MC458 — Projeto e Análise de Algoritmos I

C.C. de Souza   C.N. da Silva   O. Lee

Antes de mais nada...

- Uma versão anterior deste conjunto de slides foi preparada por Cid Carvalho de Souza e Cândida Nunes da Silva para uma instância anterior desta disciplina.
- O que vocês tem em mãos é uma versão modificada preparada para atender a meus gostos.
- Nunca é demais enfatizar que o material é apenas um guia e não deve ser usado como única fonte de estudo. Para isso consultem a bibliografia (em especial o CLR ou CLRS).

Orlando Lee

Agradecimentos (Cid e Cândida)

- Várias pessoas contribuíram direta ou indiretamente com a preparação deste material.
- Algumas destas pessoas cederam gentilmente seus arquivos digitais enquanto outras cederam gentilmente o seu tempo fazendo correções e dando sugestões.
- Uma lista destes "colaboradores" (em ordem alfabética) é dada abaixo:
  - Célia Picinin de Mello
  - José Coelho de Pina
  - Orlando Lee
  - Paulo Feofiloff
  - Pedro Rezende
  - Ricardo Dahab
  - Zanoni Dias

Indução matemática no projeto de algoritmos

Projeto de algoritmos por indução

- A seguir, usaremos a técnica de prova por indução para desenvolver algoritmos para vários problemas.

Projeto de algoritmos por indução

- A seguir, usaremos a técnica de prova por indução para desenvolver algoritmos para vários problemas.
- Isto é, a formulação do algoritmo é similar ao desenvolvimento de uma prova por indução.

Projeto de algoritmos por indução

- A seguir, usaremos a técnica de prova por indução para desenvolver algoritmos para vários problemas.
- Isto é, a formulação do algoritmo é similar ao desenvolvimento de uma prova por indução.
- Assim, para resolver o problema P projetamos um algoritmo em dois passos:

Projeto de algoritmos por indução

- A seguir, usaremos a técnica de prova por indução para desenvolver algoritmos para vários problemas.
- Isto é, a formulação do algoritmo é similar ao desenvolvimento de uma prova por indução.
- Assim, para resolver o problema P projetamos um algoritmo em dois passos:
  1. **Caso base:** exibimos a resolução de uma ou mais instâncias pequenas de P;

Projeto de algoritmos por indução

- A seguir, usaremos a técnica de prova por indução para desenvolver algoritmos para vários problemas.
- Isto é, a formulação do algoritmo é similar ao desenvolvimento de uma prova por indução.
- Assim, para resolver o problema P projetamos um algoritmo em dois passos:
  1. **Caso base:** exibimos a resolução de uma ou mais instâncias pequenas de P;
  2. **Passo de indução:** mostramos como a solução de toda instância de P pode ser obtida a partir da solução de uma ou mais instâncias menores de P.

Projeto de algoritmos por indução

Este processo indutivo resulta em algoritmos recursivos em que:

Projeto de algoritmos por indução

Este processo indutivo resulta em algoritmos recursivos em que:

- a base da indução corresponde à resolução do caso base da recursão,

Projeto de algoritmos por indução

Este processo indutivo resulta em algoritmos recursivos em que:

- a base da indução corresponde à resolução do caso base da recursão,
- a aplicação da hipótese de indução corresponde a uma ou mais chamadas recursivas, e

Projeto de algoritmos por indução

Este processo indutivo resulta em algoritmos recursivos em que:

- a base da indução corresponde à resolução do caso base da recursão,
- a aplicação da hipótese de indução corresponde a uma ou mais chamadas recursivas, e
- o passo da indução corresponde ao processo de obtenção da resposta para o caso geral a partir daquelas devolvidas pelas chamadas recursivas.

Projeto de algoritmos por indução

- Um benefício imediato é que o uso (correto) da técnica nos fornece uma prova da corretude do algoritmo.

Projeto de algoritmos por indução

- Um benefício imediato é que o uso (correto) da técnica nos fornece uma prova da corretude do algoritmo.
- A complexidade do algoritmo resultante é expressa numa recorrência.

Projeto de algoritmos por indução

- Um benefício imediato é que o uso (correto) da técnica nos fornece uma prova da corretude do algoritmo.
- A complexidade do algoritmo resultante é expressa numa recorrência.
- Freqüentemente o algoritmo é eficiente, embora existam exemplos simples em que isso não acontece.

Projeto de algoritmos por indução

- Um benefício imediato é que o uso (correto) da técnica nos fornece uma prova da corretude do algoritmo.
- A complexidade do algoritmo resultante é expressa numa recorrência.
- Freqüentemente o algoritmo é eficiente, embora existam exemplos simples em que isso não acontece.
- Iniciaremos com dois exemplos que usam indução:

Projeto de algoritmos por indução

- Um benefício imediato é que o uso (correto) da técnica nos fornece uma prova da corretude do algoritmo.
- A complexidade do algoritmo resultante é expressa numa recorrência.
- Freqüentemente o algoritmo é eficiente, embora existam exemplos simples em que isso não acontece.
- Iniciaremos com dois exemplos que usam indução:
  1. cálculo do valor de polinômios e

Projeto de algoritmos por indução

- Um benefício imediato é que o uso (correto) da técnica nos fornece uma prova da corretude do algoritmo.
- A complexidade do algoritmo resultante é expressa numa recorrência.
- Freqüentemente o algoritmo é eficiente, embora existam exemplos simples em que isso não acontece.
- Iniciaremos com dois exemplos que usam indução:
  1. cálculo do valor de polinômios e
  2. obtenção de subgrafos maximais com restrições de grau.

Exemplo 1 - Cálculo de polinômios

Problema:

Dada uma seqüência de números reais $a_n, a_{n-1}, \ldots, a_1, a_0$*, e um número real* x*, calcular o valor do polinômio*
$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ *no ponto* x*.*

Exemplo 1 - Cálculo de polinômios

Problema:

Dada uma seqüência de números reais $a_n, a_{n-1}, \ldots, a_1, a_0$, *e um número real* x, *calcular o valor do polinômio*
$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ *no ponto* x.

Naturalmente este é um problema bem simples.
Estamos interessados em projetar um algoritmo que faça o menor número de operações aritméticas (multiplicações, principalmente).

Exemplo 1 - Cálculo de polinômios

Problema:

Dados uma seqüência de números reais $a_n, a_{n-1}, \ldots, a_1, a_0$, *e um número real* x, *calcular o valor de* $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$.

Hipótese de indução : (primeira tentativa)

Dados uma seqüência de números reais $a_{n-1}, \ldots, a_1, a_0$, e um número real x, sabemos calcular o valor de $P_{n-1}(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$.

Exemplo 1 - Cálculo de polinômios

Problema:

Dados uma seqüência de números reais $a_n, a_{n-1}, \ldots, a_1, a_0$*, e um número real* x*, calcular o valor de* $P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$.

Hipótese de indução : (primeira tentativa)

Dados uma seqüência de números reais $a_{n-1}, \ldots, a_1, a_0$, e um número real x, sabemos calcular o valor de $P_{n-1}(x) = a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$.

- **Caso base**: $n = 0$. A solução é a_0.

Exemplo 1 - Cálculo de polinômios

Problema:

Dados uma seqüência de números reais $a_n, a_{n-1}, \ldots, a_1, a_0$, *e um número real* x, *calcular o valor de* $P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$.

Hipótese de indução : (primeira tentativa)

Dados uma seqüência de números reais $a_{n-1}, \ldots, a_1, a_0$, e um número real x, sabemos calcular o valor de $P_{n-1}(x) = a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$.

- **Caso base**: $n = 0$. A solução é a_0.
- Note que $P_n(x) = a_n x^n + P_{n-1}(x)$. Assim, para calcular $P_n(x)$, basta calcular x^n, multiplicar o resultado por a_n e somar o resultado com $P_{n-1}(x)$.

Exemplo 1 - Solução 1 - Algoritmo

$P_n(x) = a_n x^n + P_{n-1}(x)$

$\text{CÁLCULOPOLINÔMIO}(A, x)$
▷ **Entrada:** Coeficientes $A = a_n, a_{n-1}, \ldots, a_1, a_0$ e real x.
▷ **Saída:** O valor de $P_n(x)$.
1. **se** $n = 0$
2. **então** $P \leftarrow a_0$
3. **senão**
4. $A' \leftarrow a_{n-1}, \ldots, a_1, a_0$
5. $P' \leftarrow \text{CÁLCULOPOLINÔMIO}(A', x)$
6. $xn \leftarrow x$
7. **para** $i \leftarrow 1$ **até** $n - 1$ **faça** $xn \leftarrow xn * x$
8. $P \leftarrow a_n * xn + P'$
9. **devolva** P

Exemplo 1 - Solução 1 - Complexidade

Chamando de $T(n)$ o número de operações aritméticas realizadas pelo algoritmo, temos a seguinte recorrência para $T(n)$:

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + n \text{ multiplicações} + 1 \text{ adição}, & n > 0. \end{cases}$$

Exemplo 1 - Solução 1 - Complexidade

Chamando de $T(n)$ o número de operações aritméticas realizadas pelo algoritmo, temos a seguinte recorrência para $T(n)$:

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + n \text{ multiplicações} + 1 \text{ adição}, & n > 0. \end{cases}$$

Não é difícil ver que

$$\begin{aligned} T(n) &= \sum_{i=1}^{n} [i \text{ multiplicações} + 1 \text{ adição}] \\ &= (n+1)n/2 \text{ multiplicações} + n \text{ adições}. \end{aligned}$$

Segunda solução indutiva

Segunda solução indutiva

- Desperdício cometido na primeira solução: recálculo de potências de x.

