

Análise Projeto de Algoritmos

Loana Tito Nogueira

19-Março-2012

Indução Matemática

- Na **Demonstração por Indução**, queremos demonstrar a validade de $P(n)$, uma propriedade P com um parâmetro natural n associado, para todo valor de n

Indução Matemática

- Na **Demonstração por Indução**, queremos demonstrar a validade de $P(n)$, uma propriedade P com um parâmetro natural n associado, para todo valor de n
- Há um número infinito de casos a serem considerados, um para cada valor de n . Demonstramos os infinitos casos de uma só vez:

Indução Matemática

- Na **Demonstração por Indução**, queremos demonstrar a validade de $P(n)$, uma propriedade P com um parâmetro natural n associado, para todo valor de n
- Há um número infinito de casos a serem considerados, um para cada valor de n . Demonstramos os infinitos casos de uma só vez:
 - **Base da Indução:** Demonstramos $P(1)$

Indução Matemática

- Na **Demonstração por Indução**, queremos demonstrar a validade de $P(n)$, uma propriedade P com um parâmetro natural n associado, para todo valor de n
- Há um número infinito de casos a serem considerados, um para cada valor de n . Demonstramos os infinitos casos de uma só vez:
 - **Base da Indução:** Demonstramos $P(1)$
 - **Hipótese de Indução:** Supomos que $P(n)$ é verdadeiro

Indução Matemática

- Na **Demonstração por Indução**, queremos demonstrar a validade de $P(n)$, uma propriedade P com um parâmetro natural n associado, para todo valor de n
- Há um número infinito de casos a serem considerados, um para cada valor de n . Demonstramos os infinitos casos de uma só vez:
 - **Base da Indução:** Demonstramos $P(1)$
 - **Hipótese de Indução:** Supomos que $P(n)$ é verdadeiro
 - **Passo de Indução:** Provamos que $P(n+1)$ é verdadeiro, a partir da hipótese de indução

Exemplo: Prove, por indução, $\sum_{i=1}^k (2i-1) = k^2$

Indução Forte X Indução Fraca

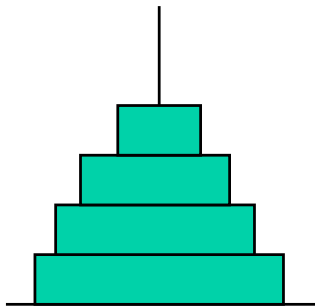
- A indução forte difere da indução fraca (ou simples) apenas na suposição da hipótese
- No caso da indução forte, devemos supor que a propriedade vale para todos os casos anteriores, não somente para o anterior, ou seja:
- **Base da indução:** demonstramos $P(1)$
- **Hipótese de Indução Forte:** Supomos que $P(k)$ é verdadeiro, para todo $k \leq n$
- **Passo da Indução:** Provamos que $P(n)$ é verdadeiro a partir da Hipótese de Indução

Exercícios:

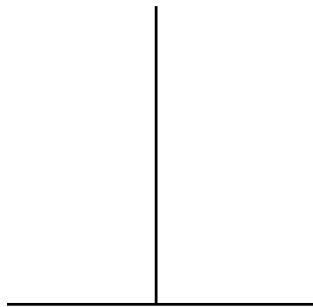
- 1- Demonstre que para todo natural x e n , $x^n - 1$ é divisível por $x - 1$
- Mostre que $\sum_{i=1}^n 3+5i = 2,5n^2 + 5.5n$
- Mostre que $\sum_{i=1}^n i = n(n+1)/2$
- Prove que todo número pode ser escrito como a soma de diferentes potências de 2.

