

Introdução à Ciência da Computação

Módulo 3

Luis Retondaro

CEFET-RJ
Campus Petrópolis

September 6, 2023



Sumário

- 1 Elaborando Algoritmos
- 2 Exercícios
- 3 Estruturas de dados homogêneas
- 4 Exercícios

Elaborando Algoritmos



“A correta especificação do problema, a análise dos requisitos de entrada e saída, o reconhecimento de padrões e a documentação detalhada são as principais ferramentas do programador. ”

Preparação

- Leia e releia as especificações do problema
- Identifique variáveis e constantes
 - Especial: contadores e acumuladores
- Identificar os padrões
 - Observando a ação real dos eventos
 - Elencando rotinas comumente empregadas
- Identificar "Loops"
- Determinar o critério de parada
 - Pelo usuário?
 - Por um "flag"?
 - Por um evento?

1. Preparação (Exemplo 1): série de Fibonacci

1, 1, 2, 3, 5, 8, 13, 21, ...

Como é gerada?



Preparação (Exemplo 1): série de Fibonacci

Como é gerada?

- Cada número é produzido somando-se os seus 2 antecessores **[PADRÃO]**

Preparação (Exemplo 1): série de Fibonacci

Como é gerada?

- Cada número é produzido somando-se os seus 2 antecessores [**PADRÃO**]

Como iniciar?



Preparação (Exemplo 1): série de Fibonacci

Como é gerada?

- Cada número é produzido somando-se os seus 2 antecessores [**PADRÃO**]

Como iniciar?

- Reservando os dois primeiros elementos: 1, 1 [**a = 1, b = 1**]

Preparação (Exemplo 1): série de Fibonacci

Como é gerada?

- Cada número é produzido somando-se os seus 2 antecessores [**PADRÃO**]

Como iniciar?

- Reservando os dois primeiros elementos: 1, 1 [**a = 1, b = 1**]

Há loops?

-

Preparação (Exemplo 1): série de Fibonacci

Como é gerada?

- Cada número é produzido somando-se os seus 2 antecessores [**PADRÃO**]

Como iniciar?

- Reservando os dois primeiros elementos: 1, 1 [**a = 1, b = 1**]

Há loops?

- O elemento correntemente calculado passará a ser o último elemento e devemos guardar o seu antecessor também, pensando no cálculo seguinte [**a = b, b = calc**]
- O processo anterior se repete infinitamente.



Preparação (Exemplo 1): série de Fibonacci

Como é gerada?

- Cada número é produzido somando-se os seus 2 antecessores [**PADRÃO**]

Como iniciar?

- Reservando os dois primeiros elementos: 1, 1 [**a = 1, b = 1**]

Há loops?

- O elemento correntemente calculado passará a ser o último elemento e devemos guardar o seu antecessor também, pensando no cálculo seguinte [**a = b, b = calc**]
- O processo anterior se repete infinitamente.

Como finalizar?



Preparação (Exemplo 1): série de Fibonacci

Como é gerada?

- Cada número é produzido somando-se os seus 2 antecessores [**PADRÃO**]

Como iniciar?

- Reservando os dois primeiros elementos: 1, 1 [**a = 1, b = 1**]

Há loops?

- O elemento correntemente calculado passará a ser o último elemento e devemos guardar o seu antecessor também, pensando no cálculo seguinte [**a = b, b = calc**]
- O processo anterior se repete infinitamente.

Como finalizar?

- A série é infinita, então maneira é "forçar" a parada indicando o número de elementos que desejamos calcular, por exemplo.



2. Descrição

- inicialize variáveis de controle (flag), acumuladores, contadores, etc.
- Escreva as atribuições passo a passo
- Caso não "visualize" o loop com naturalidade, escreva os passos repetidos sequencialmente
- Reescreva os passos com a menor quantidade de instruções possível



Descrição (Exemplo 1): série de Fibonacci

```
a = 1    # Primeiro elemento
b = 1    # Segundo elemento
flag = 10 # Supondo 10 elemento para calcular
contador = 2 # Já contamos os 2 elementos iniciais
escreva(a,b)
enquanto (contador < flag) faça
    calc = a + b
    a = b
    b = calc
    contador = contador + 1 # Incrementa o contador
escreva(calc)
fim_enquanto
```

3. Teste de Mesa, Depuração, rastreamento, "chinês"

- Elaborar tabela
- Variáveis (colunas)
- Iterações (linhas)
- Seguir o fluxo de execução da rotina
- Anotar saídas em separado

Teste de Mesa (Exemplo 1): série de Fibonacci

flag	contador	a	b	calc
10	2	1	1	-
10	3	1	1	2
10	4	1	2	3
10	5	2	3	5
10	6	3	5	8
10	7	5	8	13
10	8	8	13	21
10	9	13	21	34
10	10	21	34	55