Segunda solução indutiva

- Desperdício cometido na primeira solução: recálculo de potências de x.
- Alternativa: eliminar essa computação desnecessária trazendo o cálculo de x^{n-1} para dentro da hipótese de indução.

Segunda solução indutiva

- Desperdício cometido na primeira solução: recálculo de potências de x.
- Alternativa: eliminar essa computação desnecessária trazendo o cálculo de x^{n-1} para dentro da hipótese de indução.

Hipótese de indução reforçada:

Sabemos calcular o valor de $P_{n-1}(x) = a_{n-1}x^{n-1} + \cdots + a_1x + a_0$ e também o valor de x^{n-1}.

Segunda solução indutiva

- Desperdício cometido na primeira solução: recálculo de potências de x.
- Alternativa: eliminar essa computação desnecessária trazendo o cálculo de x^{n-1} para dentro da hipótese de indução.

Hipótese de indução reforçada:

Sabemos calcular o valor de $P_{n-1}(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ e também o valor de x^{n-1}.

- No caso base $n = 0$, a solução agora é $(a_0, 1)$.

Segunda solução indutiva

- Desperdício cometido na primeira solução: recálculo de potências de x.
- Alternativa: eliminar essa computação desnecessária trazendo o cálculo de x^{n-1} para dentro da hipótese de indução.

Hipótese de indução reforçada:

Sabemos calcular o valor de $P_{n-1}(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ e também o valor de x^{n-1}.

- No caso base $n = 0$, a solução agora é $(a_0, 1)$.
- No passo de indução, primeiro calculamos x^n multiplicando x por x^{n-1}, conforme exigido na hipótese. Em seguida, calculamos $P_n(x)$ multiplicando x^n por a_n e somando o valor obtido com $P_{n-1}(x)$.

Exemplo 1 - Solução 2 - Algoritmo

$P_n(x) = a_n x^n + P_{n-1}(x)$

CálculoPolinômio(A, x)

▷ **Entrada:** Coeficientes $A = a_n, a_{n-1}, \ldots, a_1, a_0$ e real x.

▷ **Saída:** O valor de $P_n(x)$ e o valor de x^n.

1. **se** $n = 0$
2.   **então** $P \leftarrow a_0;\ xn \leftarrow 1$
3.   **senão**
4. $A' \leftarrow a_{n-1}, \ldots, a_1, a_0$
5. $P', x' \leftarrow$ CálculoPolinômio(A', x)
6. $xn \leftarrow x * x'$
7. $P \leftarrow a_n * xn + P'$
8. **devolva** P, xn

Exemplo 1 - Solução 2 - Complexidade

Novamente, se $T(n)$ é o número de operações aritméticas realizadas pelo algoritmo, $T(n)$ é dado por:

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + 2 \text{ multiplicações} + 1 \text{ adição}, & n > 0. \end{cases}$$

Exemplo 1 - Solução 2 - Complexidade

Novamente, se $T(n)$ é o número de operações aritméticas realizadas pelo algoritmo, $T(n)$ é dado por:

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + 2 \text{ multiplicações} + 1 \text{ adição}, & n > 0. \end{cases}$$

A solução da recorrência é

$$\begin{aligned} T(n) &= \sum_{i=1}^{n} (2 \text{ multiplicações} + 1 \text{ adição}) \\ &= 2n \text{ multiplicações} + n \text{ adições}. \end{aligned}$$

Terceira solução indutiva

Terceira solução indutiva

- A escolha de considerar o polinômio $P_{n-1}(x)$ na hipótese de indução não é a única possível.

Terceira solução indutiva

- A escolha de considerar o polinômio $P_{n-1}(x)$ na hipótese de indução não é a única possível.
- Podemos **reforçar** ainda mais a HI e ter um ganho de complexidade:

Terceira solução indutiva

- A escolha de considerar o polinômio $P_{n-1}(x)$ na hipótese de indução não é a única possível.
- Podemos **reforçar** ainda mais a HI e ter um ganho de complexidade:

Hipótese de indução mais reforçada:

Sabemos calcular o valor $P'_{n-1}(x) = a_n x^{n-1} + a_{n-1} x^{n-2} \cdots + a_1$.

Terceira solução indutiva

- A escolha de considerar o polinômio $P_{n-1}(x)$ na hipótese de indução não é a única possível.
- Podemos **reforçar** ainda mais a HI e ter um ganho de complexidade:

Hipótese de indução mais reforçada:

Sabemos calcular o valor $P'_{n-1}(x) = a_n x^{n-1} + a_{n-1} x^{n-2} \cdots + a_1$.

- Note que $P_n(x) = xP'_{n-1}(x) + a_0$. Assim, bastam uma multiplicação e uma adição para obtermos $P_n(x)$ a partir de $P'_{n-1}(x)$.

Terceira solução indutiva

- A escolha de considerar o polinômio $P_{n-1}(x)$ na hipótese de indução não é a única possível.
- Podemos **reforçar** ainda mais a HI e ter um ganho de complexidade:

Hipótese de indução mais reforçada:

Sabemos calcular o valor $P'_{n-1}(x) = a_n x^{n-1} + a_{n-1}x^{n-2} \cdots + a_1$.

- Note que $P_n(x) = xP'_{n-1}(x) + a_0$. Assim, bastam uma multiplicação e uma adição para obtermos $P_n(x)$ a partir de $P'_{n-1}(x)$.
- O caso base é trivial pois, para $n = 0$, a solução é a_0.

Exemplo 1 - Solução 3 - Algoritmo

$$P_n(x) = xP'_{n-1}(x) + a_0 \qquad P'_{n-1}(x) = a_n x^{n-1} + a_{n-1}x^{n-2} \cdots + a_1$$

$\textsc{CálculoPolinômio}(A, x)$
▷ **Entrada:** Coeficientes $A = a_n, a_{n-1}, \ldots, a_1, a_0$ e real x.
▷ **Saída:** O valor de $P_n(x)$.
se $n = 0$
  então $P \leftarrow a_0$
  senão
    $A' \leftarrow a_n, a_{n-1}, \ldots, a_1$
    $P' \leftarrow \textsc{CálculoPolinômio}(A', x)$
    $P \leftarrow x * P' + a_0$
devolva P

Exemplo 1 - Solução 3 - Complexidade

Temos que $T(n)$ é dado por

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + 1 \text{ multiplicação} + 1 \text{ adição}, & n > 0. \end{cases}$$

Exemplo 1 - Solução 3 - Complexidade

Temos que $T(n)$ é dado por

$$T(n) = \begin{cases} 0, & n = 0 \\ T(n-1) + 1 \text{ multiplicação} + 1 \text{ adição}, & n > 0. \end{cases}$$

A solução é

$$\begin{aligned} T(n) &= \sum_{i=1}^{n} (1 \text{ multiplicação} + 1 \text{ adição}) \\ &= n \text{ multiplicações} + n \text{ adições}. \end{aligned}$$

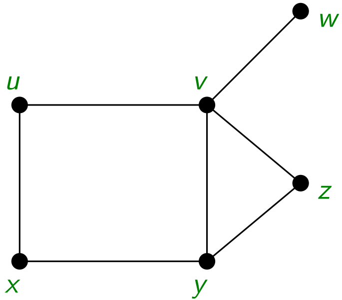
Exemplo 1 - Solução 3 - Complexidade

$$
\begin{aligned}
P_n(x) &= a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0 \\
&= (a_n x^{n-1} + a_{n-1}x^{n-2} + \cdots + a_2 x + a_1)x + a_0 \\
&= ((a_n x^{n-2} + a_{n-1}x^{n-3} + \cdots + a_2)x + a_1)x + a_0 \\
&= ((\cdots((a_n x + a_{n-1})x + a_{n-2})x \cdots + a_2)x + a_1)x + a_0.
\end{aligned}
$$

Esta forma de calcular $P_n(x)$ é chamada de **regra de Horner**.

Grafos – uma breve revisão/introdução(?)

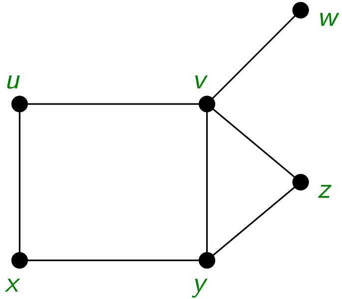
Um grafo é um par $G = (V, E)$ onde V é um conjunto finito de elementos chamados vértices e E é um conjunto de pares não-ordenados de vértices distintos chamados arestas.



Desenho do grafo $G = (V, E)$ onde $V = \{u, v, w, x, y, z\}$ e $E = \{uv, vw, ux, xy, yz, zv, vy\}$.

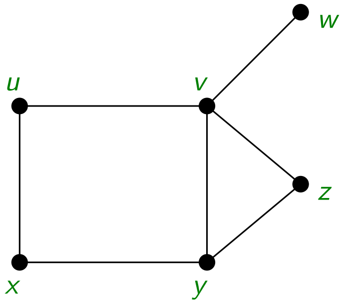
Grafos – uma breve revisão/introdução(?)

