

CI1055: Algoritmos e Estruturas de Dados I

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

Departamento de Informática/UFPR

21 de outubro de 2021

Aplicação das técnicas elementares

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

Fibonacci revisado

- ▶ Muitas vezes, o problema a ser resolvido é semelhante a um problema já implementado.
- ▶ Esta seção apresenta problemas que podem ser resolvidos ao modificar um programa já desenvolvido (neste caso fibonacci):
 - ▶ qual é o primeiro número de Fibonacci maior do que um determinado valor;
 - ▶ o número áureo;

Fibonacci revisado

- ▶ Muitas vezes, o problema a ser resolvido é semelhante a um problema já implementado.
- ▶ Esta seção apresenta problemas que podem ser resolvidos ao modificar um programa já desenvolvido (neste caso fibonacci):
 - ▶ qual é o primeiro número de Fibonacci maior do que um determinado valor;
 - ▶ o número áureo;

Fibonacci revisado

- ▶ Muitas vezes, o problema a ser resolvido é semelhante a um problema já implementado.
- ▶ Esta seção apresenta problemas que podem ser resolvidos ao modificar um programa já desenvolvido (neste caso fibonacci):
 - ▶ qual é o primeiro número de Fibonacci maior do que um determinado valor;
 - ▶ o número áureo;

Alterando o critério de parada

Problema: imprimir o primeiro número de Fibonacci maior do que um determinado valor

```
program Fibonacci ;
var ultimo , penultimo , soma, cont , n: integer ;
begin
  read (n) ; (* define quantos numeros serao impressos *)
  ultimo:= 1; (* inicializacao das variaveis principais *)
  penultimo:= 1;
  writeln (penultimo) ; (* imprime os dois primeiros valores *)
  writeln (ultimo) ;
  cont:= 3 ; (* calcula do terceiro em diante *)
  while cont <= n do
  begin
    soma:= penultimo + ultimo ;
    writeln (cont , ' ' ,soma) ;
    penultimo:= ultimo ; (* a ordem destes dois comandos *)
    ultimo:= soma;      (* eh relevante no codigo *)
    cont:= cont + 1;
  end;
end.
```

Alterando o critério de parada

Problema: imprimir o primeiro número de Fibonacci maior do que um determinado valor

```
program Fibonacci_2 ;
const max=1000;
var ultimo , penultimo , soma: integer ;
begin
  ultimo:= 1;
  penultimo:= 1;
  soma:= penultimo + ultimo ;
  while soma <= max do (* termina quando atinge o valor desejado *)
  begin
    penultimo:= ultimo ;
    ultimo:= soma;
    soma:= penultimo + ultimo ;
  end;
  writeln (soma) ;
end.
```

O número áureo

- ▶ razão entre dois termos consecutivos converge para um número irracional conhecido como número áureo, denotado pela letra grega φ , e é aproximadamente 1.6180339887499:

$$\frac{1}{1} = 1, \frac{2}{1} = 2, \frac{3}{2} = 1.5, \frac{5}{3} = 1.66,$$

$$\frac{8}{5} = 1.60, \frac{13}{8} = 1.625, \frac{21}{13} = 1.615 \dots$$

- ▶ Convergência do número: os número intermediários obtidos nos cálculos estão cada vez mais próximos uns dos outros, o que nos leva a ter a oportunidade de definirmos, por exemplo, com quantas casas decimais estaremos satisfeitos.

O número áureo

- ▶ razão entre dois termos consecutivos converge para um número irracional conhecido como número áureo, denotado pela letra grega φ , e é aproximadamente 1.6180339887499:

$$\frac{1}{1} = 1, \frac{2}{1} = 2, \frac{3}{2} = 1.5, \frac{5}{3} = 1.66,$$

$$\frac{8}{5} = 1.60, \frac{13}{8} = 1.625, \frac{21}{13} = 1.615 \dots$$

- ▶ Convergência do número: os número intermediários obtidos nos cálculos estão cada vez mais próximos uns dos outros, o que nos leva a ter a oportunidade de definirmos, por exemplo, com quantas casas decimais estaremos satisfeitos.

