

Princípios de Desenvolvimento de Algoritmos **MAC122**

Prof. Dr. Paulo Miranda
IME-USP

Funções Recursivas

Funções Recursivas

- **Definição:**

- Uma função é dita **recursiva** quando dentro dela é feita uma ou mais chamadas a ela mesma.
- A idéia é dividir um problema original em subproblemas menores de mesma natureza (**divisão**) e depois combinar as soluções obtidas para gerar a solução do problema original de tamanho maior (**conquista**).
- Os subproblemas são resolvidos recursivamente do mesmo modo em função de instâncias menores, até se tornarem problemas triviais que são resolvidos de forma direta, interrompendo a recursão.

Funções Recursivas

- **Exemplo:** Calcular o fatorial de um número.
 - **Solução não recursiva**

```
#include <stdio.h>

float fatorial(int n){
    float fat = 1.0;
    while(n>1){
        fat *= n;
        n--;
    }
    return fat;
}

int main(){
    float fat;
    fat = fatorial(6);
    printf("fatorial: %f\n",fat);
    return 0;
}
```

Funções Recursivas

- **Exemplo:** Calcular o fatorial de um número.
 - **Solução recursiva:** $n! = n.(n-1)!$

```
#include <stdio.h>

float fatorial(int n){
    if(n==0)          /* Caso trivial */
        return 1.0; /* Solução direta */

    return n*fatorial(n-1); /* Chamada recursiva */
}

int main(){
    float fat;
    fat = fatorial(6);
    printf("fatorial: %f\n",fat);
    return 0;
}
```

Funções Recursivas

- **Exemplo:** Calcular x elevado a n positivo.
 - **Solução não recursiva**

```
#include <stdio.h>

float potencia(float x, int n) {
    float pot=1.0;
    while(n>0) {
        pot *= x;
        n--;
    }
    return pot;
}
```

Funções Recursivas

- **Exemplo:** Calcular x elevado a n positivo.
 - **Solução recursiva:** $x^n = x \cdot x^{(n-1)}$

```
#include <stdio.h>
```

```
float potencia(float x, int n){  
    if(n==0)          /* Caso trivial */  
        return 1.0;  /* Solução direta */  
    else  
        return x*potencia(x, n-1); /*Chamada recursiva*/  
}
```

Funções Recursivas

- **Exemplo:** Calcular x elevado a n positivo.
 - **Solução recursiva:** $x^n = x^{(n/2)} \cdot x^{(n/2)} = (x^{(n/2)})^2$

```
#include <stdio.h>

/* Função recursiva mais eficiente */
float potencia(float x, int n){
    float pot;

    if(n==0) return 1.0; /* Caso trivial */
    if(n%2==0){ /* Se n é par... */
        pot = potencia(x, n/2);
        return pot*pot;
    }
    else{ /* Se n é ímpar... */
        pot = potencia(x, n/2);
        return pot*pot*x;
    }
}
```

Funções Recursivas

- **Exemplo:** Encontrar maior elemento de um vetor.
 - **Solução recursiva**

```
#include <stdio.h>
int maiorinteiro(int v[], int n){
    int m;
    if(n==1) return v[0]; /* Caso trivial */
    else{
        m = maiorinteiro(v, n-1);
        if(m>v[n-1]) return m;
        else return v[n-1];
    }
}
int main(){
    int max,v[5]={8,1,9,4,2};
    max = maiorinteiro(v, 5);
    printf("Max: %d\n",max);
    return 0;
}
```


Funções Recursivas

- **Exemplo:** Imprimir elementos de um vetor.

- **Solução não recursiva**

```
#include <stdio.h>

void printvetor(int v[], int n){
    int i;
    for(i=0; i<n; i++)
        printf("%d ",v[i]);
}
```

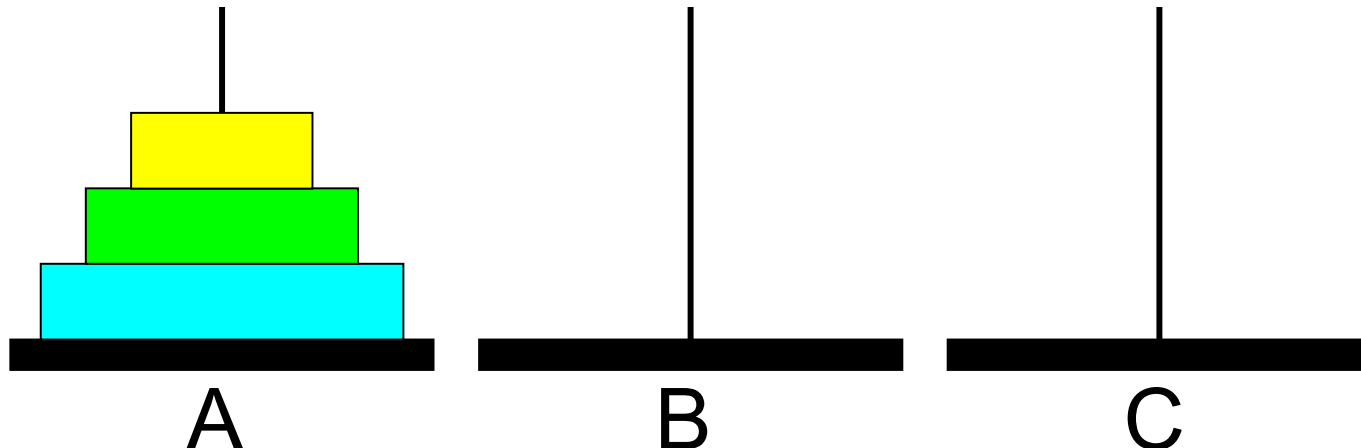
- **Solução recursiva**

```
#include <stdio.h>

void printvetor(int v[], int n){
    if(n>1)
        printvetor(v, n-1);
    printf("%d ",v[n-1]);
}
```