Torre de Hanói

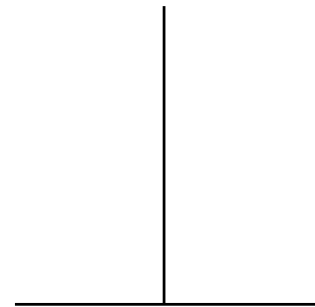
Torre de Hanói



A

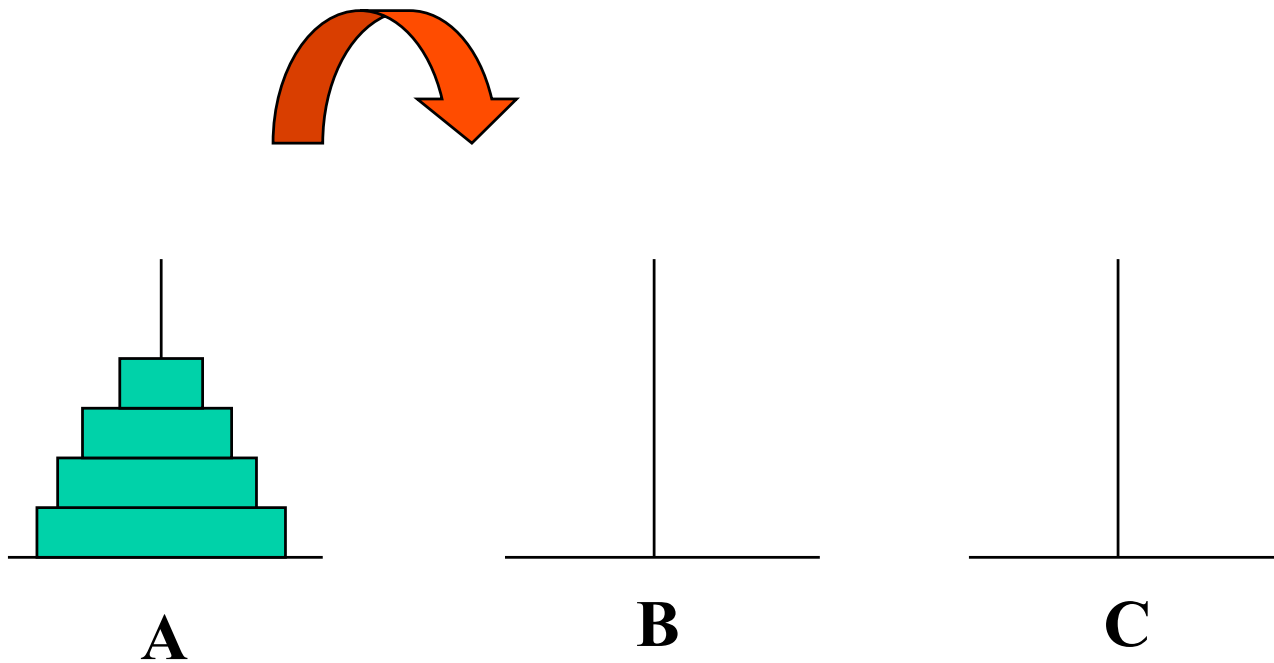


B

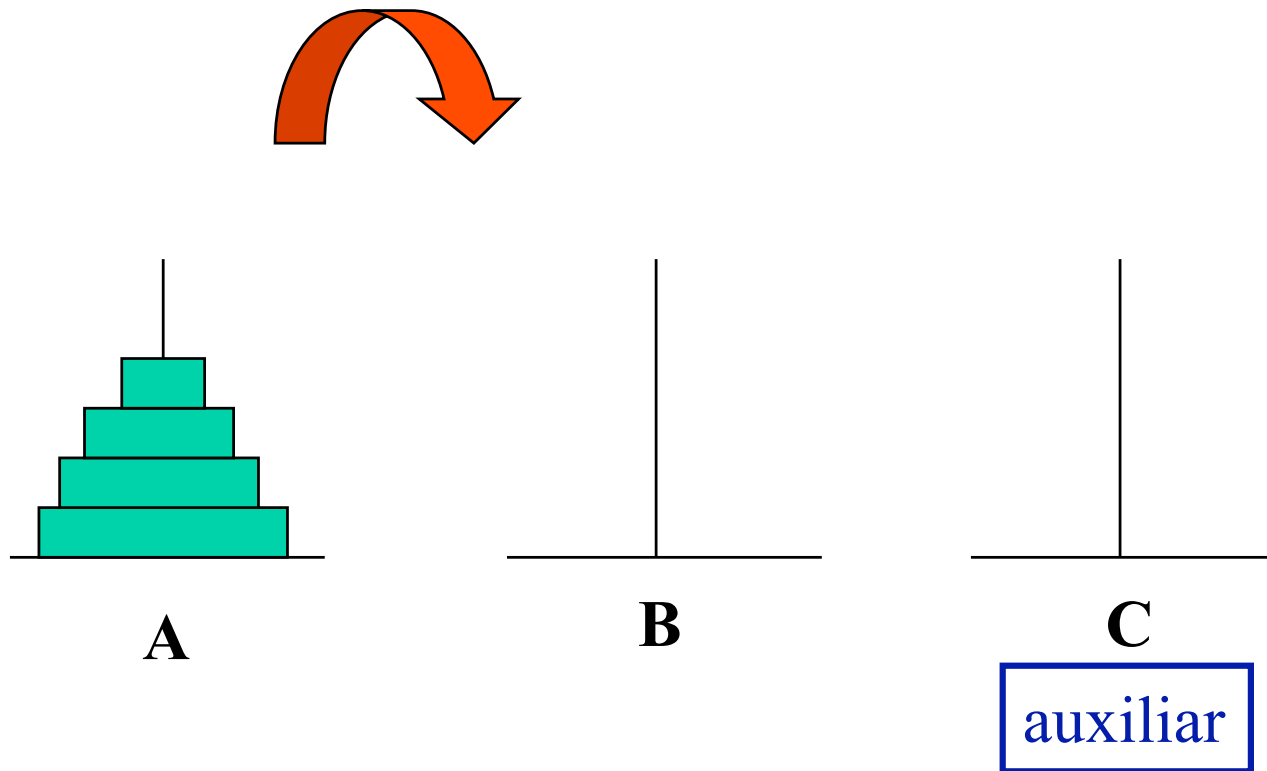


C

Torre de Hanói



Torre de Hanói



Torre de Hanói - Algoritmo

Algoritmo HANOI(n, de A para B usando C)
[Torre de Hanoi, n Discos]

