Análise de Algoritmos
Aula 13

Prof. Murilo V. G. da Silva

DINF/UFPR

(material da disciplina: André Guedes, Renato Carmo, Murilo da Silva)

Utilizarameos PD para resolver três problemas neste curso

Utilizarameos PD para resolver três problemas neste curso

- Calcular o n-ésimo número da Sequência de Fibonacci

Programação Dinâmica

Utilizarameos PD para resolver três problemas neste curso

- Calcular o n-ésimo número da Sequência de Fibonacci
- Computar a Subsequência Crescente Máxima em um vetor

Utilizarameos PD para resolver três problemas neste curso

- Calcular o n-ésimo número da Sequência de Fibonacci
- Computar a Subsequência Crescente Máxima em um vetor
- Computar Distância de Edição / Alinhamento de Sequência

Utilizarameos PD para resolver três problemas neste curso

- Calcular o n-ésimo número da Sequência de Fibonacci
- Computar a Subsequência Crescente Máxima em um vetor
- Computar Distância de Edição / Alinhamento de Sequência
- Algoritmo $\mathcal{O}^*(2^n)$ para o Caixeiro viajante

Utilizarameos PD para resolver três problemas neste curso

- Calcular o n-ésimo número da Sequência de Fibonacci
- Computar a Subsequência Crescente Máxima em um vetor
- Computar Distância de Edição / Alinhamento de Sequência
- Algoritmo $\mathcal{O}^*(2^n)$ para o Caixeiro viajante
- Uso de PD em outros algoritmos de outros cursos:
  - Todos os caminhos mínimos de um grafo
  - Problema da mochila

A *Sequência de Fibonacci* é dada por

Sequência de Fibonacci

A *Sequência de Fibonacci* é dada por

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

A *Sequência de Fibonacci* é dada por

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

O algoritmo trivial para o cálculo de $F(n)$ é

$\mathrm{Fib}_R(n)$

Se $n \leq 1$
    Devolva n
Devolva $Fib_R(n-2) + Fib_R(n-1)$

Sequência de Fibonacci

A *Sequência de Fibonacci* é dada por

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

O algoritmo trivial para o cálculo de $F(n)$ é

$\text{Fib}_R(n)$

Se $n \leq 1$
    Devolva n
Devolva $Fib_R(n-2) + Fib_R(n-1)$

- Que problema resolve?

Sequência de Fibonacci

A *Sequência de Fibonacci* é dada por

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

O algoritmo trivial para o cálculo de $F(n)$ é

$\text{Fib}_R(n)$

Se $n \leq 1$
    Devolva n
Devolva $Fib_R(n-2) + Fib_R(n-1)$

- Que problema resolve?
- Está correto?

Sequência de Fibonacci

$\text{Fib}_R(n)$
Se $n \leq 1$ Devolva n Devolva $Fib_R(n-2) + Fib_R(n-1)$

Sequência de Fibonacci

$\text{Fib}_R(n)$

Se $n \leq 1$
    Devolva n
Devolva $Fib_R(n-2) + Fib_R(n-1)$

- Quanto Custa?

$\mathrm{Fib}_R(n)$

Se $n \leq 1$
    Devolva n
Devolva $Fib_R(n-2) + Fib_R(n-1)$

- Quanto Custa?

Teorema

O tempo de execução de $\mathrm{Fib}_\mathrm{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Sequência de Fibonacci

$\mathrm{Fib}_R(n)$

Se $n \leq 1$
    Devolva n
Devolva $Fib_R(n-2) + Fib_R(n-1)$

- Quanto Custa?

Teorema

O tempo de execução de $\mathrm{Fib}_\mathrm{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Prova: Próximo slide.

Sequência de Fibonacci

$\text{Fib}_R(n)$

Se $n \leq 1$
    Devolva n
Devolva $Fib_R(n-2) + Fib_R(n-1)$

- Quanto Custa?

Teorema

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Prova: Próximo slide.

Corolario

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta(F(n)))$.

$\text{Fib}_R(n)$

Se $n \leq 1$
  Devolva n
Devolva $Fib_R(n-2) + Fib_R(n-1)$

- Quanto Custa?

Teorema

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Prova: Próximo slide.

Corolario

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta(F(n)))$.

Corolário

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Omega(1.61^n)$.

Sequência de Fibonacci

Teorema

O tempo de execução de $\mathrm{Fib_R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Sequência de Fibonacci

Teorema

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Prova

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta(S(n))$, onde

Sequência de Fibonacci

Teorema

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Prova

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta(S(n))$, onde

$$S(n) = \text{número de somas na execução de } \text{Fib}_\text{R}(n).$$

Sequência de Fibonacci

Teorema

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Prova

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta(S(n))$, onde

$$S(n) = \text{número de somas na execução de } \text{Fib}_\text{R}(n).$$

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

Sequência de Fibonacci

Teorema

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Prova

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta(S(n))$, onde

$$S(n) = \text{número de somas na execução de } \text{Fib}_\text{R}(n).$$

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

cuja solução é,

Teorema

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$.

Prova

O tempo de execução de $\text{Fib}_\text{R}(n)$ é $\Theta(S(n))$, onde

$$S(n) = \text{número de somas na execução de } \text{Fib}_\text{R}(n).$$

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

cuja solução é,

$$\begin{aligned} S(n) &= \frac{5-3\sqrt{5}}{10}\left(\frac{1-\sqrt{5}}{2}\right)^n + \frac{5+3\sqrt{5}}{10}\left(\frac{1+\sqrt{5}}{2}\right)^n - 1 \\ &= \Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right). \end{aligned}$$

É possível fazer melhor?

