

Análise da preparação de um algoritmo para a solução de um problema
Uso de três formas de representação: gráfica (fluxograma), linguagem natural (português estruturado) e programação (programa em Pascal)

Preliminares

A solução de qualquer problema requer uma abordagem sistemática que necessariamente passa por duas atitudes preliminares: (1º) reconhecimento do problema (leitura e interpretação detalhada), incluindo os dados existentes (entradas) e resultados esperados (saídas), e (2º) reconhecimento ou elaboração de um método para obter a solução. Um algoritmo é uma descrição que identifica e sequencia as etapas desse método.

Quando um novo método é elaborado, torna-se imprescindível testar a sua efetividade para alguns resultados conhecidos (obtidos por outros métodos, possivelmente manuais), considerando inclusive as eventuais condições extraordinárias.

Problema exemplo

Obter os N primeiros termos da série de Fibonacci¹, considerando que os dois primeiros termos são iguais a 1 e os demais termos são iguais a soma dos dois termos que os antecedem.

Abordagem metodológica

Embora possam ser encontradas semelhanças com uma PA (progressão aritmética), uma das primeiras lembranças de nossos estudos sobre séries de números, a determinação de um termo qualquer na série de Fibonacci não é tão simples quanto em uma PA². Para a determinação de um termo qualquer (a partir do terceiro), é necessário o conhecimento dos dois termos anteriores. Então, considerando que os dois primeiros termos são predeterminados por convenção, podemos modelar um método a partir das seguintes idéias:

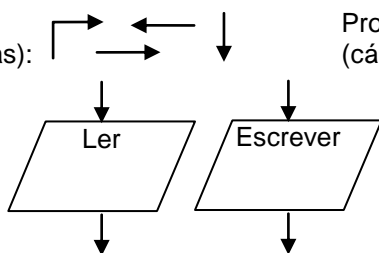
- a) usar dois indicadores (variáveis) para apontar (manter) os valores dos dois últimos termos anteriormente gerados - sejam **ant1** o último e **ant2** o penúltimo;
- b) gerar (calcular) os termos seguintes, um a um sucessivamente, usando uma variável prox para manter cada valor calculado (próximo), o qual deve ser exposto (saída), formando a sequência com os anteriores (já mostrados);
- c) ajustar os valores das variáveis ao final de cada passo para refletir a nova situação;
- d) controlar o número de passos para refletir a situação desejada (valor informado por entrada define a quantidade de termos).

Ordem:	1	2	3	4	5	6	... N
Série:	1	1	2	3	5	8	...
1º passo:	↑ ant2	↑ ant1	↑ prox				
2º passo:		↑ ant2	↑ ant1	↑ prox			
3º passo:			↑ ant2	↑ ant1	↑ prox		
4º passo:				↑ ant2	↑ ant1	↑ prox	
...							

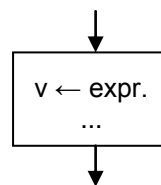
O esquema ao lado ilustra a dinâmica da geração sucessiva a partir do terceiro termo.

Simbologia para Fluxograma

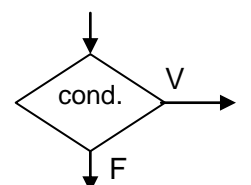
Sequência de operações (setas):



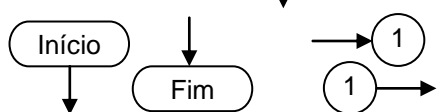
Processamento: (cálculos)



Tomada de decisão:



Terminais e conectores:

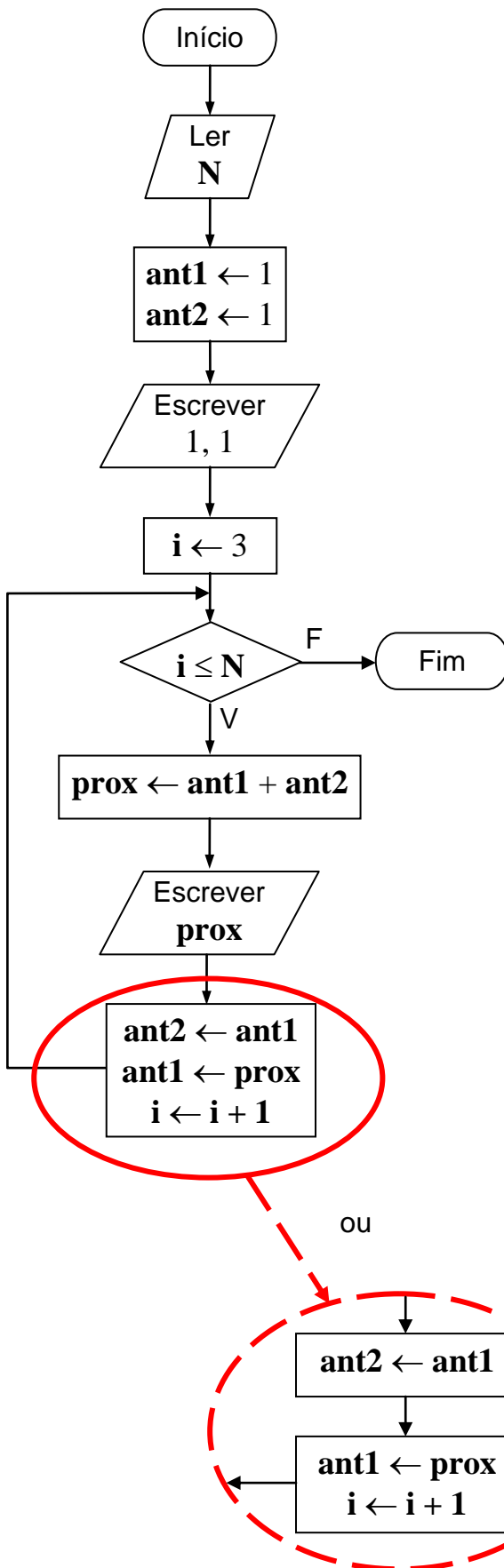


¹ Para satisfazer a curiosidade sobre a Série de Fibonacci, ver http://pt.wikipedia.org/wiki/Sequência_de_Fibonacci.