Torre de Hanói - Algoritmo

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$

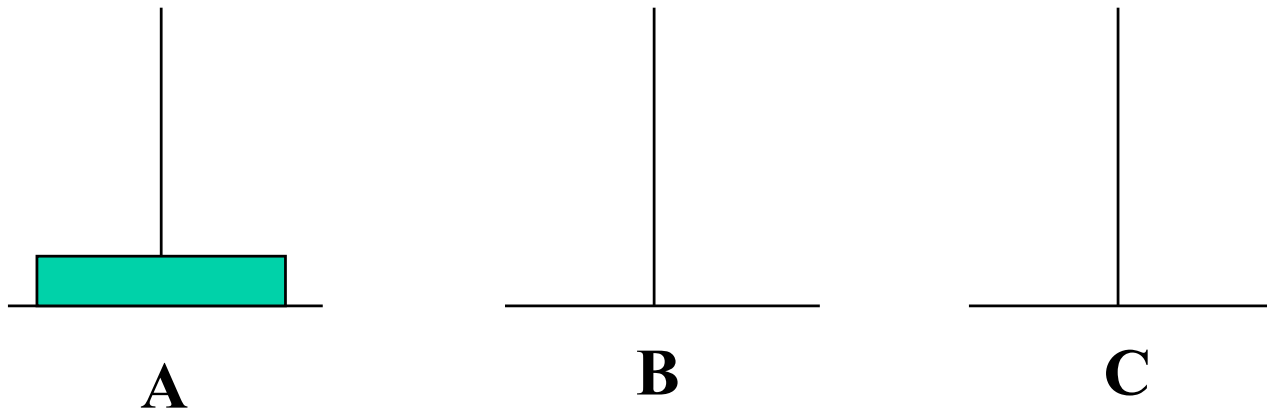
Torre de Hanói - Algoritmo

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$



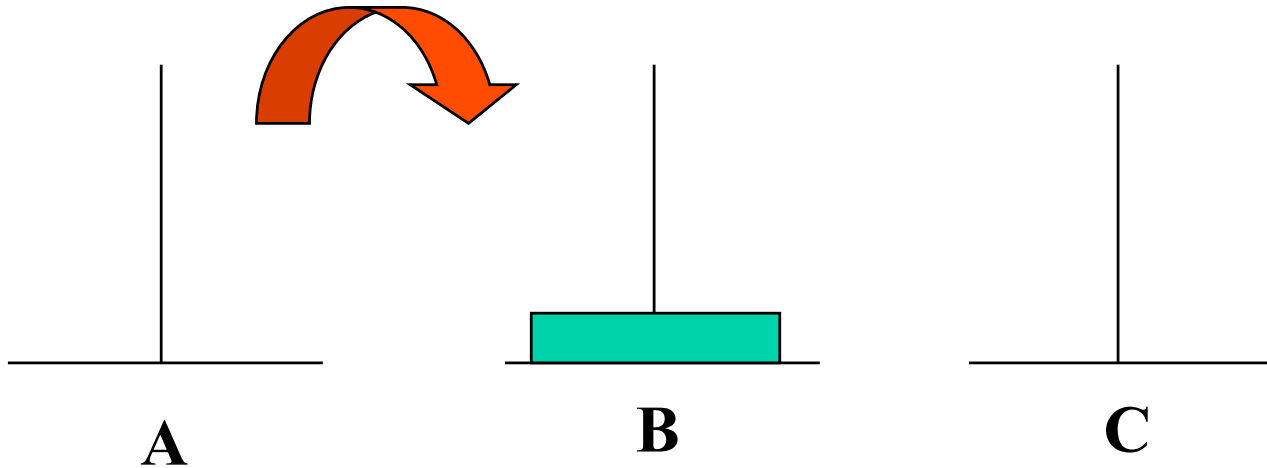
Torre de Hanói - Algoritmo

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] Se $n=1$ então $A \rightarrow B$



Torre de Hanói - Algoritmo

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$

Caso contrário

Torre de Hanói - Algoritmo

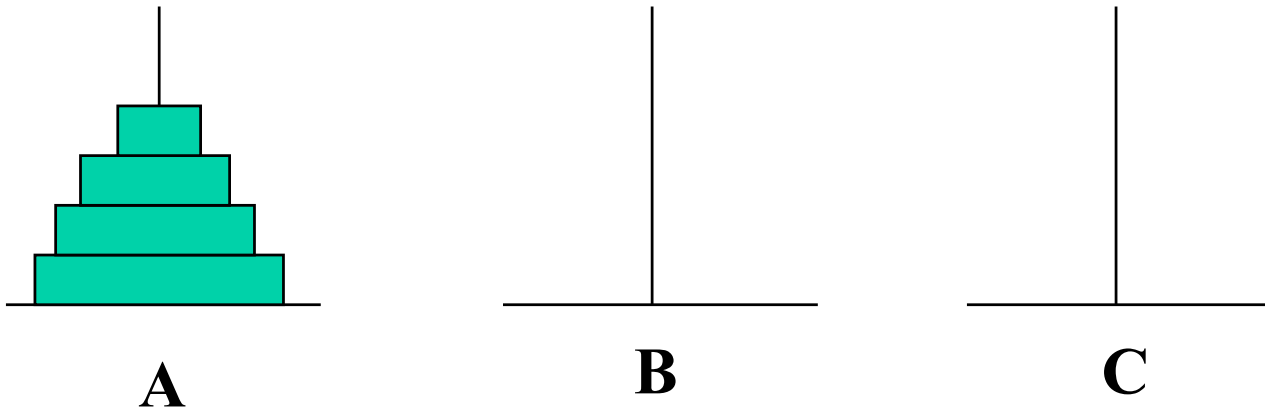
Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] Se $n=1$ então $A \rightarrow B$