É possível fazer melhor?

Na execução de $\text{Fib}_{\text{R}}(n)$:

É possível fazer melhor?

Na execução de $\text{Fib}_\text{R}(n)$:

- $\text{Fib}_\text{R}(k)$ é executado múltiplas vezes, para cada $k < n$ (Exercício 54).

É possível fazer melhor?

Na execução de $\text{Fib}_\text{R}(n)$:

- $\text{Fib}_\text{R}(k)$ é executado múltiplas vezes, para cada $k < n$ (Exercício 54).
- Podemos evitar isso da seguinte maneira:

É possível fazer melhor?

Na execução de $\text{Fib}_\text{R}(n)$:

- $\text{Fib}_\text{R}(k)$ é executado múltiplas vezes, para cada $k < n$ (Exercício 54).
- Podemos evitar isso da seguinte maneira:

$\text{Fib}_D(n)$

$v \leftarrow$ vetor indexado por $[0..n]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k] \leftarrow v[k-2] + v[k-1]$
Devolva $F[n]$

Sequência de Fibonacci

$\text{Fib}_D(n)$

$v \leftarrow$ vetor indexado por $[0..n]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k] \leftarrow v[k-2] + v[k-1]$
Devolva $F[n]$

Sequência de Fibonacci

$\text{Fib}_D(n)$

$v \leftarrow$ vetor indexado por $[0..n]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k] \leftarrow v[k-2] + v[k-1]$
Devolva $F[n]$

- Está correto?

Sequência de Fibonacci

$\text{Fib}_D(n)$

$v \leftarrow$ vetor indexado por $[0..n]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k] \leftarrow v[k-2] + v[k-1]$
Devolva $F[n]$

- Está correto?

Teorema

Ao início da execução do laço de $\text{Fib}_\text{D}(n)$ temos $v[i] = F(i)$, para todo $i \in [0..k-1]$.

$\text{Fib}_D(n)$

$v \leftarrow$ vetor indexado por $[0..n]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k] \leftarrow v[k-2] + v[k-1]$
Devolva $F[n]$

- Está correto?

Teorema

Ao início da execução do laço de $\text{Fib}_\text{D}(n)$ temos $v[i] = F(i)$, para todo $i \in [0..k-1]$.

- Quanto custa?

$\mathrm{Fib}_D(n)$

$v \leftarrow$ vetor indexado por $[0..n]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k] \leftarrow v[k-2] + v[k-1]$
Devolva $F[n]$

- Está correto?

Teorema

Ao início da execução do laço de $\mathrm{Fib}_\mathrm{D}(n)$ temos $v[i] = F(i)$, para todo $i \in [0..k-1]$.

- Quanto custa?

Teorema

A execução de $\mathrm{Fib}_\mathrm{D}(n)$ toma tempo $\Theta(n)$ e espaço $\Theta(n)$.

$\text{Fib}_D(n)$

$v \leftarrow$ vetor indexado por $[0..n]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k] \leftarrow v[k-2] + v[k-1]$
Devolva $F[n]$

- Está correto?

Teorema

Ao início da execução do laço de $\text{Fib}_\text{D}(n)$ temos $v[i] = F(i)$, para todo $i \in [0..k-1]$.

- Quanto custa?

Teorema

A execução de $\text{Fib}_\text{D}(n)$ toma tempo $\Theta(n)$ e espaço $\Theta(n)$.

Corolário

A execução de $\text{Fib}_\text{D}(n)$ toma tempo $\Theta(\log F(n)))$ e espaço $\Theta(\log F(n)))$.

Sequência de Fibonacci

$\text{Fib}_D(n)$

$v \leftarrow$ vetor indexado por $[0..n]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
  $v[k] \leftarrow v[k-2] + v[k-1]$
Devolva $F[n]$

- Está correto?

Teorema

Ao início da execução do laço de $\text{Fib}_\text{D}(n)$ temos $v[i] = F(i)$, para todo $i \in [0..k-1]$.

- Quanto custa?

Teorema

A execução de $\text{Fib}_\text{D}(n)$ toma tempo $\Theta(n)$ e espaço $\Theta(n)$.

Corolário

A execução de $\text{Fib}_\text{D}(n)$ toma tempo $\Theta(\log F(n)))$ e espaço $\Theta(\log F(n)))$.

Provas: Exercícios.

Sequência de Fibonacci

- É possível fazer melhor?

Teorema

É possível computar $F(n)$ em tempo $\Theta(n)$ e espaço $\Theta(1)$.

Sequência de Fibonacci

- É possível fazer melhor?

Teorema

É possível computar $F(n)$ em tempo $\Theta(n)$ e espaço $\Theta(1)$.

Prova:

Sequência de Fibonacci

- É possível fazer melhor?

Teorema

É possível computar $F(n)$ em tempo $\Theta(n)$ e espaço $\Theta(1)$.

Prova:

$\text{Fib}_{D1}(n)$

$v \leftarrow$ vetor indexado por $[0..1]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k \bmod 2] \leftarrow v[k-2] + v[k-1]$
Devolva $v[n \bmod 2]$

- Está correto? (Exercício)

Sequência de Fibonacci

- É possível fazer melhor?

Teorema

É possível computar $F(n)$ em tempo $\Theta(n)$ e espaço $\Theta(1)$.

Prova:

$\text{Fib}_{D1}(n)$

$v \leftarrow$ vetor indexado por $[0..1]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k \bmod 2] \leftarrow v[k-2] + v[k-1]$
Devolva $v[n \bmod 2]$

- Está correto? (Exercício)

Sequência de Fibonacci

$\text{Fib}_{D1}(n)$

$v \leftarrow$ vetor indexado por $[0..1]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k \bmod 2] \leftarrow v[k-2] + v[k-1]$
Devolva $v[n \bmod 2]$

- Quanto custa?