Se $e = uv$ é uma aresta de G, então dizemos que u e v são extremos de e e que e é incidente a u (e a v). Dois vértices u e v são adjacentes em G se uv é uma aresta de G.



Desenho do grafo $G = (V, E)$ onde $V = \{u, v, w, x, y, z\}$ e $E = \{uv, vw, ux, xy, yz, zv, vy\}$.

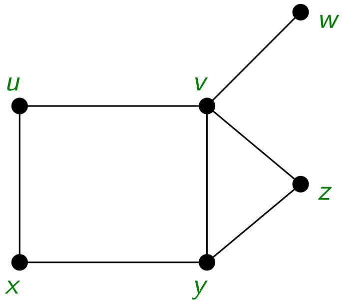
O grau de um vértice v em G, denotado por $d_G(v)$, é o número de arestas incidentes a v. Equivalentemente, é número de vértices que são adjacentes a v em G.



	u	v	w	x	y	z
$d_G(\ )$	?	?	?	?	?	?

Grau

O grau de um vértice v em G, denotado por $d_G(v)$, é o número de arestas incidentes a v. Equivalentemente, é número de vértices que são adjacentes a v em G.



	u	v	w	x	y	z
$d_G(\ )$	2	4	1	2	3	2

Teorema do Aperto de Mãos

Teorema do Aperto de Mãos

Teorema. Para todo grafo $G = (V, E)$ temos que

$$\sum d_G(v) = 2|E|.$$

Prova (Contagem dupla).

Teorema do Aperto de Mãos

Teorema. Para todo grafo $G = (V, E)$ temos que

$$\sum d_G(v) = 2|E|.$$

Prova (Contagem dupla). Considere uma aresta qualquer de G, digamos $e = uv$. Ela é contada duas vezes no lado direito da equação. Por outro lado, ela também é contada em $d_G(u)$ e em $d_G(v)$ no lado esquerdo. Portanto, a igualdade vale. ∎

Teorema do Aperto de Mãos

Teorema. Para todo grafo $G = (V, E)$ temos que

$$\sum d_G(v) = 2|E|.$$

Prova (Contagem dupla). Considere uma aresta qualquer de G, digamos $e = uv$. Ela é contada duas vezes no lado direito da equação. Por outro lado, ela também é contada em $d_G(u)$ e em $d_G(v)$ no lado esquerdo. Portanto, a igualdade vale. ∎

Este é o teorema mais importante do Universo!

Teorema do Aperto de Mãos

Teorema. Para todo grafo $G = (V, E)$ temos que

$$\sum d_G(v) = 2|E|.$$

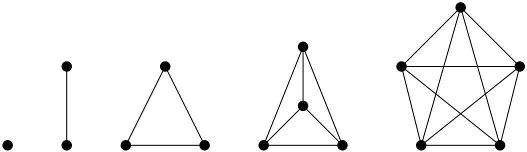
Prova (Contagem dupla). Considere uma aresta qualquer de G, digamos $e = uv$. Ela é contada duas vezes no lado direito da equação. Por outro lado, ela também é contada em $d_G(u)$ e em $d_G(v)$ no lado esquerdo. Portanto, a igualdade vale. ∎

Este é o teorema mais importante do Universo!

Ok, estou exagerando para chamar sua atenção...

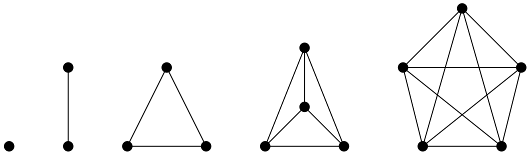
Grafos completos

Um grafo é completo se quaisquer dois vértices distintos forem adjacentes.



Grafos completos

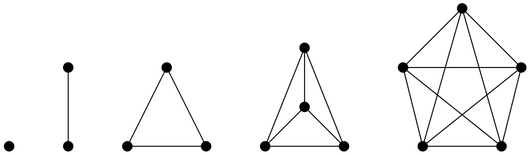
Um grafo é completo se quaisquer dois vértices distintos forem adjacentes.



Quantas arestas tem um grafo completo com n vértices?

Grafos completos

Um grafo é completo se quaisquer dois vértices distintos forem adjacentes.

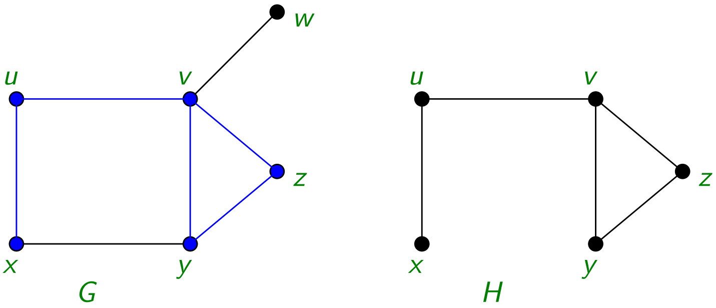


Quantas arestas tem um grafo completo com n vértices? $\binom{n}{2}$

Observação. $\binom{n}{k}$ é o número de subconjuntos de tamanho k de um conjunto de tamanho n. Cada aresta de um grafo pode ser vista como um subconjunto de tamanho dois do conjunto de vértices.

Subgrafos

Um grafo H é subgrafo de um grafo G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Usamos a notação $H \subseteq G$. Se $H \neq G$ então dizemos que H é um subgrafo próprio de G e denotamos $H \subset G$.



Subgrafos induzidos

Um subgrafo H de G é um subgrafo induzido de G se toda aresta de G com extremos em $V(H)$ também é uma aresta de $E(H)$.

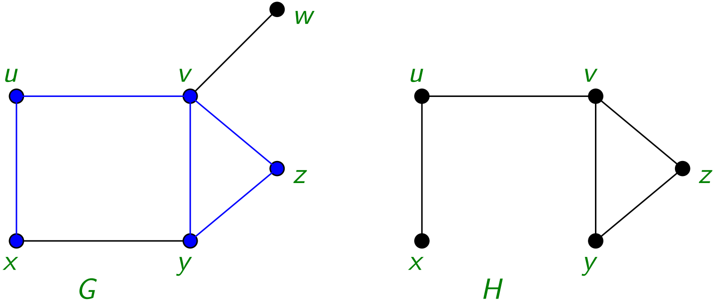


Figura: H não é um subgrafo induzido de G.

Subgrafos induzidos

Um subgrafo H de G é um subgrafo induzido de G se toda aresta de G com extremos em $V(H)$ também é uma aresta de $E(H)$.

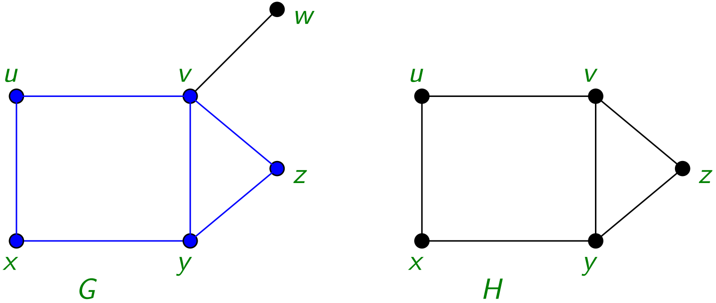


Figura: H é um subgrafo induzido de G.

Maximal versus máximo

Seja G um grafo e seja P uma propriedade sobre grafos.

Maximal versus máximo

Seja G um grafo e seja P uma propriedade sobre grafos.

Por exemplo, P poderia ser "ser conexo" ou "ter grau mínimo k".

Seja G um grafo e seja P uma propriedade sobre grafos.

Por exemplo, P poderia ser "ser conexo" ou "ter grau mínimo k".

Um subgrafo H de G é maximal em G com relação à propriedade P se

(i) H tem a propriedade P e
(ii) não existe subgrafo H' de G com a propriedade P tal que $H \subset H'$.

Maximal versus máximo

Seja G um grafo e seja P uma propriedade sobre grafos.

Por exemplo, P poderia ser "ser conexo" ou "ter grau mínimo k".

Um subgrafo H de G é maximal em G com relação à propriedade P se

1. H tem a propriedade P e
2. não existe subgrafo H' de G com a propriedade P tal que $H \subset H'$.

Um subgrafo H de G é máximo em G com relação à propriedade P se

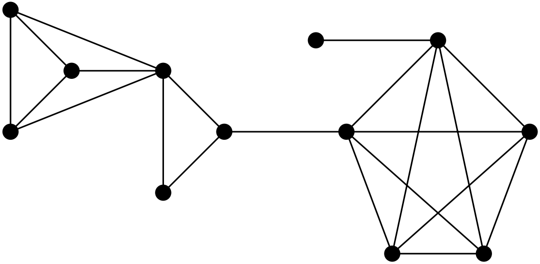
1. H tem a propriedade P e
2. $|V(H)|$ é máximo. (Esta definição pode variar dependendo do contexto.)

Maximal versus máximo

Vamos ilustrar isto com um exemplo.

Uma clique de um grafo G é um subgrafo completo de G.

Suponha que a propriedade P seja "H ser uma clique". Temos cliques maximais e cliques máximas.