Saída:

1 1 2 3 5 8 13 21 34 55

Solução passo a passo:

Incluindo fluxograma

- Média de N termos
- Fatorial de N

Exercícios



Exercícios

Elabore os seguinte algoritmos:

- 1 Calcule a exponenciação em relação a um valor base e expoente lidos do usuário.
- 2 Escreva todos os múltiplos de 5 entre 1 e 100.
- 3 Identifique se um determinado número lido é par ou ímpar.
- 4 Verificar se uma pessoa tem idade para votar, com a seguinte regra: Menor de 16 não pode votar; 16, 17 e maior que 70, o voto é facultativo; de 18 a 70 o voto é obrigatório.

Estruturas de dados homogêneas

Estrutura de dados homogênea

Vetor

- Elementos do mesmo tipo
- Exemplo de declaração: **inteiro: VET [10]**
- Alocação sequencial em memória

Exemplo:

VET =

12	30	7	18	4	-2	0	-11	5	92
0	1	2	3	4	5	6	7	8	9

índices

VET[5] == -2

Lê-se: VET índice 5 é igual a -2
ou
o sexto elemento de VET é igual a -2

Vetor (Array)

Notações

- O índice do último elemento é sempre igual ao tamanho do vetor -1
- O primeiro elemento do vetor equivale ao índice 0
- É comum usar variáveis auxiliares para indexar uma posição do vetor.

Por exemplo:

```
inteiro: i, VET[10];
```

```
    i = 4;
```

```
    VET [i] = 3;
```

Equivale a:

posição 4 de VET recebe o valor 3;

Vetor multidimensional (matriz)

Exemplo: Array Bidimensional

- Tipicamente: linha x coluna

Exemplo:

```
inteiro: i, j, MATRIZ[3][4];
```

```
  i = 2;
```

```
  j = 3;
```

```
  MATRIZ[i][j] = 5;
```

Isto produzirá:

MATRIZ =

	0	1	2	3
0				
1				
2				5

Inicializando o vetor

Para facilitar, ao invés de inicializarmos cada elemento de um vetor através do seu índice, podemos utilizar um comando **construtor**

Assim:

Ao invés de:

VET[0] = 3

VET[1] = 5

VET[2] = -1

VET[3] = 8

Podemos fazer:

VET = {3,5,-1,8}

Confira o resultado

Qual será o conteúdo da variável CRR depois de executados os comandos:

caracter: CRR = {"!", "U", "O", "T", "R", "E", "C", "A"}

para i **de** 1 **até** 3 **passo** 1

faça

AUX = CRR[i]

CRR[i] = CRR[7-i]

CRR[7-i] = AUX

fim_para

AUX = CRR[0]

CRR[0] = CRR[7]

CRR[7] = AUX

Confira o resultado

Solução:

CRR =

A	C	E	R	T	O	U	!
0	1	2	3	4	5	6	7

O que faz o algoritmo abaixo?

início

inteiro: V[12], AUX, BOLHA, LSUP, J;

V = {0, 7, 5, -3, 40, 2, 11, 30, -9, 8, 1, 10};

LSUP = 12;

enquanto LSUP > 0 **faça**

BOLHA = 0;

para J **de** 0 **até** LSUP-1 **faça**

se VET[J] > VET[J+1] **então**

AUX = VET[J];

VET[J] = VET[J+1];

VET[J+1] = AUX;

BOLHA = J;

fim_se

fim_para

LSUP = BOLHA;

fim_enquanto

escreva(VET);

Fim.

Exercícios



Exercícios

Elabore os seguintes algoritmos:

- 5 Dado um vetor de 100 elementos, verificar se existe um elemento igual a K no vetor e imprimir a posição onde ele está; Se não, imprimir: "valor não encontrado!"
- 6 Faça um programa que, dado um conjunto de N números, determine o menor valor, o maior valor e a soma dos valores.
- 7 Numa eleição existem três candidatos. Faça um programa que peça o número total de eleitores. Peça para cada eleitor votar e ao final mostrar o número de votos de cada candidato.
- 8 Dado um tabuleiro de xadrez, com a partida iniciada, indique a quantidade de peças de cada tipo. A saída deverá ser por exemplo: 3 peões brancos, 1 Rei branco, 1 torre branca, 4 peões pretos, 1 rei preto, etc...
 - Para facilitar, considere cada peça como um número (1 – PEÕES,2 – CAVALOS,3 – TORRES,4 – BISPOS,5 – REIS,6 – RAINHAS e 0 – AUSÊNCIA DE PEÇAS)