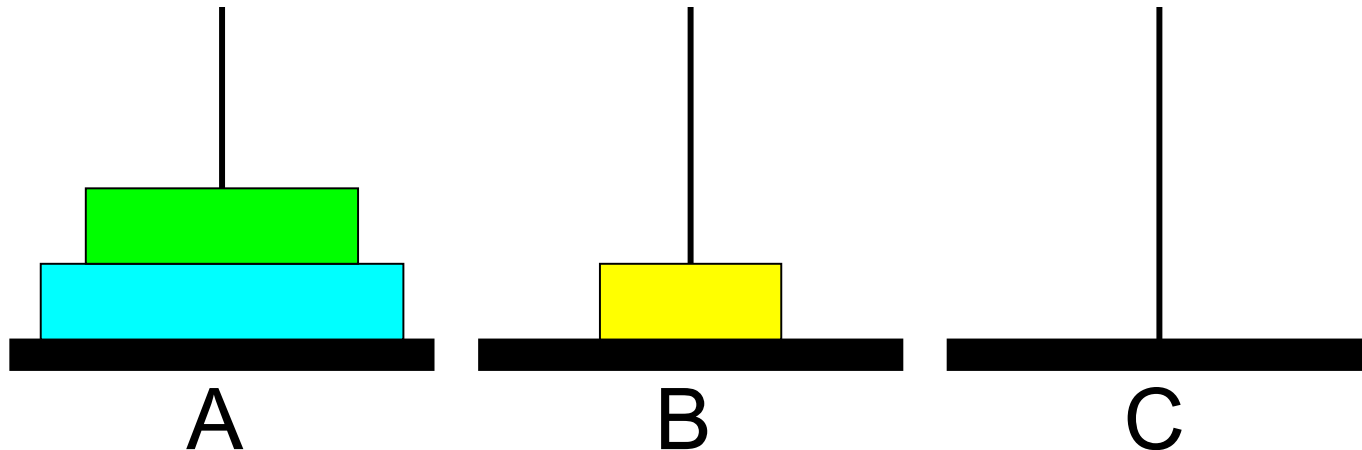
Funções Recursivas

- **Exemplo: Torre de Hanoi**
 - São dados um conjunto de N discos com diferentes tamanhos e três bases A, B e C.
 - O problema consiste em imprimir os passos necessários para transferir os discos da base A para a base B, usando a base C como auxiliar, nunca colocando discos maiores sobre menores.



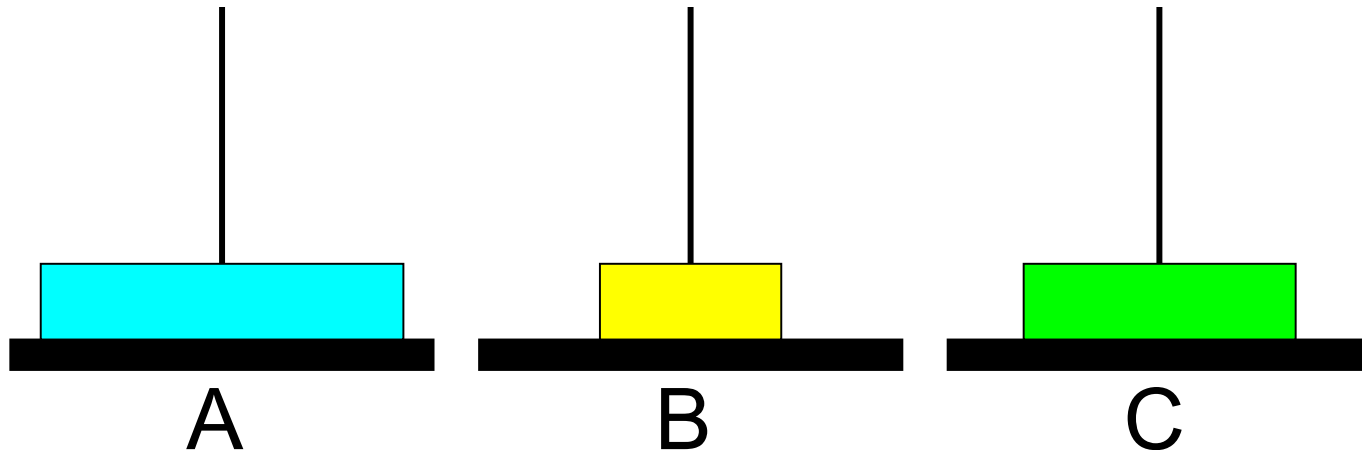
Funções Recursivas

- **Exemplo: Torre de Hanoi**
 - 1º passo: Mover de A para B.



