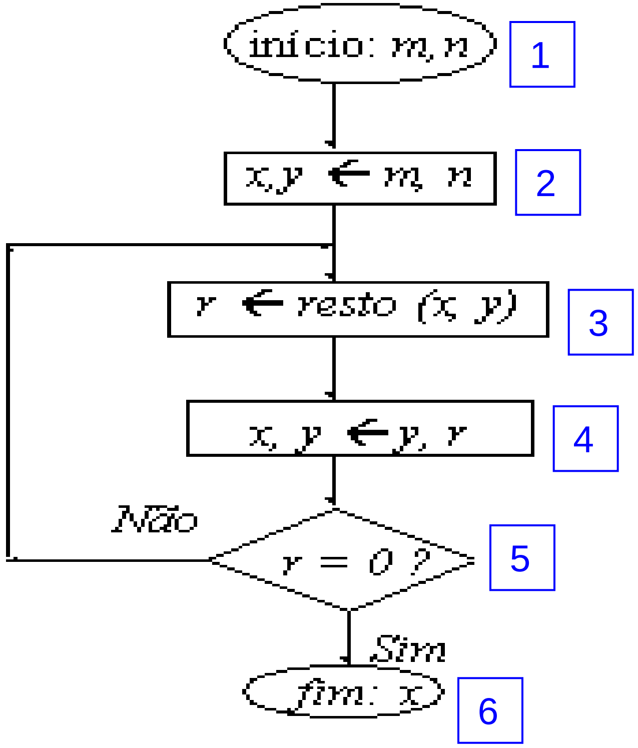
O procedimento acima para e diz “sim” se o número inteiro i for par e não negativo; para e diz “não” nos demais casos.

Algoritmo: é um procedimento que sempre para, quaisquer que sejam os valores de suas entradas.

Verificando o término de procedimentos

<u>EXEMPLO 1 - Algoritmo de Euclides</u>

Calcula o máximo divisor comum entre dois inteiros positivos m e n.



***Pergunta**:* Este procedimento termina quaisquer que sejam os valores dos dados de entrada?

Mostrar isto, neste exemplo, equivale a provar a seguinte proposição:
"Se, no passo 2 do procedimento, os valores de x e y são inteiros positivos, diferentes de zero, então os passos 2, 3 e 4 serão executados apenas um número finito de vezes, com os cálculos terminando no passo 4".

Demonstração por indução sobre o valor de y:

Prova por Indução:

- Método avançado para mostrar que todos os elementos de um conjunto infinito têm uma propriedade específica.
- Toda prova por indução consiste de 2 partes: a base e o passo de indução.
- Ex: tomemos os números naturais $\mathbb{N}$ e uma propriedade P. Nosso objetivo é provar que P(k) é verdade para todo número natural $k \in \mathbb{N}$.

Prova por Indução:

- Base: Provar que P(1) é verdadeiro.
- Passo de Indução: Para cada $1 \leq i < k$, assuma que P(i) é verdadeiro (hipótese de indução) e use esta suposição para mostrar que P(k) é verdadeiro.

Demonstração por indução sobre o valor de y:

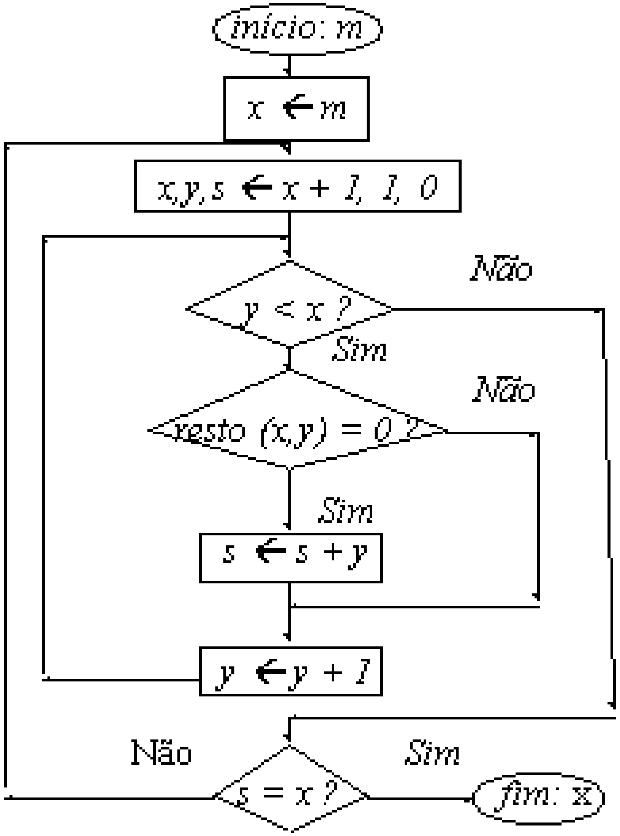
- (BASE) Se y = 1, então, após o passo 2, r = 0. Portanto, os passos 2, 3 e 4 são executados uma única vez e o cálculo termina no passo 4.
- (HIPÓSTESE DE INDUÇÃO) Suponhamos que a proposição é verdadeira para qualquer x > 0 e qualquer y, com $1 \leq y < k$, e demonstraremos que ela é verdadeira para y = k.

