

Aula 14

Programação Dinâmica

Sequência de Fibonacci

Projeto e Análise de Algoritmos

Professor Eurinardo Rodrigues Costa
Universidade Federal do Ceará
Campus Russas

2021.1

Aulas Passadas

Programação Dinâmica

Sequência de Fibonacci - Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação Dinâmica

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Aulas Passadas

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

- ▶ Técnicas para construir algoritmos:

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

- ▶ Técnicas para construir algoritmos:
 - ▶ Algoritmos Iterativos

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

- ▶ Técnicas para construir algoritmos:
 - ▶ Algoritmos Iterativos
 - ▶ Algoritmos Recursivos

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

- ▶ Técnicas para construir algoritmos:
 - ▶ Algoritmos Iterativos
 - ▶ Algoritmos Recursivos
 - ▶ Divisão e Conquista

Programação Dinâmica

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

**Programação
Dinâmica**

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Definição (Programação Dinâmica)

Definição (Programação Dinâmica)

Seguir os passos:

Definição (Programação Dinâmica)

Seguir os passos:

(1) *Verificar propriedade de **subestrutura ótima***

Definição (Programação Dinâmica)

Seguir os passos:

- (1) *Verificar propriedade de **subestrutura ótima***
- (2) *Obter uma recursão para o **valor ótimo** do problema*

Definição (Programação Dinâmica)

Seguir os passos:

- (1) *Verificar propriedade de **subestrutura ótima***
- (2) *Obter uma recursão para o **valor ótimo** do problema*
- (3) *Obter um algoritmo Bottom-UP para calcular o **valor ótimo***

Definição (Programação Dinâmica)

Seguir os passos:

- (1) *Verificar propriedade de **subestrutura ótima***
- (2) *Obter uma recursão para o **valor ótimo** do problema*
- (3) *Obter um algoritmo Bottom-UP para calcular o **valor ótimo***
- (4) *Obter uma **solução ótima** do problema.*

Sequência de Fibonacci

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Sequência de Fibonacci

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Definição (Sequência de Fibonacci)

Sequência de Fibonacci

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Definição (Sequência de Fibonacci)

n	0	1	2	3	4	5	6	7	8	9	...
$F(n)$	0	1	1	2	3	5	8	13	21	34	...

Sequência de Fibonacci

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Definição (Sequência de Fibonacci)

n	0	1	2	3	4	5	6	7	8	9	...
$F(n)$	0	1	1	2	3	5	8	13	21	34	...

Problema

Sequência de Fibonacci

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Definição (Sequência de Fibonacci)

n	0	1	2	3	4	5	6	7	8	9	...
$F(n)$	0	1	1	2	3	5	8	13	21	34	...

Problema

Calcular o Finacci de um número n .

Sequência de Fibonacci

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Definição (Sequência de Fibonacci)

n	0	1	2	3	4	5	6	7	8	9	...
$F(n)$	0	1	1	2	3	5	8	13	21	34	...

Problema

Calcular o Finacci de um número n .

Recursão

Sequência de Fibonacci

Definição (Sequência de Fibonacci)

n	0	1	2	3	4	5	6	7	8	9	...
$F(n)$	0	1	1	2	3	5	8	13	21	34	...

Problema

Calcular o Finacci de um número n .

Recursão

$$F(n) = \begin{cases} 0 & , \text{ se } n = 0 \\ 1 & , \text{ se } n = 1 \\ F(n-1) + F(n-2) & , \text{ se } n > 1 \end{cases}$$

Sequência de Fibonacci - Recursivo

EDA - Aula 14

Prof. Eurinardo

Recursão

$$F(n) = \begin{cases} 0 & , \text{ se } n = 0 \\ 1 & , \text{ se } n = 1 \\ F(n-1) + F(n-2) & , \text{ se } n > 1 \end{cases}$$

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema
Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Recursão

$$F(n) = \begin{cases} 0 & , \text{ se } n = 0 \\ 1 & , \text{ se } n = 1 \\ F(n-1) + F(n-2) & , \text{ se } n > 1 \end{cases}$$

Algoritmo 2: Fib-Rec(n)

Entrada: n : número inteiro ≥ 0

Saída: Fibonacci de n

```
1 se  $n = 0$  então
2   |   retorne 0
3 se  $n = 1$  então
4   |   retorne 1
5 retorne Fib-Rec( $n - 1$ ) + Fib-Rec( $n - 2$ )
```

Aulas Passadas

Programação
DinâmicaSequência de
Fibonacci -
ProblemaProblema
Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Recursão

$$F(n) = \begin{cases} 0 & , \text{ se } n = 0 \\ 1 & , \text{ se } n = 1 \\ F(n-1) + F(n-2) & , \text{ se } n > 1 \end{cases}$$

Algoritmo 3: Fib-Rec(n)

Entrada: n : número inteiro ≥ 0

Saída: Fibonacci de n

```
1 se  $n = 0$  então
2   |   retorne 0
3 se  $n = 1$  então
4   |   retorne 1
5 retorne Fib-Rec( $n - 1$ ) + Fib-Rec( $n - 2$ )
```

Tempo $\Omega(\sqrt{2}^n)$

Aulas Passadas

Programação
DinâmicaSequência de
Fibonacci -
ProblemaProblema
Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Sequência de Fibonacci - Memoização

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Sequência de Fibonacci - Memoização

EDA - Aula 14

Prof. Eurinardo

Algoritmo 6: Fib-Mem(n)