Toda clique máxima é maximal, mas a recíproca não é verdade.

Exemplo 2
Subgrafos Maximais

- Suponha que você esteja planejando uma festa onde queira maximizar as interações entre as pessoas. Uma festa animada, mas sem recursos ilícitos para aumentar a animação.

Exemplo 2
Subgrafos Maximais

- Suponha que você esteja planejando uma festa onde queira maximizar as interações entre as pessoas. Uma festa animada, mas sem recursos ilícitos para aumentar a animação.
- Uma das maneiras de conseguir isso é fazer uma lista dos possíveis convidados e, para cada um desses, a lista dos convidados com quem ele(a) interagiria durante a festa.

Exemplo 2
Subgrafos Maximais

- Suponha que você esteja planejando uma festa onde queira maximizar as interações entre as pessoas. Uma festa animada, mas sem recursos ilícitos para aumentar a animação.
- Uma das maneiras de conseguir isso é fazer uma lista dos possíveis convidados e, para cada um desses, a lista dos convidados com quem ele(a) interagiria durante a festa.
- Em seguida, é só localizar o maior subgrupo de convidados que interagiriam com no mínimo k outros convidados dentro do subgrupo.

Exemplo 2 - Subgrafos Maximais

Formulação do problema usando grafos:

Exemplo 2 - Subgrafos Maximais

Formulação do problema usando grafos:

Dado um grafo G com n vértices e um inteiro $k > 0$, encontrar em G um subgrafo maximal induzido H tal que o grau de cada vértice de H seja pelo menos k. Note que podemos ter $H = (\emptyset, \emptyset)$.

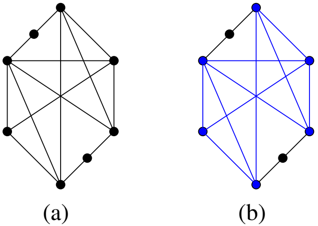


Figura: (a) para $k \in \{1, 2\}$ H é o próprio grafo G. (b) Subgrafo máximo com grau mínimo pelo menos $k = 3$. Para $k \geq 4$ a única solução é $H = (\emptyset, \emptyset)$.

Neste problema os conceitos de maximal e máximo coincidem. Ou seja, todo subgrafo maximal de grau mínimo pelo menos k também é um subgrafo máximo de grau mínimo pelo menos k. Isto ficará evidente assim que virmos a solução do problema.

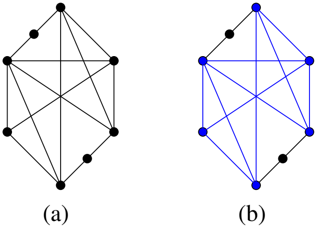


Figura: (a) para $k \in \{1, 2\}$ H é o próprio grafo G. (b) Subgrafo máximo com grau mínimo pelo menos $k = 3$. Para $k \geq 4$ a única solução é $H = (\emptyset, \emptyset)$.

Exemplo 2 -Solução por indução

Problema.

Dado um grafo G com n vértices e um inteiro $k > 0$, encontrar um subgrafo maximal induzido H de G tal que o grau de cada vértice de H seja pelo menos k. Note que podemos ter $H = (\emptyset, \emptyset)$.

Caso base: se $n \leq k$ então todo vértice de G tem grau $\leq k - 1$. Assim, a solução é $H = (\emptyset, \emptyset)$.

Exemplo 2 -Solução por indução

Problema.

Dado um grafo G com n vértices e um inteiro $k > 0$, encontrar um subgrafo maximal induzido H de G tal que o grau de cada vértice de H seja pelo menos k. Note que podemos ter $H = (\emptyset, \emptyset)$.

Caso base: se $n \leq k$ então todo vértice de G tem grau $\leq k - 1$. Assim, a solução é $H = (\emptyset, \emptyset)$.

Hipótese de indução. Sabemos como encontrar um subgrafo maximal H com grau mínimo pelo menos k de um grafo G' com menos que n vértices.

Exemplo 2 - Solução por indução

Exemplo 2 - Solução por indução

- **Passo de indução.** Seja G um grafo com $n > k + 1$ vértices.

Exemplo 2 - Solução por indução

- **Passo de indução.** Seja G um grafo com $n > k + 1$ vértices.
- Se todos os vértices de G têm grau $\geq k$, então basta tomar $H = G$.

Exemplo 2 - Solução por indução

- **Passo de indução.** Seja G um grafo com $n > k + 1$ vértices.
- Se todos os vértices de G têm grau $\geq k$, então basta tomar $H = G$.
- Caso contrário, seja v um vértice com grau $< k$.
  É fácil ver que nenhum subgrafo H no qual todo vértice tem grau $\geq k$ pode conter v.

Exemplo 2 - Solução por indução

- **Passo de indução.** Seja G um grafo com $n > k + 1$ vértices.
- Se todos os vértices de G têm grau $\geq k$, então basta tomar $H = G$.
- Caso contrário, seja v um vértice com grau $< k$.
  É fácil ver que nenhum subgrafo H no qual todo vértice tem grau $\geq k$ pode conter v.
  Assim, v pode ser removido e a HI aplicada ao grafo resultante $G' = G - v$.

Exemplo 2 - Solução por indução

- **Passo de indução.** Seja G um grafo com $n > k + 1$ vértices.
- Se todos os vértices de G têm grau $\geq k$, então basta tomar $H = G$.
- Caso contrário, seja v um vértice com grau $< k$.
  É fácil ver que nenhum subgrafo H no qual todo vértice tem grau $\geq k$ pode conter v.
  Assim, v pode ser removido e a HI aplicada ao grafo resultante $G' = G - v$.
- Seja H' o subgrafo devolvido pela aplicação da HI em G'. Então H' também é resposta para G.

Exemplo 2 - Algoritmo

SubgrafoMaximal(G, k)

$\triangleright$ **Entrada:** Grafo G com n vértices e um inteiro $k \geq 0$.

$\triangleright$ **Saída:** Subgrafo maximal induzido H de G com grau mínimo pelo menos k.

se $n < k + 1$
  então $H \leftarrow (\emptyset, \emptyset)$
  senão se todo vértice de G tem grau $\geq k$
    então $H \leftarrow G$
    senão
      seja v um vértice de G com grau $< k$
      $H \leftarrow$ SubgrafoMaximal($G - v, k$)
devolva H

Exemplo 2 - Algoritmo

SubgrafoMaximal(G, k)
$\triangleright$ **Entrada:** Grafo G com n vértices e um inteiro $k \geq 0$.
$\triangleright$ **Saída:** Subgrafo maximal induzido H de G com grau mínimo pelo menos k.

1. **se** $n < k + 1$
2. **então** $H \leftarrow (\emptyset, \emptyset)$
3. **senão se** todo vértice de G tem grau $\geq k$
4. **então** $H \leftarrow G$
5. **senão**
6. seja v um vértice de G com grau $< k$
7. $H \leftarrow$ SubgrafoMaximal$(G - v, k)$
8. **devolva** H

Não faremos análise de complexidade. Seria preciso introduzir representação de grafos que está fora do escopo do curso.

Exemplo 3
Fatores de balanceamento em árvores binárias

Árvores binárias balanceadas são estruturas de dados que minimizam o tempo de busca de informações nela armazenadas. A idéia é que, para todo nó v da árvore, o fator de balanceamento (f.b.) de v (diferença entre a altura da subárvore esquerda e a altura da subárvore direita de v) não desvie muito de zero.

Exemplo 3
Fatores de balanceamento em árvores binárias

Árvores binárias balanceadas são estruturas de dados que minimizam o tempo de busca de informações nela armazenadas. A idéia é que, para todo nó v da árvore, o fator de balanceamento (f.b.) de v (diferença entre a altura da subárvore esquerda e a altura da subárvore direita de v) não desvie muito de zero.

- Convenciona-se que a árvore vazia tem fator de balanceamento zero e altura igual a -1.

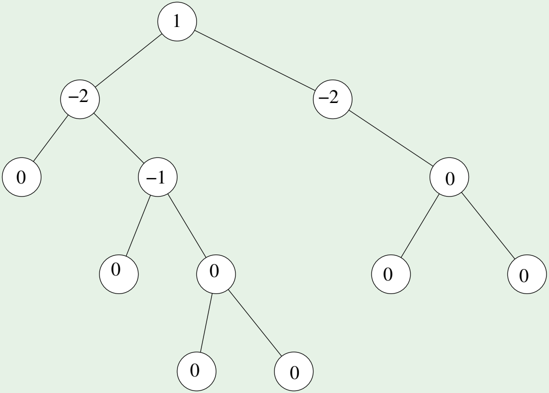
Exemplo 3
Fatores de balanceamento em árvores binárias

Árvores binárias balanceadas são estruturas de dados que minimizam o tempo de busca de informações nela armazenadas. A idéia é que, para todo nó v da árvore, o fator de balanceamento (f.b.) de v (diferença entre a altura da subárvore esquerda e a altura da subárvore direita de v) não desvie muito de zero.

- Convenciona-se que a árvore vazia tem fator de balanceamento zero e altura igual a -1.
- Árvores AVL são exemplos de árvores binárias balanceadas, em que o f.b. de cada nó é $-1, 0$ ou $+1$.

Exemplo 3 - Fatores de balanceamento

Exemplo de uma árvore binária e os fatores de balanceamento de seus nós.