- Por definição do resto da divisão de inteiros positivos, teremos, após a execução do passo 2, $0 \leq r < k$. Se r = 0, então a execução termina, numa única vez. Se r > 0, com a execução dos passos 3 e 4, teremos x = k > 0 e y = r com $0 < r < k$, e a execução volta ao passo 2. Por hipótese de indução, como $1 \leq y < k$, os passos 2, 3 e 4 serão executados um número finito p de vezes, com os cálculos terminando no passo 4. Ao todo teremos, então, p+1 execuções para y = k.

- Conclui-se, então, que o Algoritmo de Euclides termina para quaisquer inteiros positivos m e n.

Verificando o término de procedimentos

<u>EXEMPLO 2 - </u>Procedimento para determinar o menor número perfeito maior do que um inteiro positivo m dado (*k é perfeito se for igual à soma de todos os seus divisores exceto o próprio k*).

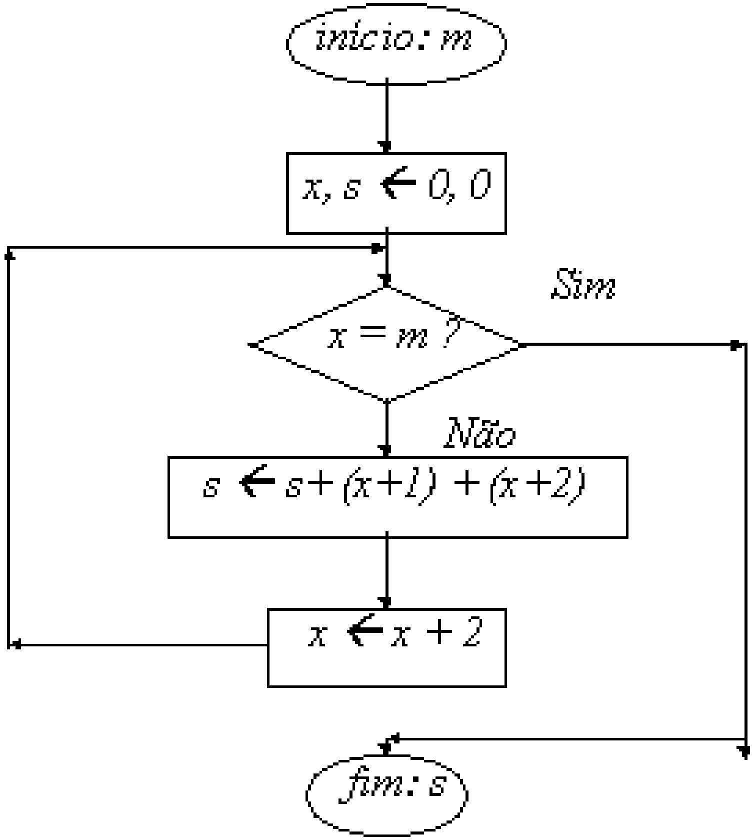


***Pergunta*: Este procedimento termina para qualquer valor de m?**

Resposta: Apenas para certos valores de m. Por exemplo, se m = 4, então ele para com x=6. Porém, no caso geral, a resposta não é conhecida, pois a existência ou não de um número infinito de números perfeitos é um problema em aberto. Se existirem infinitos números perfeitos, então a execução do procedimento termina para qualquer m; caso contrário, se K é o maior número perfeito, então o procedimento executa uma sequência infinita de cálculos para todo m $\geq$ K.

Verificando o término de procedimentos

<u>EXEMPLO 3 - </u> Cálculo de s = Σ_0^m (i) com m inteiro positivo.



Pergunta: Esse procedimento sempre para?

Resposta: Não. Termina apenas para valores pares de m, pois os valores da sequência são 0, 2, 4, 6,...
Para valores ímpares, a igualdade
x = m nunca é verdadeira e a execução do procedimento não termina.

Procedimento, Algoritmo e Computabilidade

- Para estudar o processo de computação do ponto de vista teórico, com a finalidade de caracterizar o que é ou não computável, é necessário introduzir um modelo que represente o que se entende por computação.

- Há vários modelos: máquinas de Turing, funções recursivas parciais, baseados em algoritmos de Markov, em linguagens de programações convencionais, etc.

- Conjectura de Alonzo Church: Todos os modelos razoáveis do processo de computação, definidos e por definir, são equivalentes (Tese de Church).

- Neste curso, usaremos a Máquina de Turing (MT) como modelo de computabilidade, ou seja, *Uma MT será capaz de executar uma função se e somente se essa função for computável, ou seja, passível de ser executada como um procedimento.*