

CI1055: Algoritmos e Estruturas de Dados I

Profs. Drs. Marcos Castilho, Carmem Hara e Bruno Müller Jr

Departamento de Informática/UFPR

3 de agosto de 2020

Aplicação das técnicas elementares

- ▶ (*) Fatorial
- ▶ Fibonacci revisado
- ▶ Palíndromos

Fatorial

Problema: Imprimir o valor do fatorial de todos os números entre 1 e n , sendo n fornecido pelo usuário.

- ▶ quais as semelhanças deste com os já vistos?
- ▶ (subproblema) como calcula o fatorial de n ?

$$fat(n) = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

- ▶ aplicar a técnica de acumuladores;
- ▶ ideia: colocar a solução para o subproblema em um laço;
- ▶ primeiro passo: implementar o subproblema:

Fatorial

Problema: Imprimir o valor do fatorial de todos os números entre 1 e n , sendo n fornecido pelo usuário.

- ▶ quais as semelhanças deste com os já vistos?
- ▶ (subproblema) como calcula o fatorial de n ?

$$fat(n) = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

- ▶ aplicar a técnica de acumuladores;
- ▶ ideia: colocar a solução para o subproblema em um laço;
- ▶ primeiro passo: implementar o subproblema:

Fatorial

Problema: Imprimir o valor do fatorial de todos os números entre 1 e n , sendo n fornecido pelo usuário.

- ▶ quais as semelhanças deste com os já vistos?
- ▶ (subproblema) como calcula o fatorial de n ?

$$fat(n) = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

- ▶ aplicar a técnica de acumuladores;
- ▶ ideia: colocar a solução para o subproblema em um laço;
- ▶ primeiro passo: implementar o subproblema:

Fatorial

Problema: Imprimir o valor do fatorial de todos os números entre 1 e n , sendo n fornecido pelo usuário.

- ▶ quais as semelhanças deste com os já vistos?
- ▶ (subproblema) como calcula o fatorial de n ?

$$fat(n) = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

- ▶ aplicar a técnica de acumuladores;
- ▶ ideia: colocar a solução para o subproblema em um laço;
- ▶ primeiro passo: implementar o subproblema:

Fatorial

Problema: Imprimir o valor do fatorial de todos os números entre 1 e n , sendo n fornecido pelo usuário.

- ▶ quais as semelhanças deste com os já vistos?
- ▶ (subproblema) como calcula o fatorial de n ?

$$fat(n) = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

- ▶ aplicar a técnica de acumuladores;
- ▶ ideia: colocar a solução para o subproblema em um laço;
- ▶ primeiro passo: implementar o subproblema:

Fatorial de n

$$\text{fat}(5) = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

```
program fatorial ;
var i , n, fat : integer ;
begin
  read (n) ;
  fat:= 1; (* inicializacao do acumulador *)
  i:= n;
  write ('fat( ',n,')= ');
  while i >= 1 do
  begin
    fat:= fat * i ;
    if i > 1 then
      write ( i , 'x' )
    else
      write ( i , '= ');
    i:= i - 1;
  end;
  writeln ( fat ) ;
end.
```


Fatorial de todos até n

```
program fatorial1_n ;
var cont, i, n, fat : integer ;
begin
  read (n);
  cont:= 1;
  while cont <= n do
  begin
    write ('fat(' ,cont ,')= ');
    fat:= 1; (* inicializacao do acumulador *)
    i:= cont;
    while i >= 1 do
    begin
      fat:= fat * i ;
      if i > 1 then
        write ( i , 'x' )
      else
        write ( i , '= ' ) ;
      i:= i - 1;
    end;
    writeln ( fat ) ;
    cont:= cont + 1;
  end;
end.
```

Fatorial de todos até n

Saída do programa com entrada n=7:

fat(1)= 1= 1

fat(2)= 2x1= 2

fat(3)= 3x2x1= 6

fat(4)= 4x3x2x1= 24

fat(5)= 5x4x3x2x1= 120

fat(6)= 6x5x4x3x2x1= 720

fat(7)= 7x6x5x4x3x2x1= 5040

Uma novo ângulo

- ▶ Em muitos casos, é possível escrever programas mais simples e eficientes olhando para o problema por outro ângulo;
- ▶ O programa foi desenvolvido a partir da fórmula abaixo:

$$fat(n) = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

- ▶ Porém, outra forma de obter este mesmo resultado é:

$$fat(n) = n \times fat(n - 1)$$

```
fat(1)= 1= 1
fat(2)= 2xfat(1)= 2
fat(3)= 3xfat(2)= 6
fat(4)= 4xfat(3)= 24
fat(5)= 5xfat(4)= 120
fat(6)= 6xfat(5)= 720
fat(7)= 7xfat(6)= 5040
```

Uma novo ângulo

- ▶ Em muitos casos, é possível escrever programas mais simples e eficientes olhando para o problema por outro ângulo;
- ▶ O programa foi desenvolvido a partir da fórmula abaixo:

$$\text{fat}(n) = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

- ▶ Porém, outra forma de obter este mesmo resultado é:

$$\text{fat}(n) = n \times \text{fat}(n - 1)$$

```
fat(1)= 1= 1
fat(2)= 2xfat(1)= 2
fat(3)= 3xfat(2)= 6
fat(4)= 4xfat(3)= 24
fat(5)= 5xfat(4)= 120
fat(6)= 6xfat(5)= 720
fat(7)= 7xfat(6)= 5040
```

Uma novo ângulo

- ▶ Em muitos casos, é possível escrever programas mais simples e eficientes olhando para o problema por outro ângulo;
- ▶ O programa foi desenvolvido a partir da fórmula abaixo:

$$fat(n) = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$$

- ▶ Porém, outra forma de obter este mesmo resultado é:

$$fat(n) = n \times fat(n - 1)$$

```
fat(1)= 1= 1
fat(2)= 2xfat(1)= 2
fat(3)= 3xfat(2)= 6
fat(4)= 4xfat(3)= 24
fat(5)= 5xfat(4)= 120
fat(6)= 6xfat(5)= 720
fat(7)= 7xfat(6)= 5040
```

Uma novo ângulo

- ▶ Este novo ângulo sugere reaproveitar os valores calculados, produzindo o programa abaixo:

```
rogram fatorial1_n_v2 ;
var cont , n, fat : integer ;
begin
  read (n) ;
  cont:= 1;
  fat:= 1; (* inicializacao do acumulador *)
  while cont <= n do
    begin
      fat:= fat * cont ;
      writeln ('fat(' ,cont ,')= ' , fat ) ;
      cont:= cont + 1;
    end;
end.
```

Fim

- ▶ este material está no livro no capítulo 7, seção 7.5

Licença

- ▶ Slides feitos em \LaTeX usando beamer
- ▶ Licença

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>