Exemplo 3 - Indução

Problema:
Dada uma árvore binária A com n nós, calcular os fatores de balanceamento de cada nó de A.

Exemplo 3 - Indução

Problema:

Dada uma árvore binária A com n nós, calcular os fatores de balanceamento de cada nó de A.

- Vamos projetar o algoritmo indutivamente.

Exemplo 3 - Indução

Problema:
Dada uma árvore binária A com n nós, calcular os fatores de balanceamento de cada nó de A.

- Vamos projetar o algoritmo indutivamente.

Hipótese de indução:
Sabemos como calcular fatores de balanceamento de árvores binárias com menos que n nós.

Exemplo 3 - Indução

Problema:

Dada uma árvore binária A com n nós, calcular os fatores de balanceamento de cada nó de A.

- Vamos projetar o algoritmo indutivamente.

Hipótese de indução:

Sabemos como calcular fatores de balanceamento de árvores binárias com menos que n nós.

- **Caso base**: quando $n = 0$, convencionamos que o f.b. é igual a zero.

Exemplo 3 - Indução

Problema:

Dada uma árvore binária A com n nós, calcular os fatores de balanceamento de cada nó de A.

- Vamos projetar o algoritmo indutivamente.

Hipótese de indução:

Sabemos como calcular fatores de balanceamento de árvores binárias com menos que n nós.

- **Caso base**: quando $n = 0$, convencionamos que o f.b. é igual a zero.
- Vamos mostrar agora como usar a hipótese para calcular f.b.s de uma árvore A com exatamente n nós.

Exemplo 3 - Indução

Exemplo 3 - Indução

A idéia é aplicar a HI às subárvores esquerda e direita da raiz e, em seguida, calcular o f.b. da raiz.

Exemplo 3 - Indução

A idéia é aplicar a HI às subárvores esquerda e direita da raiz e, em seguida, calcular o f.b. da raiz.

Dificuldade:

Exemplo 3 - Indução

A idéia é aplicar a HI às subárvores esquerda e direita da raiz e, em seguida, calcular o f.b. da raiz.

Dificuldade: o f.b. da raiz depende das alturas das subárvores esquerda e direita e não dos seus f.b.s.

Exemplo 3 - Indução

A idéia é aplicar a HI às subárvores esquerda e direita da raiz e, em seguida, calcular o f.b. da raiz.

Dificuldade: o f.b. da raiz depende das alturas das subárvores esquerda e direita e não dos seus f.b.s.

Conclusão: é necessária uma HI mais forte!

Exemplo 3 - Indução

A idéia é aplicar a HI às subárvores esquerda e direita da raiz e, em seguida, calcular o f.b. da raiz.

Dificuldade: o f.b. da raiz depende das alturas das subárvores esquerda e direita e não dos seus f.b.s.

Conclusão: é necessária uma HI mais forte!

Nova hipótese de indução:

Para um $n > 0$ qualquer, sabemos como calcular fatores de balanceamento e alturas de árvores com menos que n nós.

Exemplo 3 - Indução

Exemplo 3 - Indução

- Novamente, o caso base $n = 0$ é fácil pois, por convenção, o f.b. é zero e a altura igual a -1.

Exemplo 3 - Indução

- Novamente, o caso base $n = 0$ é fácil pois, por convenção, o f.b. é zero e a altura igual a -1.
- Seja A uma árvore binária com n nós, para um $n > 0$ qualquer.

- Novamente, o caso base $n = 0$ é fácil pois, por convenção, o f.b. é zero e a altura igual a -1.
- Seja A uma árvore binária com n nós, para um $n > 0$ qualquer.

  Sejam (f_e, h_e) e (f_d, h_d) os f.b.s e alturas das subárvores esquerda (A_e) e direita (A_d) de A que, por HI, sabemos calcular.

Exemplo 3 - Indução

- Novamente, o caso base $n = 0$ é fácil pois, por convenção, o f.b. é zero e a altura igual a -1.
- Seja A uma árvore binária com n nós, para um $n > 0$ qualquer.

  Sejam (f_e, h_e) e (f_d, h_d) os f.b.s e alturas das subárvores esquerda (A_e) e direita (A_d) de A que, por HI, sabemos calcular.

  Então, o f.b. da raiz de A é $h_e - h_d$ e a altura da árvore é $\max(h_e, h_d) + 1$.

Exemplo 3 - Indução

- Novamente, o caso base $n = 0$ é fácil pois, por convenção, o f.b. é zero e a altura igual a -1.
- Seja A uma árvore binária com n nós, para um $n > 0$ qualquer.

  Sejam (f_e, h_e) e (f_d, h_d) os f.b.s e alturas das subárvores esquerda (A_e) e direita (A_d) de A que, por HI, sabemos calcular.

  Então, o f.b. da raiz de A é $h_e - h_d$ e a altura da árvore é $\max(h_e, h_d) + 1$.

  Isto completa o cálculo dos f.b.s e da altura de A.

Exemplo 3 - Indução

- Novamente, o caso base $n = 0$ é fácil pois, por convenção, o f.b. é zero e a altura igual a -1.
- Seja A uma árvore binária com n nós, para um $n > 0$ qualquer.

  Sejam (f_e, h_e) e (f_d, h_d) os f.b.s e alturas das subárvores esquerda (A_e) e direita (A_d) de A que, por HI, sabemos calcular.

  Então, o f.b. da raiz de A é $h_e - h_d$ e a altura da árvore é $\max(h_e, h_d) + 1$.

  Isto completa o cálculo dos f.b.s e da altura de A.

De novo, o fortalecimento da hipótese tornou a resolução do problema mais fácil.

Exemplo 3 - Algoritmo

FATORALTURA(A)
▷ **Entrada:** Uma árvore binária A com $n \geq 0$ nós
▷ **Saída:** A com os f.b.s nos seus nós e a altura de A

```
se n = 0
    então h ← −1
    senão
        seja r a raiz de A
        h_e ← FatorAltura(A_e)
        h_d ← FatorAltura(A_d)
        ▷ armazena o f.b. na raiz em r.f
        r.f ← h_e − h_d
        h ← max(h_e, h_d) + 1
devolva h
```

Seja $T(n)$ o número de operações executadas pelo algoritmo para calcular os f.b.s e a altura de uma árvore A de n nós. Então

$$T(n) = \begin{cases} \Theta(1), & n = 0 \\ T(n_e) + T(n_d) + \Theta(1), & n > 0, \end{cases}$$

Exemplo 3 - Complexidade

Seja $T(n)$ o número de operações executadas pelo algoritmo para calcular os f.b.s e a altura de uma árvore A de n nós. Então

$$T(n) = \begin{cases} \Theta(1), & n = 0 \\ T(n_e) + T(n_d) + \Theta(1), & n > 0, \end{cases}$$

onde n_e, n_d são os números de nós das subárvores esquerda e direita.

Exemplo 3 - Complexidade

O pior caso da recorrência parece ser quando, ao longo da recursão, um de n_e, n_d é sempre igual a zero (e o outro é sempre igual a $n-1$).

Exemplo 3 - Complexidade

O pior caso da recorrência parece ser quando, ao longo da recursão, um de n_e, n_d é sempre igual a zero (e o outro é sempre igual a $n-1$).

Chega-se então a:

$$T(n) = T(1) + \sum_{i=2}^{n} \Theta(1) = (n+1)\Theta(1) + n\Theta(1) \in \Theta(n).$$

O pior caso da recorrência parece ser quando, ao longo da recursão, um de n_e, n_d é sempre igual a zero (e o outro é sempre igual a $n-1$).

Chega-se então a:

$$T(n) = T(1) + \sum_{i=2}^{n} \Theta(1) = (n+1)\Theta(1) + n\Theta(1) \in \Theta(n).$$

Exercício. Há algum ganho de complexidade quando ambos n_e e n_d são aproximadamente $n/2$ ao longo da recursão?

Exemplo 4: o problema da celebridade

Em um conjunto S de n pessoas, uma celebridade é alguém que é conhecido por todas as pessoas de S mas que não conhece ninguém. (Celebridades são pessoas de difícil convívio...).

Exemplo 4: o problema da celebridade

Em um conjunto S de n pessoas, uma celebridade é alguém que é conhecido por todas as pessoas de S mas que não conhece ninguém. (Celebridades são pessoas de difícil convívio...).

Note que pode existir no máximo uma celebridade em S!

Exemplo 4: o problema da celebridade

Em um conjunto S de n pessoas, uma celebridade é alguém que é conhecido por todas as pessoas de S mas que não conhece ninguém. (Celebridades são pessoas de difícil convívio...).

Note que pode existir no máximo uma celebridade em S!

Problema:

Dado um conjunto S de n pessoas, determinar se existe uma celebridade em S.