O número áureo

Problema: utilizar a sequência de fibonacci para imprimir o número áureo com a precisão desejada.

- ▶ Precisão desejada?
- ▶ Convergência: $|aureo(n) - aureo(n - 1)| \leq precisao$
- ▶ Pode não convergir, por exemplo precisão 10^{-10000} ;
- ▶ Incluir regra alternativa, por exemplo: se o laço executar um certo número de vezes, digamos, mil, e não chegar no erro estabelecido, então o programa deve encerrar com uma mensagem de erro informando esta situação (exercício).

O número áureo

Problema: utilizar a sequência de fibonacci para imprimir o número áureo com a precisão desejada.

- ▶ Precisão desejada?
- ▶ Convergência: $|aureo(n) - aureo(n - 1)| \leq precisao$
- ▶ Pode não convergir, por exemplo precisão 10^{-10000} ;
- ▶ Incluir regra alternativa, por exemplo: se o laço executar um certo número de vezes, digamos, mil, e não chegar no erro estabelecido, então o programa deve encerrar com uma mensagem de erro informando esta situação (exercício).

O número áureo

Problema: utilizar a sequência de fibonacci para imprimir o número áureo com a precisão desejada.

- ▶ Precisão desejada?
- ▶ Convergência: $|aureo(n) - aureo(n - 1)| \leq precisao$
- ▶ Pode não convergir, por exemplo precisão 10^{-10000} ;
- ▶ Incluir regra alternativa, por exemplo: se o laço executar um certo número de vezes, digamos, mil, e não chegar no erro estabelecido, então o programa deve encerrar com uma mensagem de erro informando esta situação (exercício).

O número áureo

Problema: utilizar a sequência de fibonacci para imprimir o número áureo com a precisão desejada.

- ▶ Precisão desejada?
- ▶ Convergência: $|aureo(n) - aureo(n - 1)| \leq precisao$
- ▶ Pode não convergir, por exemplo precisão 10^{-10000} ;
- ▶ Incluir regra alternativa, por exemplo: se o laço executar um certo número de vezes, digamos, mil, e não chegar no erro estabelecido, então o programa deve encerrar com uma mensagem de erro informando esta situação (exercício).

O número áureo

Problema: utilizar a sequencia de fibonacci para imprimir o número áureo com a precisão desejada

```
program Fibonacci ;
var ultimo , penultimo , soma, cont , n: integer ;
begin
  read (n) ; (* define quantos numeros serao impressos *)
  ultimo:= 1; (* inicializacao das variaveis principais *)
  penultimo:= 1;
  writeln (penultimo) ; (* imprime os dois primeiros valores *)
  writeln (ultimo) ;
  cont:= 3 ; (* calcula do terceiro em diante *)
  while cont <= n do
  begin
    soma:= penultimo + ultimo ;
    writeln (cont , ' ' ,soma) ;
    penultimo:= ultimo ; (* a ordem destes dois comandos *)
    ultimo:= soma;      (* eh relevante no codigo *)
    cont:= cont + 1;
  end;
end.
```

O número áureo

```
program numero aureo ;
const PRECISAO=0.000000000000001;
var ultimo , penultimo , soma: integer ;
    naureo , naureo_anterior : real ;
begin
    ultimo:= 1;    (* inicializacao das variaveis principais *)
    penultimo:= 1;
    naureo_anterior:= -1; (* para funcionar o primeiro teste *)
    naureo:= 1;
    writeln (naureo:15:14) ;
    (* calcula do terceiro em diante *)
    while abs(naureo - naureo_anterior ) >= PRECISAO do
    begin
        soma:= penultimo + ultimo ;
        naureo_anterior:= naureo ;
        naureo:= soma/ultimo ;
        writeln (naureo:15:14) ;
        penultimo:= ultimo ;
        ultimo:= soma;
    end;
end.
```

Fim

- ▶ este material está no livro no capítulo 7, seção 7.6

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/>