Sequência de Fibonacci

$\text{Fib}_{D1}(n)$

$v \leftarrow$ vetor indexado por $[0..1]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k \bmod 2] \leftarrow v[k-2] + v[k-1]$
Devolva $v[n \bmod 2]$

- Quanto custa?

Teorema

A execução de $\text{Fib}_{\text{D1}}(n)$ toma tempo $\Theta(n)$ e espaço $\Theta(1)$.

Sequência de Fibonacci

$\text{Fib}_{D1}(n)$

$v \leftarrow$ vetor indexado por $[0..1]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k \bmod 2] \leftarrow v[k-2] + v[k-1]$
Devolva $v[n \bmod 2]$

- Quanto custa?

Teorema

A execução de $\text{Fib}_{\text{D1}}(n)$ toma tempo $\Theta(n)$ e espaço $\Theta(1)$.

Prova: Exercício.

Sequência de Fibonacci

$\text{Fib}_{D1}(n)$

$v \leftarrow$ vetor indexado por [0..1]
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k \bmod 2] \leftarrow v[k-2] + v[k-1]$
Devolva $v[n \bmod 2]$

- Quanto custa?

Teorema

A execução de $\text{Fib}_{\text{D1}}(n)$ toma tempo $\Theta(n)$ e espaço $\Theta(1)$.

Prova: Exercício.

- É possível fazer melhor?

Sequência de Fibonacci

$\text{Fib}_{D1}(n)$

$v \leftarrow$ vetor indexado por $[0..1]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k \bmod 2] \leftarrow v[k-2] + v[k-1]$
Devolva $v[n \bmod 2]$

- Quanto custa?

Teorema

A execução de $\text{Fib}_{\text{D1}}(n)$ toma tempo $\Theta(n)$ e espaço $\Theta(1)$.

Prova: Exercício.

- É possível fazer melhor?

Teorema

É possível computar $F(n)$ em tempo $\Theta(\log n)$ e espaço $\Theta(1)$.

Sequência de Fibonacci

$\text{Fib}_{D1}(n)$

$v \leftarrow$ vetor indexado por $[0..1]$
$v[0] \leftarrow 0$
$v[1] \leftarrow 1$
Para k *de* 2 *até* n
    $v[k \bmod 2] \leftarrow v[k-2] + v[k-1]$
Devolva $v[n \bmod 2]$

- Quanto custa?

Teorema

A execução de $\text{Fib}_{\text{D1}}(n)$ toma tempo $\Theta(n)$ e espaço $\Theta(1)$.

Prova: Exercício.

- É possível fazer melhor?

Teorema

É possível computar $F(n)$ em tempo $\Theta(\log n)$ e espaço $\Theta(1)$.

Prova: Exercício.

Subsequência Crescente Máxima

Subsequência Crescente Máxima (SCM)

- Entrada: Um vetor X de números inteiros.

Subsequência Crescente Máxima

Subsequência Crescente Máxima (SCM)

- Entrada: Um vetor X de números inteiros.
- Saída: Um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Subsequência Crescente Máxima

Subsequência Crescente Máxima (SCM)

- Entrada: Um vetor X de números inteiros.
- Saída: Um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9

Subsequência Crescente Máxima

Subsequência Crescente Máxima (SCM)

- Entrada: Um vetor X de números inteiros.
- Saída: Um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Subsequência Crescente Máxima (SCM)

- Entrada: Um vetor X de números inteiros.
- Saída: Um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Uma resposta da instância acima é o vetor S abaixo.

Subsequência Crescente Máxima

Subsequência Crescente Máxima (SCM)

- Entrada: Um vetor X de números inteiros.
- Saída: Um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Uma resposta da instância acima é o vetor S abaixo.

2	5	6	7

Subsequência Crescente Máxima

Subsequência Crescente Máxima (SCM)

- Entrada: Um vetor X de números inteiros.
- Saída: Um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Uma resposta da instância acima é o vetor S abaixo.

2	5	6	7

Observe: O conteúdo da resposta (i.e., o conteúdo do vetor S, que é 2, 5, 6, 7),

Subsequência Crescente Máxima

Subsequência Crescente Máxima (SCM)

- Entrada: Um vetor X de números inteiros.
- Saída: Um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Uma resposta da instância acima é o vetor S abaixo.

2	5	6	7

Observe: O conteúdo da resposta (i.e., o conteúdo do vetor S, que é 2, 5, 6, 7), indexa a maior sequência crescente de valores da instância de entrada X, ou seja, $2, 3, 6, 9$.

Subsequência Crescente Máxima

Exemplo do slide anterior:

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Solução do exemplo.

2	5	6	7

Subsequência Crescente Máxima

Exemplo do slide anterior:

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Solução do exemplo.

2	5	6	7

- Pergunta: como encontrar a resposta?

Subsequência Crescente Máxima

Exemplo do slide anterior:

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Solução do exemplo.

2	5	6	7

- Pergunta: como encontrar a resposta?
- Atentar para o teorema abaixo:

Subsequência Crescente Máxima

Exemplo do slide anterior:

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Solução do exemplo.

2	5	6	7

- Pergunta: como encontrar a resposta?
- Atentar para o teorema abaixo:

Teorema

Um algoritmo que examina cada uma das subsequências possíveis e escolhe a maior toma tempo $\Omega(n2^n)$ onde $n = |X|$

Subsequência Crescente Máxima

Exemplo do slide anterior:

1	2	3	4	5	6	7
5	2	8	6	3	6	9

Solução do exemplo.