Caso contrário



Torre de Hanói - Algoritmo

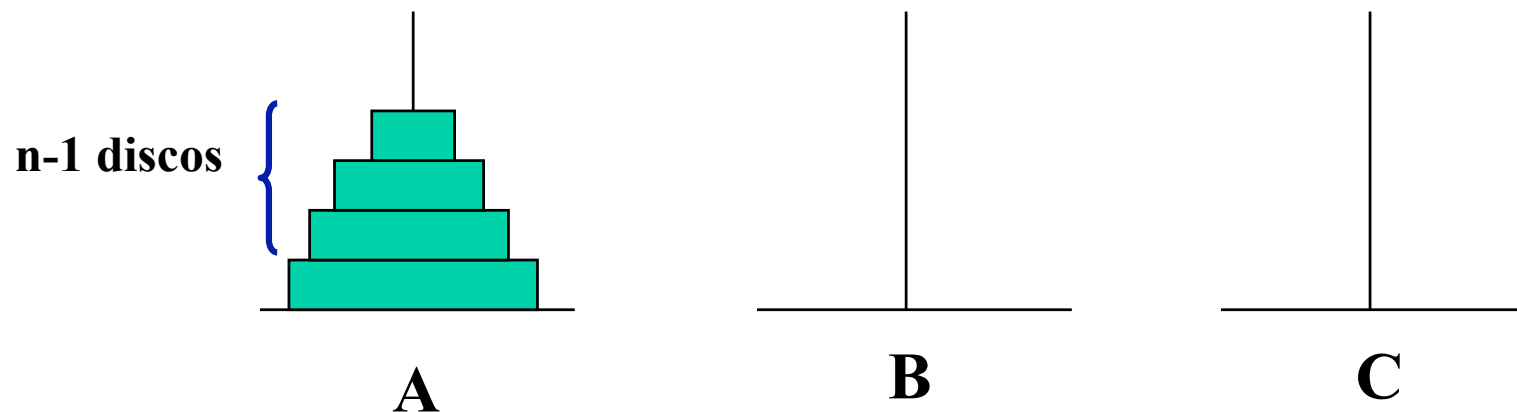
Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] Se $n=1$ então $A \rightarrow B$

Caso contrário



Torre de Hanói - Algoritmo

Algoritmo HANOI(n, de A para B usando C)

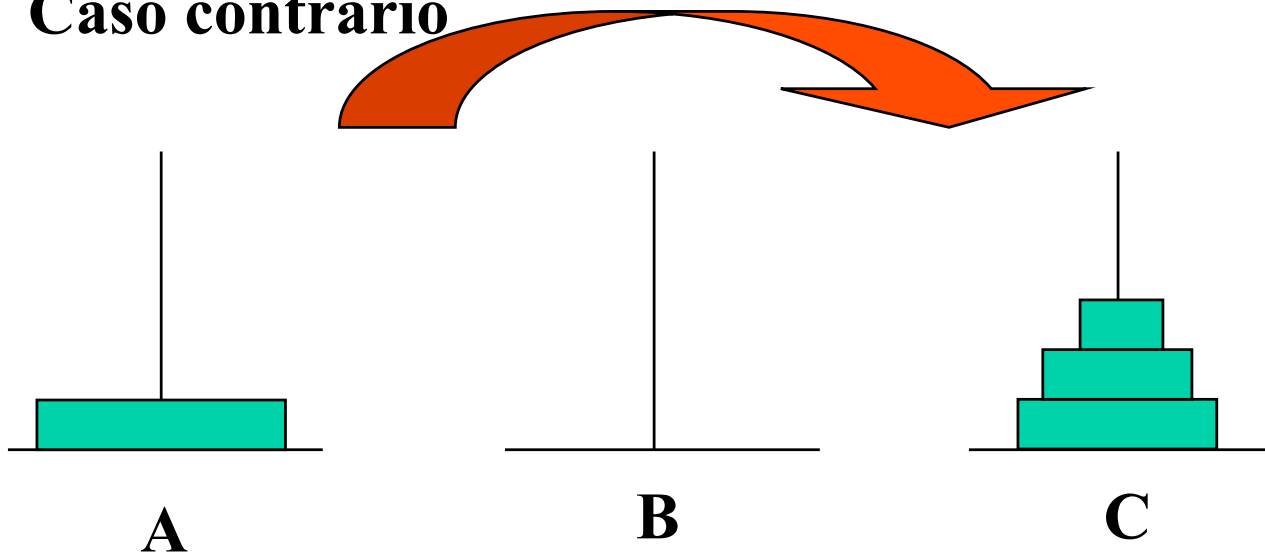
[Torre de Hanoi, n Discos]

Início

[1] Se $n=1$ então $A \rightarrow B$

Caso contrário

HANOI(n-1, de A para C usando B);