Entrada: n : número inteiro ≥ 0

Saída: Fibonacci de n

- 1 **se** $n \leq 1$ **então retorne** n ;
 - 2 **criar vetor** $F[0 \dots n]$ com -1's
 - 3 $F[0] \leftarrow 0$;
 - 4 $F[1] \leftarrow 1$;
 - 5 **retorne** Fib-Mem-Rec(n, F)
-

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema
Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Algoritmo 8: Fib-Mem(n)

Entrada: n : número inteiro ≥ 0

Saída: Fibonacci de n

- 1 **se** $n \leq 1$ **então retorne** n ;
 - 2 **criar vetor** $F[0 \dots n]$ com -1's
 - 3 $F[0] \leftarrow 0$;
 - 4 $F[1] \leftarrow 1$;
 - 5 **retorne** Fib-Mem-Rec(n, F)
-

Algoritmo 9: Fib-Mem-Rec(n, F)

- 1 **se** $F[n] \geq 0$ **então**
 - 2 **retorne** $F[n]$
 - 3 $F[n] \leftarrow \text{Fib-Mem-Rec}(n - 1) + \text{Fib-Mem-Rec}(n - 2)$
 - 4 **retorne** $F[n]$
-

Sequência de Fibonacci - Memoização

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema
Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Algoritmo 10: Fib-Mem(n)

Entrada: n : número inteiro ≥ 0

Saída: Fibonacci de n

- 1 **se** $n \leq 1$ **então retorne** n ;
 - 2 **criar vetor** $F[0 \dots n]$ com -1's
 - 3 $F[0] \leftarrow 0$;
 - 4 $F[1] \leftarrow 1$;
 - 5 **retorne** Fib-Mem-Rec(n, F)
-

Algoritmo 11: Fib-Mem-Rec(n, F)

- 1 **se** $F[n] \geq 0$ **então**
 - 2 **retorne** $F[n]$
 - 3 $F[n] \leftarrow \text{Fib-Mem-Rec}(n - 1) + \text{Fib-Mem-Rec}(n - 2)$
 - 4 **retorne** $F[n]$
-

Tempo $O(n)$

Sequência de Fibonacci - PD

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Sequência de Fibonacci - PD

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Algoritmo 13: Fibonacci(n)

Entrada: n : número inteiro ≥ 0

Saída: Fibonacci de n

Sequência de Fibonacci - PD

EDA - Aula 14

Prof. Eurinardo

Aulas Passadas

Programação
Dinâmica

Sequência de
Fibonacci -
Problema

Problema

Recursão

Algoritmo Recursivo

Algoritmo de Memoização

Algoritmo de Programação
Dinâmica

Algoritmo 14: Fibonacci(n)

Entrada: n : número inteiro ≥ 0

Saída: Fibonacci de n

```
1 se  $n \leq 1$  então
2   |   retorne  $n$ 
```

Algoritmo 15: Fibonacci(n)

Entrada: n : número inteiro ≥ 0 **Saída:** Fibonacci de n

```
1 se  $n \leq 1$  então  
2   | retorne  $n$   
3 criar vetor  $F[0 \dots n]$   
4  $F[0] \leftarrow 0$ ;  
5  $F[1] \leftarrow 1$ ;
```

Algoritmo 16: Fibonacci(n)

Entrada: n : número inteiro ≥ 0 **Saída:** Fibonacci de n

```
1 se  $n \leq 1$  então
2   retorne  $n$ 
3 criar vetor  $F[0 \dots n]$ 
4  $F[0] \leftarrow 0$ ;
5  $F[1] \leftarrow 1$ ;
6 para  $i \leftarrow 2$  até  $n$  faça
  |
```

Algoritmo 17: Fibonacci(n)

Entrada: n : número inteiro ≥ 0 **Saída:** Fibonacci de n

```
1 se  $n \leq 1$  então
2   |   retorne  $n$ 
3 criar vetor  $F[0 \dots n]$ 
4  $F[0] \leftarrow 0$ ;
5  $F[1] \leftarrow 1$ ;
6 para  $i \leftarrow 2$  até  $n$  faça
7   |    $F[i] \leftarrow F[i-1] + F[i-2]$ 
```

Algoritmo 18: Fibonacci(n)

Entrada: n : número inteiro ≥ 0 **Saída:** Fibonacci de n

```
1 se  $n \leq 1$  então
2   | retorne  $n$ 
3 criar vetor  $F[0 \dots n]$ 
4  $F[0] \leftarrow 0$ ;
5  $F[1] \leftarrow 1$ ;
6 para  $i \leftarrow 2$  até  $n$  faça
7   |  $F[i] \leftarrow F[i - 1] + F[i - 2]$ 
8 retorne  $F[n]$ 
```

Algoritmo 19: Fibonacci(n)

Entrada: n : número inteiro ≥ 0 **Saída:** Fibonacci de n

```
1 se  $n \leq 1$  então
2   |   retorne  $n$ 
3 criar vetor  $F[0 \dots n]$ 
4  $F[0] \leftarrow 0$ ;
5  $F[1] \leftarrow 1$ ;
6 para  $i \leftarrow 2$  até  $n$  faça
7   |    $F[i] \leftarrow F[i-1] + F[i-2]$ 
8 retorne  $F[n]$ 
```

Tempo $O(n)$



LEISERSON, C.E., STEIN, C., RIVEST, R.L.,
CORMEN T.H.

Algoritmos: teoria e prática, 3ed.

Editora Campus, ano 2012.

Obrigado!