² Qualquer termo de uma PA pode ser determinado facilmente a partir do conhecimento de sua ordem na série (n), da razão ou diferença comum entre termos (r) e do primeiro termo da série (a₁), bastando aplicar $a_n = a_1 + r.(n-1)$.

Algoritmo preparado

Forma de Fluxograma:



Forma de Português estruturado:

a) Baseado no modelo do livro do Jaime Evaristo:

1. Leia o valor de N .
2. Faça $ant1 = 1$ e $ant2 = 1$.
3. Escreva os valores de $ant1$ e $ant2$.
4. Faça $i = 3$.
5. Enquanto $i \leq N$ repita as instruções 5.1 a 5.5.
 - 5.1. Calcule $prox = ant1 + ant2$.
 - 5.2. Escreva o valor de $prox$.
 - 5.3. Faça $ant2 = ant1$.
 - 5.4. Faça $ant1 = prox$.
 - 5.5. Incremente o valor de i .

b) Baseado no modelo do livro do Forbellone:

início

inteiro: $N, i, ant1, ant2, prox$

leia (N)

$ant1 \leftarrow 1$

$ant2 \leftarrow 1$

escreva ($ant1, ant2$)

$i \leftarrow 3$

enquanto $i \leq N$ **faça**

$prox \leftarrow ant1 + ant2$

escreva ($prox$)

$ant2 \leftarrow ant1$

$ant1 \leftarrow prox$

$i \leftarrow i + 1$

fimenquanto

fim.

Esta forma alternativa é mais adequada, pois destaca que as ações " $ant2 \leftarrow ant1$ " e " $ant1 \leftarrow prox$ " devem ser executadas estritamente nesta ordem. *Qual seria a ocorrência indesejada se esta ordem não fosse obedecida?*

Observe-se que as duas ações dentro do segundo bloco poderiam ser consideradas sem ordem específica entre elas, já que a ordem é especificada pelas setas.

Forma de Programa na linguagem Pascal:

```
// -----
// Programa que mostra os N primeiros termos da série de Fibonacci.
// Nesta série, os dois primeiros termos são iguais a 1 e os demais
// termos são iguais a soma dos dois termos que os antecedem.
// O valor de N deve ser fornecido pelo usuário.
//
// Autor: Exemplo da disciplina de Algoritmos e Programação (2016/1)
// -----

program Fibonacci;
var ant1, ant2, prox : integer ;
    i : integer;
    N : integer;
begin
    // solicita o número de elementos da série
    write('Informe o valor de N: ');
    readln(N);

    // Imprime primeiros dois elementos da série
    ant1:= 1;
    ant2:= 1;
    write('1 1');      { ou write(ant1, ' ', ant2); }

    // Cálculo da série
    i:= 3;
    while i <= N do
        begin
            prox:= ant1 + ant2;
            write(' ', prox);
            ant2:= ant1;
            ant1:= prox;
            i:= i + 1;
        end;

    readln;
end.
```

Observações:

- 1) Programa escrito com o modelo de "indentação" recomendado pelo professor.
- 2) Palavras reservadas do Pascal foram destacadas em negrito para fins didáticos apenas.
- 3) Recursos destacados neste exemplo:
 - estrutura de um programa;
 - comentários;
 - identificadores, variáveis, tipos de dados, números e *strings* (cadeias de caracteres);
 - declaração de variáveis;
 - comandos de atribuição;
 - comandos de entrada e de saída;
 - comando de repetição;
 - uso de contador (inicialização, incremento e verificação de limite);
 - bloco de comandos (comando composto).

Variações do problema:

Muitas vezes, a solução de um problema é encaminhada de forma equivocada, geralmente como decorrência de falha ou desatenção nas atitudes relacionadas à abordagem preliminar, ou seja, na interpretação do problema e na escolha dos métodos associados. Isto pode conduzir à obtenção de um "belo" algoritmo que resolve um problema diferente do apresentado.

Tomando como referência o problema inicialmente proposto, eis aqui dois outros problemas alternativos muito similares ao original (que dependem da mesma abordagem metodológica), porém com resultados esperados bem diferentes:

- a) *Obter o N-ésimo termo (termo de ordem N) da série de Fibonacci, considerando que os dois primeiros termos são iguais a 1 e os demais termos são iguais a soma dos dois termos que os antecedem;*
- b) *Obter o primeiro termo da série de Fibonacci que é maior do que N, considerando que os dois primeiros termos são iguais a 1 e os demais termos são iguais a soma dos dois termos que os antecedem.*

Exercícios:

1) **Prepare versões alteradas do algoritmo original para resolver os problemas dos casos "a" e "b", compare e entenda bem as diferenças entre os 3 casos.**

2) **Produza uma versão alterada do algoritmo correspondente ao caso "a", providenciando um controle para validar a solicitação, limitando a aceitação do valor de N aos valores viáveis – $1 \leq N < 47$ – já que o sistema é incapaz de armazenar o 47º termo da série (pela limitação dos números inteiros armazenados em 4 bytes).**

3) **Escolha um dos casos (um que seja adequado) e altere o programa correspondente, empregando o comando "FOR .. DO" em substituição ao comando "WHILE .. DO".**