2	5	6	7

- Pergunta: como encontrar a resposta?
- Atentar para o teorema abaixo:

Teorema

Um algoritmo que examina cada uma das subsequências possíveis e escolhe a maior toma tempo $\Omega(n2^n)$ onde $n = |X|$

Prova: Exercício.

Subsequência Crescente Máxima

Exemplo do slide anterior:

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Solução do exemplo.

2	5	6	7

Subsequência Crescente Máxima

Exemplo do slide anterior:

5	2	8	6	3	6	9

1 2 3 4 5 6 7

Solução do exemplo.

2	5	6	7

Subsequência Crescente Máxima

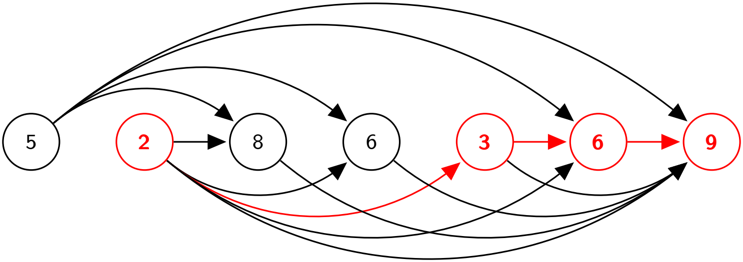
Exemplo do slide anterior:

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Solução do exemplo.

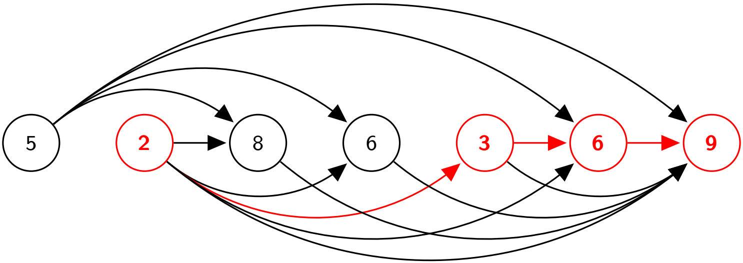
2	5	6	7

Observe que a solução é um caminho máximo neste DAG:



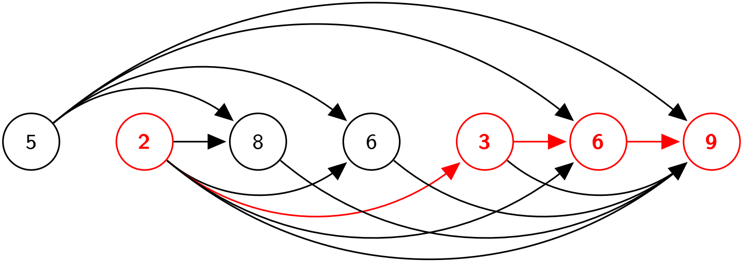
Subsequência Crescente Máxima

A solução do nosso problema é um caminho máximo em um DAG:



Subsequência Crescente Máxima

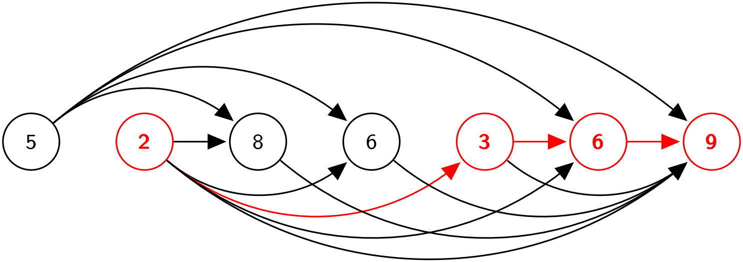
A solução do nosso problema é um caminho máximo em um DAG:



Vamos começar com algo mais simples:

Subsequência Crescente Máxima

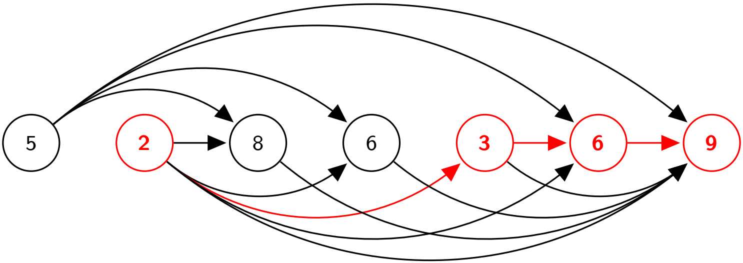
A solução do nosso problema é um caminho máximo em um DAG:



Vamos começar com algo mais simples: o **tamanho** do caminho máximo

Subsequência Crescente Máxima

A solução do nosso problema é um caminho máximo em um DAG:



Vamos começar com algo mais simples: o **tamanho** do caminho máximo

- Neste caso: a solução é 4

Tamanho da Subsequência Crescente Máxima

Tamanho Subsequência Crescente Máxima (TSCM)

- Entrada: Um vetor X de números inteiros.

Tamanho da Subsequência Crescente Máxima

Tamanho Subsequência Crescente Máxima (TSCM)

- Entrada: Um vetor X de números inteiros.
- Saída: O tamanho de um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Tamanho da Subsequência Crescente Máxima

Tamanho Subsequência Crescente Máxima (TSCM)

- Entrada: Um vetor X de números inteiros.
- Saída: O tamanho de um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9

Tamanho da Subsequência Crescente Máxima

Tamanho Subsequência Crescente Máxima (TSCM)

- Entrada: Um vetor X de números inteiros.
- Saída: O tamanho de um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Tamanho da Subsequência Crescente Máxima

Tamanho Subsequência Crescente Máxima (TSCM)

- Entrada: Um vetor X de números inteiros.
- Saída: O tamanho de um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

1	2	3	4	5	6	7
5	2	8	6	3	6	9

Uma resposta da instância acima é **4**.