Torre de Hanói - Algoritmo

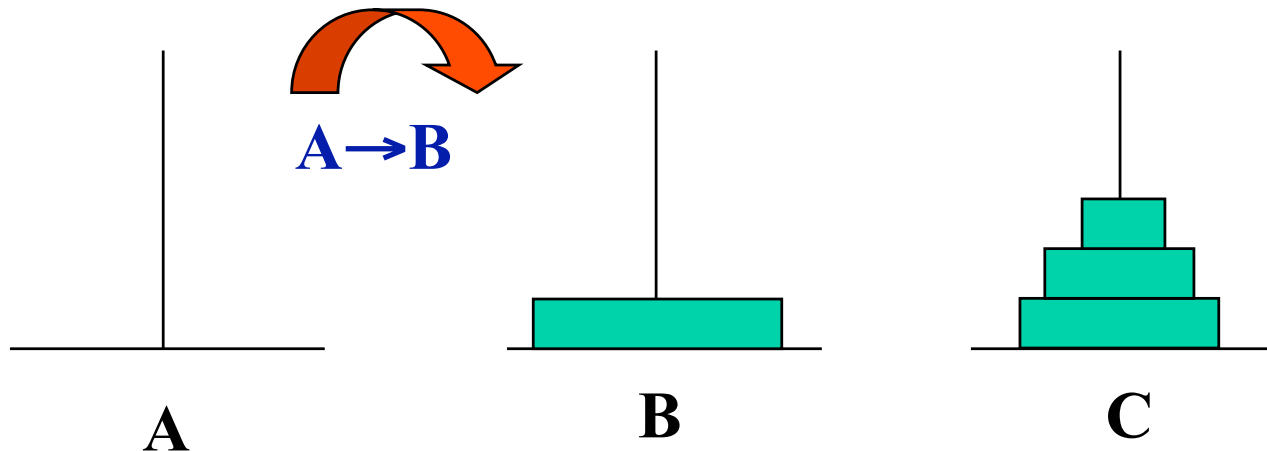
Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] Se $n=1$ então $A \rightarrow B$

Caso contrário



Torre de Hanói - Algoritmo

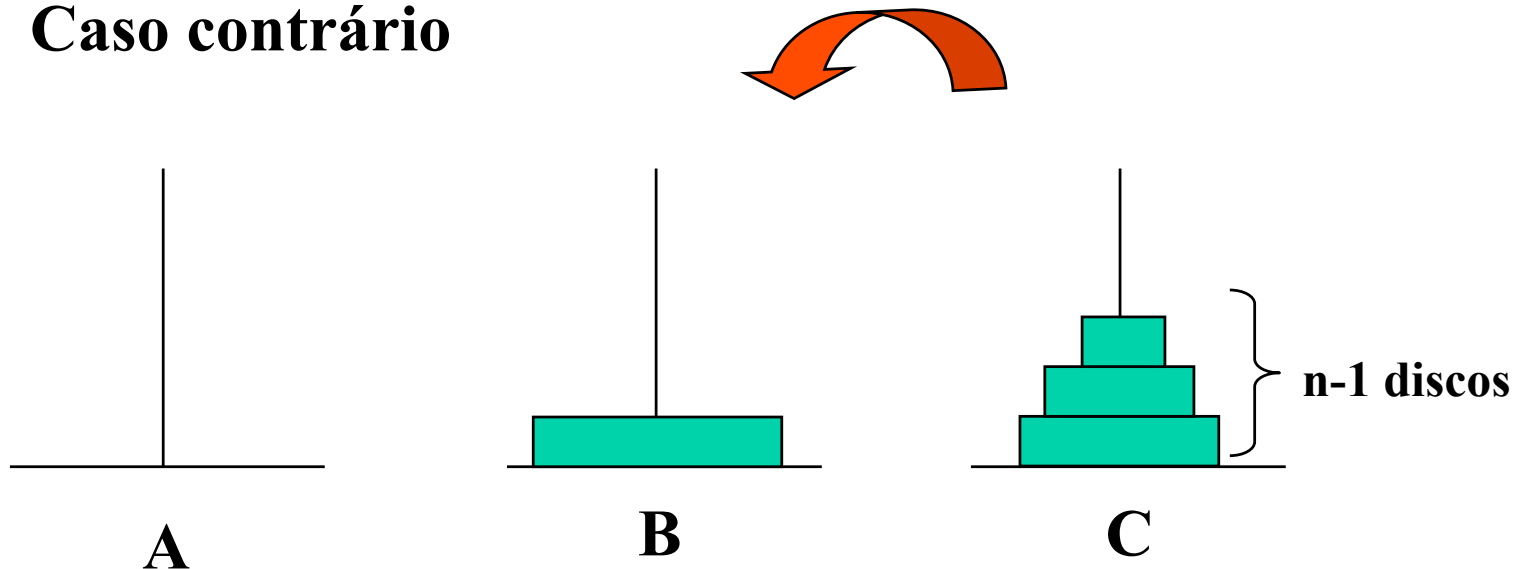
Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] Se $n=1$ então $A \rightarrow B$