Exemplo 4: o problema da celebridade

Formalização: para um conjunto S de n pessoas, associamos uma matriz M de dimensões $n \times n$ tal que $M[i,j] = 1$ se a pessoa i conhece a pessoa j e $M[i,j] = 0$ caso contrário. As entradas $M[i,i]$ não estão definidas.

$$\begin{pmatrix} * & 0 & 1 & 1 & 0 \\ 1 & * & 0 & 1 & 0 \\ 0 & 1 & * & 0 & 1 \\ 0 & 0 & 0 & * & 0 \\ 1 & 0 & 1 & 1 & * \end{pmatrix} \qquad \begin{pmatrix} * & 0 & 1 & 1 & 0 \\ 1 & * & 0 & 1 & 0 \\ 0 & 1 & * & 1 & 1 \\ 0 & 0 & 0 & * & 0 \\ 1 & 0 & 1 & 1 & * \end{pmatrix}$$

Exemplo 4: o problema da celebridade

Exemplo 4: o problema da celebridade

Problema:

Dado um conjunto de n pessoas e a matriz associada M encontrar (se existir) uma celebridade no conjunto.
Ou seja, determinar se existe um índice k tal que todos os elementos da coluna k são 1s e todos os elementos da linha k são 0s.

$$\begin{pmatrix} * & 0 & 1 & 1 & 0 \\ 1 & * & 0 & 1 & 0 \\ 0 & 1 & * & 0 & 1 \\ 0 & 0 & 0 & * & 0 \\ 1 & 0 & 1 & 1 & * \end{pmatrix} \qquad \begin{pmatrix} * & 0 & 1 & 1 & 0 \\ 1 & * & 0 & 1 & 0 \\ 0 & 1 & * & 1 & 1 \\ 0 & 0 & 0 & * & 0 \\ 1 & 0 & 1 & 1 & * \end{pmatrix}$$

Exemplo 4: o problema da celebridade

Problema:

Dado um conjunto de n pessoas e a matriz associada M encontrar (se existir) uma celebridade no conjunto.
Ou seja, determinar se existe um índice k tal que todos os elementos da coluna k são 1s e todos os elementos da linha k são 0s.

$$\begin{pmatrix} * & 0 & 1 & 1 & 0 \\ 1 & * & 0 & 1 & 0 \\ 0 & 1 & * & 0 & 1 \\ 0 & 0 & 0 & * & 0 \\ 1 & 0 & 1 & 1 & * \end{pmatrix} \qquad \begin{pmatrix} * & 0 & 1 & 1 & 0 \\ 1 & * & 0 & 1 & 0 \\ 0 & 1 & * & 1 & 1 \\ 0 & 0 & 0 & * & 0 \\ 1 & 0 & 1 & 1 & * \end{pmatrix}$$

Existe uma solução simples mas laboriosa: para cada pessoa i, verifique todos os elementos da linha i e da coluna i. O custo dessa solução é $2n(n-1)$.

Exemplo 4: o problema da celebridade

Exemplo 4: o problema da celebridade

Um argumento indutivo que parece ser mais eficiente é o seguinte:

Um argumento indutivo que parece ser mais eficiente é o seguinte:

Hipótese de Indução:

Sabemos encontrar uma celebridade (se existir) em um conjunto de $n-1$ *pessoas.*

Um argumento indutivo que parece ser mais eficiente é o seguinte:

Hipótese de Indução:

Sabemos encontrar uma celebridade (se existir) em um conjunto de $n-1$ pessoas.

- **Caso base:** se $n = 1$, podemos considerar que o único elemento é uma celebridade.

Exemplo 4: o problema da celebridade

Um argumento indutivo que parece ser mais eficiente é o seguinte:

Hipótese de Indução:

Sabemos encontrar uma celebridade (se existir) em um conjunto de $n - 1$ pessoas.

- **Caso base:** se $n = 1$, podemos considerar que o único elemento é uma celebridade.
- Outra opção é considerar o caso base como $n = 2$.

Exemplo 4: o problema da celebridade

Um argumento indutivo que parece ser mais eficiente é o seguinte:

Hipótese de Indução:

Sabemos encontrar uma celebridade (se existir) em um conjunto de $n-1$ *pessoas.*

- **Caso base:** se $n = 1$, podemos considerar que o único elemento é uma celebridade.
- Outra opção é considerar o caso base como $n = 2$.
  A solução é simples: existe uma celebridade se, e somente se, $M[1,2] \oplus M[2,1] = 1$. Mais uma comparação define a celebridade: se $M[1,2] = 0$, então a celebridade é a pessoa 1; senão, é a pessoa 2.

Exemplo 4: o problema da celebridade

Tome então um conjunto $S = \{1, 2, \ldots, n\}, n > 2$, de pessoas e a matriz M associada. Considere o conjunto $S' = S \setminus \{n\}$;

Exemplo 4: o problema da celebridade

Tome então um conjunto $S = \{1, 2, \ldots, n\}, n > 2$, de pessoas e a matriz M associada. Considere o conjunto $S' = S \setminus \{n\}$;

Há dois casos possíveis:

Exemplo 4: o problema da celebridade

Tome então um conjunto $S = \{1, 2, \ldots, n\}, n > 2$, de pessoas e a matriz M associada. Considere o conjunto $S' = S \setminus \{n\}$;

Há dois casos possíveis:

1. Existe uma celebridade em S', digamos a pessoa k. Então, k é celebridade em S se, e somente se, $M[n, k] = 1$ e $M[k, n] = 0$.

Exemplo 4: o problema da celebridade

Tome então um conjunto $S = \{1, 2, \ldots, n\}, n > 2$, de pessoas e a matriz M associada. Considere o conjunto $S' = S \setminus \{n\}$;

Há dois casos possíveis:

1. Existe uma celebridade em S', digamos a pessoa k. Então, k é celebridade em S se, e somente se, $M[n, k] = 1$ e $M[k, n] = 0$.
2. Se k não for celebridade em S ou não existir celebridade em S', então a pessoa n é celebridade em S se $M[n, j] = 0$ e $M[j, n] = 1$ para todo $1 \leq j < n$; caso contrário não há celebridade em S.

Exemplo 4: o problema da celebridade

Tome então um conjunto $S = \{1, 2, \ldots, n\}, n > 2$, de pessoas e a matriz M associada. Considere o conjunto $S' = S \setminus \{n\}$;

Há dois casos possíveis:

1. Existe uma celebridade em S', digamos a pessoa k. Então, k é celebridade em S se, e somente se, $M[n, k] = 1$ e $M[k, n] = 0$.
2. Se k não for celebridade em S ou não existir celebridade em S', então a pessoa n é celebridade em S se $M[n, j] = 0$ e $M[j, n] = 1$ para todo $1 \leq j < n$; caso contrário não há celebridade em S.

Essa primeira tentativa, infelizmente, também conduz a um algoritmo quadrático. Por quê?

A segunda tentativa baseia-se em um fato muito simples:

Exemplo 4 - Segunda tentativa

A segunda tentativa baseia-se em um fato muito simples:

Dadas duas pessoas i e j, é possível determinar se uma delas **não** *é uma celebridade com apenas uma comparação: se $M[i,j]=1$, então i não é celebridade; caso contrário j não é celebridade.*

Exemplo 4 - Segunda tentativa

A segunda tentativa baseia-se em um fato muito simples:

Dadas duas pessoas i e j, é possível determinar se uma delas **não** *é uma celebridade com apenas uma comparação: se $M[i, j] = 1$, então i não é celebridade; caso contrário j não é celebridade.*

Usaremos este argumento aplicando a hipótese de indução sobre o conjunto de $n - 1$ pessoas obtidas removendo dentre as n pessoas alguém que sabemos não ser celebridade.

O caso base e a hipótese de indução são os mesmos que antes.

Exemplo 4 - Segunda tentativa

Sejam então S um conjunto de $n > 2$ pessoas e M a matriz associada.

Exemplo 4 - Segunda tentativa

Sejam então S um conjunto de $n > 2$ pessoas e M a matriz associada.

Sejam i e j quaisquer duas pessoas e suponha que usando o argumento acima determinamos que j não é celebridade.

Sejam então S um conjunto de $n > 2$ pessoas e M a matriz associada.

Sejam i e j quaisquer duas pessoas e suponha que usando o argumento acima determinamos que j não é celebridade.

Seja $S' = S \setminus \{j\}$ e considere os dois casos possíveis:

Exemplo 4 - Segunda tentativa

Sejam então S um conjunto de $n > 2$ pessoas e M a matriz associada.

Sejam i e j quaisquer duas pessoas e suponha que usando o argumento acima determinamos que j não é celebridade.

Seja $S' = S \setminus \{j\}$ e considere os dois casos possíveis:

1. Existe uma celebridade em S', digamos a pessoa k. Se $M[j,k] = 1$ e $M[k,j] = 0$, então k é celebridade em S; caso contrário não há uma celebridade em S.

Exemplo 4 - Segunda tentativa

Sejam então S um conjunto de $n > 2$ pessoas e M a matriz associada.

Sejam i e j quaisquer duas pessoas e suponha que usando o argumento acima determinamos que j não é celebridade.

Seja $S' = S \setminus \{j\}$ e considere os dois casos possíveis:

1. Existe uma celebridade em S', digamos a pessoa k. Se $M[j,k] = 1$ e $M[k,j] = 0$, então k é celebridade em S; caso contrário não há uma celebridade em S.
2. Não existe uma celebridade em S'. Então não existe uma celebridade em S.

CELEBRIDADE(S, M)

$\triangleright$ **Entrada:** uma matriz M associada a um conjunto $S = \{1, 2, \ldots, n\}$ de pessoas.