Tamanho da Subsequência Crescente Máxima

Tamanho Subsequência Crescente Máxima (TSCM)

- Entrada: Um vetor X de números inteiros.
- Saída: O tamanho de um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

1	2	3	4	5	6	7
5	2	8	6	3	6	9

Uma resposta da instância acima é **4**.

Pois o vetor S de tamanho 4 abaixo satisfaz a maximalidade do problema.

Tamanho da Subsequência Crescente Máxima

Tamanho Subsequência Crescente Máxima (TSCM)

- Entrada: Um vetor X de números inteiros.
- Saída: O tamanho de um vetor S de índices de X de tamanho máximo satisfazendo

$$\begin{aligned} S[i-1]] &< S[i], \\ X[S[i-1]] &< X[S[i]], \text{ para todo } 1 < i \leq |S|. \end{aligned}$$

Exemplo: Considere a seguinte instância (um vetor) X do problema.

5	2	8	6	3	6	9
1	2	3	4	5	6	7

Uma resposta da instância acima é **4**.

Pois o vetor S de tamanho 4 abaixo satisfaz a maximalidade do problema.

2	5	6	7

Subsequência Crescente Máxima

Simplificando ainda mais:

Tamanho Subsequência Crescente Máxima com Final Definido (TSCMFD)

- Entrada: Um par (X, k), onde

$$\begin{array}{ll} X & \text{é um vetor de inteiros indexado por } [1..n] \\ k & \in [1..n]. \end{array}$$

Subsequência Crescente Máxima

Simplificando ainda mais:

Tamanho Subsequência Crescente Máxima com Final Definido (TSCMFD)

- Entrada: Um par (X, k), onde

$$\begin{array}{ll} X & \text{é um vetor de inteiros indexado por } [1..n] \\ k & \in [1..n]. \end{array}$$

- Saída: O tamanho de um vetor S de índices de X de tamanho máximo t.q.

$$\begin{array}{rcl} S[i-1]] & < & S[i], \\ X[S[i-1]] & < & X[S[i]], \text{ para todo } 1 < i \leq |S|, \\ S[|S|] & = & k. \end{array}$$

Subsequência Crescente Máxima

Simplificando ainda mais:

Tamanho Subsequência Crescente Máxima com Final Definido (TSCMFD)

- Entrada: Um par (X, k), onde

$$\begin{array}{ll} X & \text{é um vetor de inteiros indexado por } [1..n] \\ k & \in [1..n]. \end{array}$$

- Saída: O tamanho de um vetor S de índices de X de tamanho máximo t.q.

$$\begin{array}{rcl} S[i-1]] & < & S[i], \\ X[S[i-1]] & < & X[S[i]], \text{ para todo } 1 < i \leq |S|, \\ S[|S|] & = & k. \end{array}$$

Considere o vetor X abaixo:

1	2	3	4	5	6	7	8	9
5	2	8	6	3	6	9	3	3

Subsequência Crescente Máxima

Simplificando ainda mais:

Tamanho Subsequência Crescente Máxima com Final Definido (TSCMFD)

- Entrada: Um par (X, k), onde

$$\begin{array}{ll} X & \text{é um vetor de inteiros indexado por } [1..n] \\ k & \in [1..n]. \end{array}$$

- Saída: O tamanho de um vetor S de índices de X de tamanho máximo t.q.

$$\begin{array}{rcl} S[i-1]] & < & S[i], \\ X[S[i-1]] & < & X[S[i]], \text{ para todo } 1 < i \leq |S|, \\ S[|S|] & = & k. \end{array}$$

Considere o vetor X abaixo:

1	2	3	4	5	6	7	8	9
5	2	8	6	3	6	9	3	3

Respostas para instâncias:

Subsequência Crescente Máxima

Simplificando ainda mais:

Tamanho Subsequência Crescente Máxima com Final Definido (TSCMFD)

- Entrada: Um par (X, k), onde

$$\begin{array}{ll} X & \text{é um vetor de inteiros indexado por } [1..n] \\ k & \in [1..n]. \end{array}$$

- Saída: O tamanho de um vetor S de índices de X de tamanho máximo t.q.

$$\begin{array}{rcl} S[i-1]] & < & S[i], \\ X[S[i-1]] & < & X[S[i]], \text{ para todo } 1 < i \leq |S|, \\ S[|S|] & = & k. \end{array}$$

Considere o vetor X abaixo:

5	2	8	6	3	6	9	3	3
1	2	3	4	5	6	7	8	9

Respostas para instâncias:

- Resposta para $(X, 7)$:

Subsequência Crescente Máxima

Simplificando ainda mais:

Tamanho Subsequência Crescente Máxima com Final Definido (TSCMFD)

- Entrada: Um par (X, k), onde

$$\begin{array}{ll} X & \text{é um vetor de inteiros indexado por } [1..n] \\ k & \in [1..n]. \end{array}$$

- Saída: O tamanho de um vetor S de índices de X de tamanho máximo t.q.

$$\begin{array}{rcl} S[i-1]] & < & S[i], \\ X[S[i-1]] & < & X[S[i]], \text{ para todo } 1 < i \leq |S|, \\ S[|S|] & = & k. \end{array}$$

Considere o vetor X abaixo:

1	2	3	4	5	6	7	8	9
5	2	8	6	3	6	9	3	3

Respostas para instâncias:

- Resposta para $(X, 7)$: 4

Subsequência Crescente Máxima

Simplificando ainda mais:

Tamanho Subsequência Crescente Máxima com Final Definido (TSCMFD)

- Entrada: Um par (X, k), onde

$$\begin{array}{ll} X & \text{é um vetor de inteiros indexado por } [1..n] \\ k & \in [1..n]. \end{array}$$

- Saída: O tamanho de um vetor S de índices de X de tamanho máximo t.q.

$$\begin{array}{rcl} S[i-1]] & < & S[i], \\ X[S[i-1]] & < & X[S[i]], \text{ para todo } 1 < i \leq |S|, \\ S[|S|] & = & k. \end{array}$$

Considere o vetor X abaixo:

1	2	3	4	5	6	7	8	9
5	2	8	6	3	6	9	3	3

Respostas para instâncias:

- Resposta para $(X, 7)$: 4
- Resposta para $(X, 9)$:

Subsequência Crescente Máxima

Simplificando ainda mais:

Tamanho Subsequência Crescente Máxima com Final Definido (TSCMFD)

- Entrada: Um par (X, k), onde

$$\begin{array}{ll} X & \text{é um vetor de inteiros indexado por } [1..n] \\ k & \in [1..n]. \end{array}$$

- Saída: O tamanho de um vetor S de índices de X de tamanho máximo t.q.

$$\begin{array}{rcl} S[i-1]] & < & S[i], \\ X[S[i-1]] & < & X[S[i]], \text{ para todo } 1 < i \leq |S|, \\ S[|S|] & = & k. \end{array}$$

Considere o vetor X abaixo:

5	2	8	6	3	6	9	3	3
1	2	3	4	5	6	7	8	9

Respostas para instâncias:

- Resposta para $(X, 7)$: 4
- Resposta para $(X, 9)$: 3

Tamanho da Subsequência Crescente Máxima

Fazendo

$$\begin{aligned} r(X)\text{:} &\quad \text{resposta da instância } X \text{ de } \mathrm{TSCM} \text{ e} \\ r(X,k)\text{:} &\quad \text{resposta da instância } (X,k) \text{ de } \mathrm{TSCMFD}, \end{aligned}$$

Tamanho da Subsequência Crescente Máxima

Fazendo

$r(X)$: resposta da instância X de TSCM e

$r(X,k)$: resposta da instância (X,k) de TSCMFD,

$$r(X) = \max\{r(X,k) \colon 1 \leq k \leq |X|\}$$

Fazendo

$r(X)$: resposta da instância X de TSCM e

$r(X, k)$: resposta da instância (X, k) de TSCMFD,

$$r(X) = \max\{r(X, k) \colon 1 \leq k \leq |X|\}$$

$$r(X, k) = \max\{1\} \cup \{r(X, i) + 1 \mid X[i] < X[k] \colon 1 \leq i < k\}.$$

Tamanho da Subsequência Crescente Máxima

$$r(X) = \max\{r(X,k) \colon 1 \le k \le |X|\}$$

Tamanho da Subsequência Crescente Máxima

$$r(X) = \max\{r(X,k) \colon 1 \le k \le |X|\}$$

$$r(X,k) = \max\{1\} \cup \{r(X,i)+1 \mid X[i] < X[k] \colon 1 \le i < k\}.$$

Tamanho da Subsequência Crescente Máxima

$$r(X) = \max\{r(X,k) \colon 1 \leq k \leq |X|\}$$

$$r(X,k) = \max\{1\} \cup \{r(X,i) + 1 \mid X[i] < X[k] \colon 1 \leq i < k\}.$$

TSCM(X)

```
tmax ← 1
Para k de 1 até |X|
    r ← TSCM_R(X, k)
    Se tmax < r
        tmax ← r
Devolva tmax
```

Tamanho da Subsequência Crescente Máxima

$$r(X) = \max\{r(X,k)\colon 1 \le k \le |X|\}$$

$$r(X,k) = \max\{1\} \cup \{r(X,i) + 1 \mid X[i] < X[k]\colon 1 \le i < k\}.$$

TSCM(X)

tmax $\leftarrow 1$
Para k *de* 1 *até* $|X|$
    $r \leftarrow \mathrm{TSCM}_R(X, k)$
    Se $tmax < r$
        tmax $\leftarrow r$
Devolva $tmax$

onde

Tamanho da Subsequência Crescente Máxima

$$r(X) = \max\{r(X,k) \colon 1 \le k \le |X|\}$$

$$r(X,k) = \max\{1\} \cup \{r(X,i) + 1 \mid X[i] < X[k] \colon 1 \le i < k\}.$$

TSCM(X)

---

tmax $\leftarrow 1$
Para k *de* 1 *até* $|X|$
    $r \leftarrow \mathrm{TSCM}_R(X, k)$
    Se $tmax < r$
        tmax $\leftarrow r$
Devolva $tmax$

---

onde

TSCM$_R$(X, k)

---

tmax $\leftarrow 1$
Para i *de* 1 *até* $k - 1$
    $r \leftarrow \mathrm{TSCM}_R(X, i)$
    Se $X[i] < X[k]$
        $r \leftarrow r + 1$
    Se $tmax < r$
        tmax $\leftarrow r$
Devolva $tmax$

Tamanho da Subsequência Crescente Máxima

- Está correto?

Tamanho da Subsequência Crescente Máxima

- Está correto? Sim, é a transcrição das recorrências

Tamanho da Subsequência Crescente Máxima

- Está correto? Sim, é a transcrição das recorrências
- Quanto custa?