Caso contrário



Torre de Hanói - Algoritmo

Algoritmo HANOI(n, de A para B usando C)

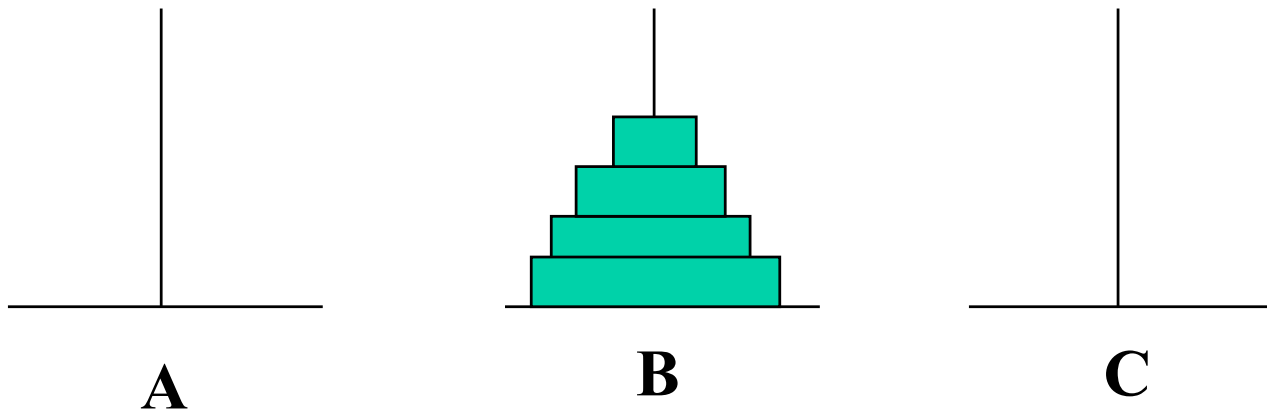
[Torre de Hanoi, n Discos]

Início

[1] Se $n=1$ então $A \rightarrow B$

Caso contrário

HANOI(n-1, de C para B usando A);



Torre de Hanói - Algoritmo

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$

Caso contrário

Início

[2] **HANOI**(n-1, de A para C usando B);

[3] $A \rightarrow B$;

[4] **HANOI**(N-1, DE C PARA B USANDO A)

Fim;

Fim.

O Algoritmo recursivo HANOI resolve o problema das Torres de Hanoi com n discos

O Algoritmo recursivo HANOI resolve o problema das Torres de Hanoi com n discos

- Prova (por indução)
- Base da indução
- Hipótese Indutiva
- Passo da Indução

O Algoritmo recursivo HANOI resolve o problema das Torres de Hanoi com n discos

- Prova (por indução)
- Base da indução $n=1$ (um único disco)
- Hipótese Indutiva
- Passo da Indução

O Algoritmo recursivo HANOI resolve o problema das Torres de Hanoi com n discos

- Prova (por indução)
- Base da indução $n=1$ (um único disco)

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] Se $n=1$ então $A \rightarrow B$

Caso contrário

Início

[2] HANOI($n-1$, de A para C usando B);

[3] $A \rightarrow B$;

[4] HANOI($n-1$, DE C PARA B USANDO A)

Fim;

Fim.