$\triangleright$ **Saída:** Um inteiro $k \leq n$ que é celebridade em S ou $k = 0$

se $|S| = 1$
    então $k \leftarrow$ elemento em S
    senão
        sejam i, j quaisquer duas pessoas em S
        se $M[i, j] = 1$ **então** $s \leftarrow i$ **senão** $s \leftarrow j$
        $S' \leftarrow S \setminus \{s\}$
        $k \leftarrow$ CELEBRIDADE(S', M)
        se $k > 0$ **então**
            se $(M[s, k] \neq 1)$ **ou** $(M[k, s] \neq 0)$ **então** $k \leftarrow 0$
devolva k

Exemplo 4 - Complexidade

O algoritmo resultante tem complexidade linear em n.

Exemplo 4 - Complexidade

O algoritmo resultante tem complexidade linear em n.

A recorrência $T(n)$ para o número de operações executadas pelo algoritmo é:

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ T(n-1) + \Theta(1), & n > 1. \end{cases}$$

O algoritmo resultante tem complexidade linear em n.

A recorrência $T(n)$ para o número de operações executadas pelo algoritmo é:

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ T(n-1) + \Theta(1), & n > 1. \end{cases}$$

A solução desta recorrência é

$$\sum_{i=1}^{n} \Theta(1) = n\Theta(1) = \Theta(n).$$

Projeto por indução - Exemplo 5
Subseqüência consecutiva máxima (SCM)

Problema:

Dada uma seqüência $X = x_1, x_2, \ldots, x_n$ *de números reais, encontrar uma* subseqüência consecutiva $Y = x_i, x_{i+1}, \ldots, x_j$ *de* X, *onde* $1 \leq i, j \leq n$, *cuja soma seja máxima dentre todas as subseqüências consecutivas.*

Projeto por indução - Exemplo 5
Subseqüência consecutiva máxima (SCM)

Problema:

Dada uma seqüência $X = x_1, x_2, \ldots, x_n$ *de números reais, encontrar uma* subseqüência consecutiva $Y = x_i, x_{i+1}, \ldots, x_j$ *de* X, *onde* $1 \leq i, j \leq n$, *cuja soma seja máxima dentre todas as subseqüências consecutivas.*

Exemplos:

Projeto por indução - Exemplo 5
Subseqüência consecutiva máxima (SCM)

Problema:

Dada uma seqüência $X = x_1, x_2, \ldots, x_n$ *de números reais, encontrar uma* subseqüência consecutiva $Y = x_i, x_{i+1}, \ldots, x_j$ *de* X, *onde* $1 \leq i, j \leq n$, *cuja soma seja máxima dentre todas as subseqüências consecutivas.*

Exemplos:

$X = (4, 2, -7, 3, 0, -2, 1, 5, -2)$ Resp: $Y = (3, 0, -2, 1, 5)$

Projeto por indução - Exemplo 5
Subseqüência consecutiva máxima (SCM)

Problema:

Dada uma seqüência $X = x_1, x_2, \ldots, x_n$ *de números reais, encontrar uma* subseqüência consecutiva $Y = x_i, x_{i+1}, \ldots, x_j$ *de* X, *onde* $1 \leq i, j \leq n$, *cuja soma seja máxima dentre todas as subseqüências consecutivas.*

Exemplos:

$X = (4, 2, -7, 3, 0, -2, 1, 5, -2)$ Resp: $Y = (3, 0, -2, 1, 5)$
$X = (1, -1, 2, 1, -2, 1)$ Resp: $Y = (1, -1, 2, 1)$ ou $Y = (2, 1)$

Subseqüência consecutiva máxima (SCM)

Problema:

Dada uma seqüência $X = x_1, x_2, \ldots, x_n$ *de números reais, encontrar uma* subseqüência consecutiva $Y = x_i, x_{i+1}, \ldots, x_j$ *de* X, *onde* $1 \leq i, j \leq n$, *cuja soma seja máxima dentre todas as subseqüências consecutivas.*

Exemplos:

$X = (4, 2, -7, 3, 0, -2, 1, 5, -2)$ Resp: $Y = (3, 0, -2, 1, 5)$

$X = (1, -1, 2, 1, -2, 1)$ Resp: $Y = (1, -1, 2, 1)$ ou $Y = (2, 1)$

$X = (-1, 0, -2)$ Resp: $Y = (0)$ ou $Y = (\ )$

Projeto por indução - Exemplo 5
Subseqüência consecutiva máxima (SCM)

Problema:

Dada uma seqüência $X = x_1, x_2, \ldots, x_n$ *de números reais, encontrar uma* subseqüência consecutiva $Y = x_i, x_{i+1}, \ldots, x_j$ *de* X, *onde* $1 \leq i, j \leq n$, *cuja soma seja máxima dentre todas as subseqüências consecutivas.*

Exemplos:

$X = (4, 2, -7, 3, 0, -2, 1, 5, -2)$ Resp: $Y = (3, 0, -2, 1, 5)$

$X = (1, -1, 2, 1, -2, 1)$ Resp: $Y = (1, -1, 2, 1)$ ou $Y = (2, 1)$

$X = (-1, 0, -2)$ Resp: $Y = (0)$ ou $Y = (\ )$

$X = (-3, -1)$ Resp: $Y = (\ )$

Exemplo 5 - Indução

Como antes, vamos examinar o que podemos obter de uma **hipótese de indução** simples:

Como antes, vamos examinar o que podemos obter de uma **hipótese de indução** simples:

Hipótese de indução:

Sabemos calcular a SCM *de seqüências de comprimento* $n - 1$.

Exemplo 5 - Indução

Como antes, vamos examinar o que podemos obter de uma **hipótese de indução** simples:

Hipótese de indução:

Sabemos calcular a SCM *de seqüências de comprimento* $n - 1$.

- **Caso base:** $n = 1$. A SCM de $X = x_1$ é $Y = x_1$, se $x_1 \geq 0$, e $Y = (\ )$, caso contrário.

Exemplo 5 - Indução

Como antes, vamos examinar o que podemos obter de uma **hipótese de indução** simples:

Hipótese de indução:

Sabemos calcular a SCM *de seqüências de comprimento* $n - 1$.

- **Caso base:** $n = 1$. A SCM de $X = x_1$ é $Y = x_1$, se $x_1 \geq 0$, e $Y = (\ )$, caso contrário.
- Seja então $X = x_1, x_2, \ldots, x_{n-1}, x_n$ uma seqüência com $n > 1$.

Como antes, vamos examinar o que podemos obter de uma **hipótese de indução** simples:

Hipótese de indução:

Sabemos calcular a SCM *de seqüências de comprimento* $n - 1$.

- **Caso base:** $n = 1$. A SCM de $X = x_1$ é $Y = x_1$, se $x_1 \geq 0$, e $Y = (\ )$, caso contrário.
- Seja então $X = x_1, x_2, \ldots, x_{n-1}, x_n$ uma seqüência com $n > 1$.
- Considere a seqüência $X' = x_1, x_2, \ldots, x_{n-1}$.

Exemplo 5 - Indução

Como antes, vamos examinar o que podemos obter de uma **hipótese de indução** simples:

Hipótese de indução:

Sabemos calcular a SCM *de seqüências de comprimento* $n - 1$.

- **Caso base:** $n = 1$. A SCM de $X = x_1$ é $Y = x_1$, se $x_1 \geq 0$, e $Y = (\ )$, caso contrário.
- Seja então $X = x_1, x_2, \ldots, x_{n-1}, x_n$ uma seqüência com $n > 1$.
- Considere a seqüência $X' = x_1, x_2, \ldots, x_{n-1}$.
- Seja $Y' = x_i, x_{i+1}, \ldots, x_j$ a SCM de X', obtida aplicando-se a HI

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\qquad x_n$

$Y' = x_i, x_{i+1}, \ldots, x_j$ $\qquad Y = ?$

Há três casos a examinar:

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad x_n$

$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad Y = ?$

Há três casos a examinar:

1. $Y' = (\ )$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário..

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad x_n$
$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad Y = ?$

Há três casos a examinar:

1. $Y' = (\ )$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário..
2. $j = n - 1$. Temos $Y = Y' \| x_n$, se $x_n \geq 0$, e $Y = Y'$, caso contrário.

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\qquad x_n$

$Y' = x_i, x_{i+1}, \ldots, x_j$ $\qquad Y = ?$

Há três casos a examinar:

1. $Y' = (\ )$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário..
2. $j = n - 1$. Temos $Y = Y' \| x_n$, se $x_n \geq 0$, e $Y = Y'$, caso contrário.
3. $j < n - 1$. Aqui há dois subcasos a considerar:

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad x_n$
$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad Y = ?$

Há três casos a examinar:

1. $Y' = (\ )$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário..
2. $j = n - 1$. Temos $Y = Y' || x_n$, se $x_n \geq 0$, e $Y = Y'$, caso contrário.
3. $j < n - 1$. Aqui há dois subcasos a considerar:
    1. Y' também é SCM de X; isto é, $Y = Y'$.