Tamanho da Subsequência Crescente Máxima

- Está correto? Sim, é a transcrição das recorrências
- Quanto custa? $\Theta(n^3)$ Exercício.

Tamanho da Subsequência Crescente Máxima

- Está correto? Sim, é a transcrição das recorrências
- Quanto custa? $\Theta(n^3)$ Exercício.
- Da para fazer melhor?

Tamanho da Subsequência Crescente Máxima

- Está correto? Sim, é a transcrição das recorrências
- Quanto custa? $\Theta(n^3)$ Exercício.
- Da para fazer melhor? , sim, usando PD.

Tamanho da Subsequência Crescente Máxima

Usando PD (primeira tentativa):

$\mathrm{TSCM}_D(X, k)$

Tam $\leftarrow$ vetor indexado por $[1..k]$
Para $i \leftarrow 1$ *até* k
    $\mathrm{Tam}[i] \leftarrow \mathrm{Preenche}(\mathrm{Tam}, i, X)$
Devolva $Tam[k]$

Usando PD (primeira tentativa):

$\text{TSCM}_D(X, k)$

Tam $\leftarrow$ vetor indexado por $[1..k]$
Para $i \leftarrow 1$ *até* k
    $\text{Tam}[i] \leftarrow \text{Preenche}(\text{Tam}, i, X)$
Devolva $Tam[k]$

---

onde

Tamanho da Subsequência Crescente Máxima

Usando PD (primeira tentativa):

$\text{TSCM}_D(X, k)$

Tam $\leftarrow$ vetor indexado por $[1..k]$
Para $i \leftarrow 1$ *até* k
    $\text{Tam}[i] \leftarrow \text{Preenche}(\text{Tam}, i, X)$
Devolva $Tam[k]$

---

onde

$\text{Preenche}(\text{Tam}, i, X)$

---

tmax $\leftarrow 1$
Para j *de* 1 *até* $i - 1$
    Se $X[j] < X[i]$ *e* $tmax < Tam[j] + 1$
        $\text{tmax} \leftarrow \text{Tam}[j] + 1$
Devolva *tmax*

Está correto?

Está correto?

Lema 82

Se ao início da execução de $\text{Preenche}(\text{Tam}, i, X)$,

$$\text{Tam}[j] = r(X, j) \text{ para todo } j \in [1..i-1],$$

então

$$\text{Preenche}(\text{Tam}, i, X) = r(X, i).$$

Tamanho da Subsequência Crescente Máxima

Está correto?

Lema 82

Se ao início da execução de $\text{Preenche}(\text{Tam}, i, X)$,

$$\text{Tam}[j] = r(X, j) \text{ para todo } j \in [1..i-1],$$

então

$$\text{Preenche}(\text{Tam}, i, X) = r(X, i).$$

Quanto custa?

Tamanho da Subsequência Crescente Máxima

Está correto?

Lema 82

Se ao início da execução de Preenche(Tam, i, X),

$$\text{Tam}[j] = r(X, j) \text{ para todo } j \in [1..i-1],$$

então

$$\text{Preenche}(\text{Tam}, i, X) = r(X, i).$$

Quanto custa?

Lema 83

O tempo de execução de Preenche(Tam, i, X) é $\Theta(i)$.

Tamanho da Subsequência Crescente Máxima

Está correto?

Lema 82

Se ao início da execução de Preenche(Tam, i, X),

$$\mathrm{Tam}[j] = r(X, j) \text{ para todo } j \in [1..i-1],$$

então

$$\mathrm{Preenche}(\mathrm{Tam}, i, X) = r(X, i).$$

Quanto custa?

Lema 83

O tempo de execução de Preenche(Tam, i, X) é $\Theta(i)$.

Prova

Da análise direta do algoritmo temos

$$T_{\mathrm{Preenche}}(\mathrm{Tam}, i, X) = \Theta(1) + (i-1)\Theta(1) = \Theta(i).$$

Tamanho da Subsequência Crescente Máxima

Corolario

O tempo de execução de $\text{TSCM}_D(X, k)$ é $\Theta(k^2)$.

Tamanho da Subsequência Crescente Máxima

Corolario

O tempo de execução de $\text{TSCM}_D(X, k)$ é $\Theta(k^2)$.

Prova

Da análise direta do algoritmo temos

$$T_{\text{TSCM}_D}(X, k) = \Theta(1) + \sum_{i=1}^{k} \Theta(i) = \Theta(1) + \Theta(k^2) = \Theta(k^2).$$

Tamanho da Subsequência Crescente Máxima

Corolario

O tempo de execução de $\text{TSCM}_D(X, k)$ é $\Theta(k^2)$.

Prova

Da análise direta do algoritmo temos

$$T_{\text{TSCM}_D}(X, k) = \Theta(1) + \sum_{i=1}^{k} \Theta(i) = \Theta(1) + \Theta(k^2) = \Theta(k^2).$$

corolario

O tempo de execução de $\text{TSCM}(X)$ é $\Theta(|X|^3)$.

Tamanho da Subsequência Crescente Máxima

Corolario

O tempo de execução de $\text{TSCM}_D(X, k)$ é $\Theta(k^2)$.

Prova

Da análise direta do algoritmo temos

$$T_{\text{TSCM}_D}(X, k) = \Theta(1) + \sum_{i=1}^{k} \Theta(i) = \Theta(1) + \Theta(k^2) = \Theta(k^2).$$

corolario

O tempo de execução de $\text{TSCM}(X)$ é $\Theta(|X|^3)$.