O Algoritmo recursivo HANOI resolve o problema das Torres de Hanoi com n discos

- Prova (por indução)
- Base da indução $n=1$ (um único disco) OK!!
- Hipótese Indutiva
- Passo da Indução

O Algoritmo recursivo HANOI resolve o problema das Torres de Hanoi com n discos

- Prova (por indução)
- **Base da indução** $n=1$ (um único disco) OK!!
- **Hipótese Indutiva:** O algoritmo funciona corretamente para $n-1$ discos (os $n-1$ discos são movidos corretamente de A para Z)
- **Passo da Indução:**

O Algoritmo recursivo HANOI resolve o problema das Torres de Hanoi com n discos

- Prova (por indução)
- **Base da indução** $n=1$ (um único disco) OK!!
- **Hipótese Indutiva:** O algoritmo funciona corretamente para $n-1$ discos (os $n-1$ discos são movidos corretamente de A para Z)
- **Passo da Indução:** suponha que o suporte X tem $n>1$ discos inicialmente

O Algoritmo recursivo HANOI resolve o problema das Torres de Hanoi com n discos

- Prova (por indução)
- **Base da indução** $n=1$ (um único disco) OK!!
- **Hipótese Indutiva:** O algoritmo funciona corretamente para $n-1$ discos (os $n-1$ discos são movidos corretamente de A para Z)
- **Passo da Indução:** suponha que o suporte X tem $n>1$ discos inicialmente

•
•
•

Recorrência

- É uma equação ou desigualdade que descreve uma função em termos de seus valores em entradas menores.

Recorrência

- É uma equação ou desigualdade que descreve uma função em termos de seus valores em entradas menores.
- Quando um algoritmo contém uma chamada recursiva a ele próprio, seu tempo de execução pode frequentemente ser descrito por uma recorrência.

Exemplo: Torre de Hanói

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$

Caso contrário

Início

[2] **HANOI**(n-1, de A para C usando B);

[3] $A \rightarrow B$;

[4] **HANOI**(N-1, DE C PARA B USANDO A)

Fim;

Fim

Exemplo: Torre de Hanói

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$ (1)

Caso contrário

Início

[2] **HANOI**(n-1, de A para C usando B);

[3] $A \rightarrow B$;

[4] **HANOI**(N-1, DE C PARA B USANDO A)

Fim;

Fim

Exemplo: Torre de Hanói

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$ (1)

Caso contrário

Início

[2] **HANOI**(n-1, de A para C usando B); $T(n-1)$

[3] $A \rightarrow B$;

[4] **HANOI**(N-1, DE C PARA B USANDO A)

Fim;

Fim

Exemplo: Torre de Hanói

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$ (1)

Caso contrário

Início

[2] **HANOI**(n-1, de A para C usando B); $T(n-1)$

[3] $A \rightarrow B$; (1)

[4] **HANOI**(N-1, DE C PARA B USANDO A)

Fim;

Fim

Exemplo: Torre de Hanói

Algoritmo HANOI(n, de A para B usando C)

[Torre de Hanoi, n Discos]

Início

[1] **Se** $n=1$ **então** $A \rightarrow B$ (1)

Caso contrário

Início

[2] **HANOI**(n-1, de A para C usando B); $T(n-1)$

[3] $A \rightarrow B$; (1)

[4] **HANOI**(N-1, DE C PARA B USANDO A) $T(n-1)$

Fim;

Fim

Exemplo: Torre de Hanói

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

Exemplo: Torre de Hanói

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

Lembre-se que $\sum_{j=1}^n a^j = \frac{a^{n+1} - 1}{a - 1}$

Resolvendo algumas recorrências

$$T(n) = \begin{cases} 1, & \text{se } n = 1 \\ 2T(n/2) + 1, & \text{se } n > 1 \end{cases}$$

Resolvendo algumas recorrências

$$T(n) = \begin{cases} 1, & \text{se } n = 1 \\ 2T(n/2) + n^2, & \text{se } n > 1 \end{cases}$$

Resolvendo algumas recorrências

$$T(n) = \begin{cases} 1, & \text{se } n = 1 \\ 2T(n/2) + n, & \text{se } n > 1 \end{cases}$$