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\qquad x_n$
$Y' = x_i, x_{i+1}, \ldots, x_j$ $\qquad Y = ?$

Há três casos a examinar:

1. $Y' = (\ )$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário..
2. $j = n - 1$. Temos $Y = Y' || x_n$, se $x_n \geq 0$, e $Y = Y'$, caso contrário.
3. $j < n - 1$. Aqui há dois subcasos a considerar:
   1. Y' também é SCM de X; isto é, $Y = Y'$.
   2. Y' não é a SCM de X. Isto significa que x_n é parte de uma SCM Y de X. Esta tem que ser da forma $x_k, x_{k+1}, \ldots, x_{n-1}, x_n$, para algum $k \leq n$. Não há informação suficiente na HI para resolver este caso.

Exemplo 5 - Indução

- **O que falta na HI?**

Exemplo 5 - Indução

- **O que falta na HI?**
- É evidente que, quando

$$Y = x_k, x_{k+1}, \ldots, x_{n-1}, x_n,$$

então $Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ é um **sufixo (de soma) máximo(a)** de

$$X' = x_1, x_2, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_{n-1}.$$

- **O que falta na HI?**
- É evidente que, quando

$$Y = x_k, x_{k+1}, \ldots, x_{n-1}, x_n,$$

então $Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ é um **sufixo (de soma) máximo(a)** de

$$X' = x_1, x_2, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_{n-1}.$$

- Assim, se conhecermos o sufixo máximo Z' de X', além da SCM Y' de X', podemos resolver o problema para X.

  Note que permitimos o sufixo vazio.

Exemplo 5 - Indução

Parece então natural enunciar a seguinte HI fortalecida:

Exemplo 5 - Indução

Parece então natural enunciar a seguinte HI fortalecida:

Hipótese de indução reforçada:

Sabemos calcular a SCM *e o sufixo máximo de seqüências de comprimento* $n-1$.

Exemplo 5 - Indução

Parece então natural enunciar a seguinte HI fortalecida:

Hipótese de indução reforçada:

Sabemos calcular a SCM *e o sufixo máximo de seqüências de comprimento* $n-1$.

Caso base: para $n = 1$, a SCM de $X = x_1$ é $Y = x_1$, se $x_1 \geq 0$, e $Y = (\ )$, caso contrário. Nesse caso, o sufixo máximo é igual a SCM.

Parece então natural enunciar a seguinte HI fortalecida:

Hipótese de indução reforçada:

Sabemos calcular a SCM *e o sufixo máximo de seqüências de comprimento* $n - 1$.

Caso base: para $n = 1$, a SCM de $X = x_1$ é $Y = x_1$, se $x_1 \geq 0$, e $Y = (\ )$, caso contrário. Nesse caso, o sufixo máximo é igual a SCM.

Suponha agora que $n > 1$ e sejam Y' e Z' a SCM e o sufixo máximo de $X' = x_1, x_2, \ldots, x_{n-1}$ obtidas da HI.

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad x_n$

$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad Y = ?$

$Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ $\quad Z = ?$

Vamos examinar os mesmos três casos:

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad$ x_n

$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad$ $Y = ?$

$Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ $\quad$ $Z = ?$

Vamos examinar os mesmos três casos:

1. $Y' = (\ )$. Note que $Y' = Z'$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário. Ainda, $Z = Y$.

Exemplo 5 - Indução

$X' = x_1, x_2, \dots, x_{n-1}$ $\qquad x_n$
$Y' = x_i, x_{i+1}, \dots, x_j$ $\qquad Y = ?$
$Z' = x_k, x_{k+1}, \dots, x_{n-1}$ $\qquad Z = ?$

Vamos examinar os mesmos três casos:

1. $Y' = (\ )$. Note que $Y' = Z'$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário. Ainda, $Z = Y$.
2. $j = n - 1$. Note que $Y' = Z'$. Neste caso, $Y = Z' \| x_n$, se $x_n \geq 0$, e $Y = Y'$, caso contrário. Ainda, Z é o maior entre $Z' \| x_n$ e $(\ )$.

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad x_n$

$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad Y = ?$

$Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ $\quad Z = ?$

Vamos examinar os mesmos três casos:

1. $Y' = (\ )$. Note que $Y' = Z'$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário. Ainda, $Z = Y$.
2. $j = n - 1$. Note que $Y' = Z'$. Neste caso, $Y = Z' \| x_n$, se $x_n \geq 0$, e $Y = Y'$, caso contrário. Ainda, Z é o maior entre $Z' \| x_n$ e $(\ )$.
3. $j < n - 1$.

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad x_n$
$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad Y = ?$
$Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ $\quad Z = ?$

Vamos examinar os mesmos três casos:

1. $Y' = (\ )$. Note que $Y' = Z'$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário. Ainda, $Z = Y$.
2. $j = n - 1$. Note que $Y' = Z'$. Neste caso, $Y = Z' \| x_n$, se $x_n \geq 0$, e $Y = Y'$, caso contrário. Ainda, Z é o maior entre $Z' \| x_n$ e $(\ )$.
3. $j < n - 1$.
    1. Se $Z' \| x_n$ for maior que Y', então $Y = Z = Z' \| x_n$.

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad$ x_n
$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad$ $Y = ?$
$Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ $\quad$ $Z = ?$

Vamos examinar os mesmos três casos:

1. $Y' = (\ )$. Note que $Y' = Z'$. Neste caso, $Y = x_n$, se $x_n \geq 0$, e $Y = (\ )$, caso contrário. Ainda, $Z = Y$.
2. $j = n - 1$. Note que $Y' = Z'$. Neste caso, $Y = Z' || x_n$, se $x_n \geq 0$, e $Y = Y'$, caso contrário. Ainda, Z é o maior entre $Z' || x_n$ e $(\ )$.
3. $j < n - 1$.
   1. Se $Z' || x_n$ for maior que Y', então $Y = Z = Z' || x_n$.
   2. Caso contrário, $Y = Y'$ e Z é o maior entre $Z' || x_n$ e $(\ )$.

Exemplo 5 - Indução

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad x_n$
$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad Y = ?$
$Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ $\quad Z = ?$

A análise pode ser simplificada observando o seguinte:

$X' = x_1, x_2, \ldots, x_{n-1}$ $\quad x_n$
$Y' = x_i, x_{i+1}, \ldots, x_j$ $\quad Y = ?$
$Z' = x_k, x_{k+1}, \ldots, x_{n-1}$ $\quad Z = ?$

A análise pode ser simplificada observando o seguinte:

(a) $Y = Z = Z'||x_n$, OU
(b) $Y = Y'$ e ($Z = Z'||x_n$ ou $Z = (\ )$).

Se $Z'||x_n$ for maior que Y' então ocorre (a), caso contrário, ocorre (b).

$\mathrm{SCM}(X, n)$

▷ **Entrada:** um inteiro n e uma seqüência de n números reais $X = x_1, x_2, \ldots, x_n$.

▷ **Saída:** Inteiros i, j, k e reais $MaxSeq, MaxSuf$ tais que:

- x_i, x_j são o primeiro e o último elemento da SCM de X, cujo valor é $MaxSeq$;
- x_k é o primeiro elemento do sufixo máximo de X, cujo valor é $MaxSuf$;
- $j = 0$ significa que X é composta de reais negativos somente. Neste caso, $MaxSeq = 0$;
- $k = 0$ significa que o sufixo máximo de X é vazio. Neste caso, $MaxSuf = 0$.

Exemplo 5 - Algoritmo

$\mathrm{SCM}(X, n)$

se $n = 1$
    então
        se $x_1 < 0$
            então $i, j, k \leftarrow 0$; $MaxSeq, MaxSuf \leftarrow 0$
            senão $i, j, k \leftarrow 1$; $MaxSeq, MaxSuf \leftarrow x_1$
    senão
        $(i, j, k, MaxSeq, MaxSuf) \leftarrow \mathrm{SCM}(X, n-1)$
        se $k = 0$ **então** $k \leftarrow n$ $\triangleright$ sufixo era negativo
        $MaxSuf \leftarrow MaxSuf + x_n$ $\triangleright Z' || x_n$
        se $MaxSuf > MaxSeq$ $\triangleright Z' || x_n$ é maior que Y'?
            então $i \leftarrow k$; $j \leftarrow n$; $MaxSeq \leftarrow MaxSuf$
            senão se $MaxSuf < 0$ **então** $MaxSuf \leftarrow 0$; $k \leftarrow 0$
    devolva $i, j, k, MaxSeq, MaxSuf$

Exemplo 5 - Complexidade

A complexidade $T(n)$ de SCM é simples de ser calculada.

Exemplo 5 - Complexidade

A complexidade $T(n)$ de SCM é simples de ser calculada.

Como no último exemplo,

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ T(n-1) + \Theta(1), & n > 1. \end{cases}$$

A complexidade $T(n)$ de SCM é simples de ser calculada.

Como no último exemplo,

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ T(n-1) + \Theta(1), & n > 1. \end{cases}$$

A solução desta recorrência é

$$\sum_{i=1}^{n} \Theta(1) = n\Theta(1) = \Theta(n).$$

Exemplo 5 - Complexidade

A complexidade $T(n)$ de SCM é simples de ser calculada.

Como no último exemplo,

$$T(n) = \begin{cases} \Theta(1), & n = 1 \\ T(n-1) + \Theta(1), & n > 1. \end{cases}$$

A solução desta recorrência é

$$\sum_{i=1}^{n} \Theta(1) = n\Theta(1) = \Theta(n).$$

Reforçar a hipótese de indução: preciso lembrar disso ...