Prova

Da análise direta do algoritmo temos

$$T_{\text{TSCM}}(X) = \Theta(1) + \sum_{k=1}^{|X|} \Theta(k^2) = \Theta(1) + \Theta(|X|^3) = \Theta(|X|^3).$$

Tamanho da Subsequência Crescente Máxima

Agora de fato fazendo melhor:

TSCM'(X)

Tam $\leftarrow$ vetor indexado por $[1..|X|]$
tmax $\leftarrow 1$
Para i *de* 1 *até* $|X|$
    Tam[i] $\leftarrow$ Preenche(X, i, Tam)
    Se $Tam[tmax] < Tam[i]$
        tmax $\leftarrow i$
Devolva $Tam[tmax]$

Tamanho da Subsequência Crescente Máxima

Agora de fato fazendo melhor:

TSCM'(X)

Tam $\leftarrow$ vetor indexado por $[1..|X|]$
tmax $\leftarrow 1$
Para i *de* 1 *até* $|X|$
    Tam[i] $\leftarrow$ Preenche(X, i, Tam)
    Se $Tam[tmax] < Tam[i]$
        tmax $\leftarrow i$
Devolva $Tam[tmax]$

- Está correto?

Agora de fato fazendo melhor:

TSCM'(X)

Tam $\leftarrow$ vetor indexado por $[1..|X|]$
tmax $\leftarrow 1$
Para i *de* 1 *até* $|X|$
    Tam[i] $\leftarrow$ Preenche(X, i, Tam)
    Se $Tam[tmax] < Tam[i]$
        tmax $\leftarrow i$
Devolva $Tam[tmax]$

- Está correto? Tanto quanto TSCM.

Tamanho da Subsequência Crescente Máxima

Agora de fato fazendo melhor:

TSCM'(X)

Tam $\leftarrow$ vetor indexado por $[1..|X|]$
tmax $\leftarrow 1$
Para *i de* 1 *até* $|X|$
    Tam[i] $\leftarrow$ Preenche(X, i, Tam)
    Se $Tam[tmax] < Tam[i]$
        tmax $\leftarrow i$
Devolva $Tam[tmax]$

- Está correto? Tanto quanto TSCM.
- Quanto custa?

Tamanho da Subsequência Crescente Máxima

Agora de fato fazendo melhor:

TSCM'(X)

Tam $\leftarrow$ vetor indexado por $[1..|X|]$
tmax $\leftarrow 1$
Para *i de* 1 *até* $|X|$
    Tam[i] $\leftarrow$ Preenche(X, i, Tam)
    Se $Tam[tmax] < Tam[i]$
        tmax $\leftarrow i$
Devolva $Tam[tmax]$

- Está correto? Tanto quanto TSCM.
- Quanto custa?

corolario

O tempo de execução de TSCM$'(X)$ é $\Theta(|X|^2)$.

Agora de fato fazendo melhor:

TSCM'(X)

Tam $\leftarrow$ vetor indexado por $[1..|X|]$
tmax $\leftarrow 1$
Para *i de* 1 *até* $|X|$
    Tam[i] $\leftarrow$ Preenche(X, i, Tam)
    Se $Tam[tmax] < Tam[i]$
        tmax $\leftarrow i$
Devolva $Tam[tmax]$

- Está correto? Tanto quanto TSCM.
- Quanto custa?

corolario

O tempo de execução de $\mathrm{TSCM}'(X)$ é $\Theta(|X|^2)$.

Prova

Da análise direta do algoritmo temos

$$\begin{aligned} T_{\mathrm{TSCM}'}(X) &= \Theta(1) + \sum_{i=1}^{|X|}(\Theta(i) + \Theta(1)) \\ &= \Theta(1) + \Theta(|X|^2) + \Theta(|X|) \\ &= \Theta(|X|^2). \end{aligned}$$

Subsequência Crescente Máxima

E o problema original?

E o problema original?

Subsequência Crescente Máxima

E o problema original?

- Seja Ant$[i]$ é o índice do elemento anterior a $X[i]$ numa subsequência crescente máxima de X finalizada em i.

Subsequência Crescente Máxima

E o problema original?

- Seja Ant[i] é o índice do elemento anterior a $X[i]$ numa subsequência crescente máxima de X finalizada em i.

SCM(X)

---

indmax $\leftarrow 1$
Para i *de* 1 *até* $|X|$
    Preenche$(\text{Tam}, \text{Ant}, i, X)$
    Se $Tam[indmax] < Tam[i]$
        indmax $\leftarrow i$
Devolva $Reconstroi(Ant, indmax, Tam[indmax])$

Subsequência Crescente Máxima

E o problema original?

- Seja Ant[i] é o índice do elemento anterior a $X[i]$ numa subsequência crescente máxima de X finalizada em i.

SCM(X)

---

indmax $\leftarrow 1$
Para i *de* 1 *até* $|X|$
    Preenche(Tam, Ant, i, X)
    Se $Tam[indmax] < Tam[i]$
        indmax $\leftarrow i$
Devolva $Reconstroi(Ant, indmax, Tam[indmax])$

Preenche(Tam, Ant, i, X)

Tam[i] $\leftarrow 1$
Ant[i] $\leftarrow i$
Para j *de* 1 *até* $i - 1$
    Se $X[j] < X[i]$ *e* $Tam[i] < Tam[j] + 1$
        Tam[i] $\leftarrow$ Tam[j] $+ 1$
        Ant[i] $\leftarrow j$

Tamanho da Subsequência Crescente Máxima

Reconstroi(Ant, indmax, tmax)

$S \leftarrow$ vetor indexado por [1..tmax]
$S[\text{tmax}] = \text{indmax}$
Para i *de* $tmax - 1$ *até* 1
    $S[i] \leftarrow \text{Ant}[S[i+1]]$
Devolva S