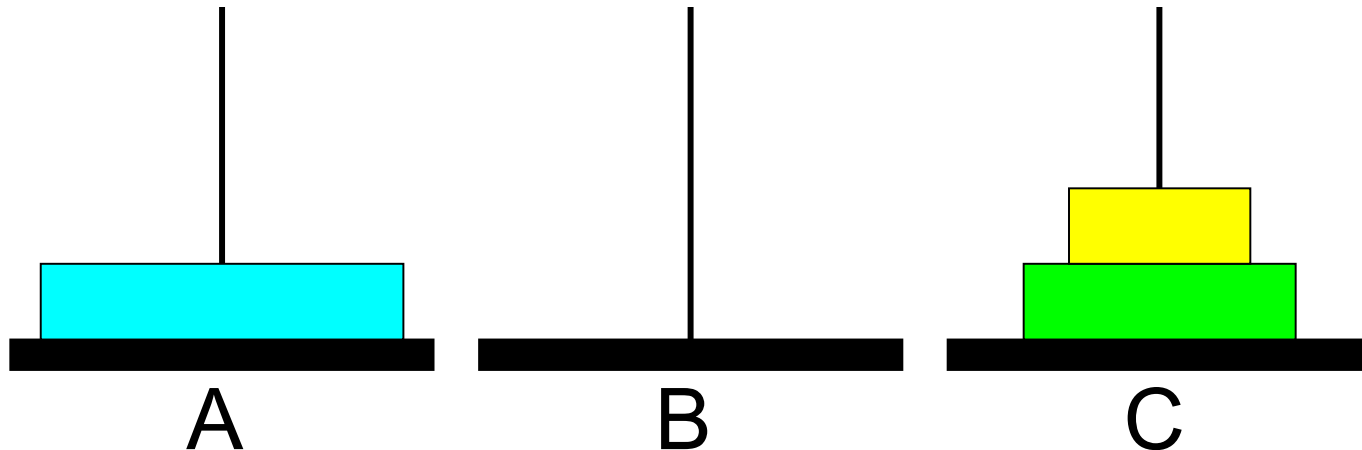
Funções Recursivas

- **Exemplo: Torre de Hanoi**
 - 2º passo: Mover de A para C.



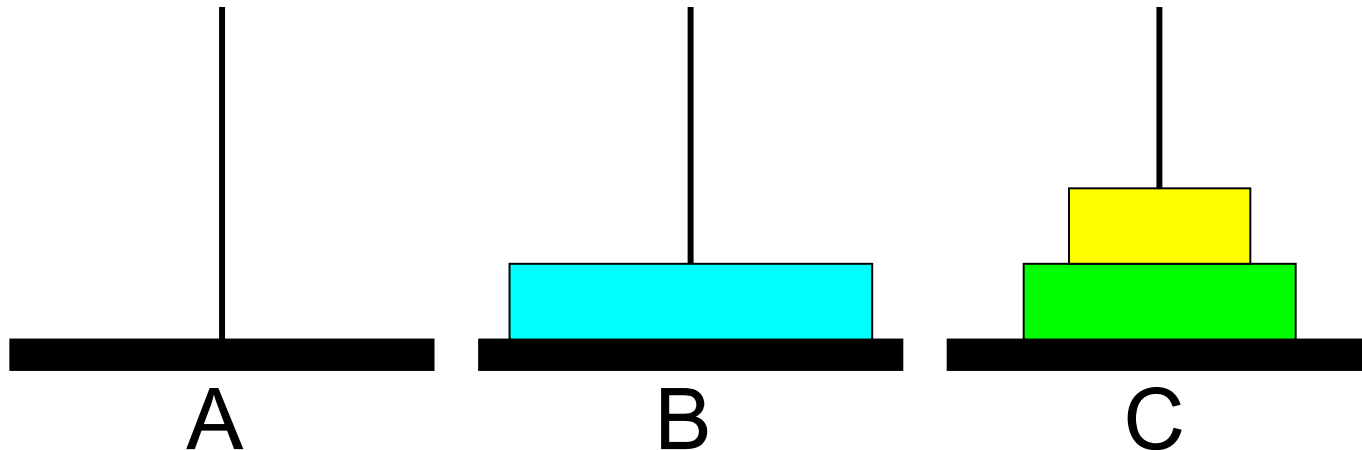
Funções Recursivas

- **Exemplo: Torre de Hanoi**
 - 3º passo: Mover de B para C.



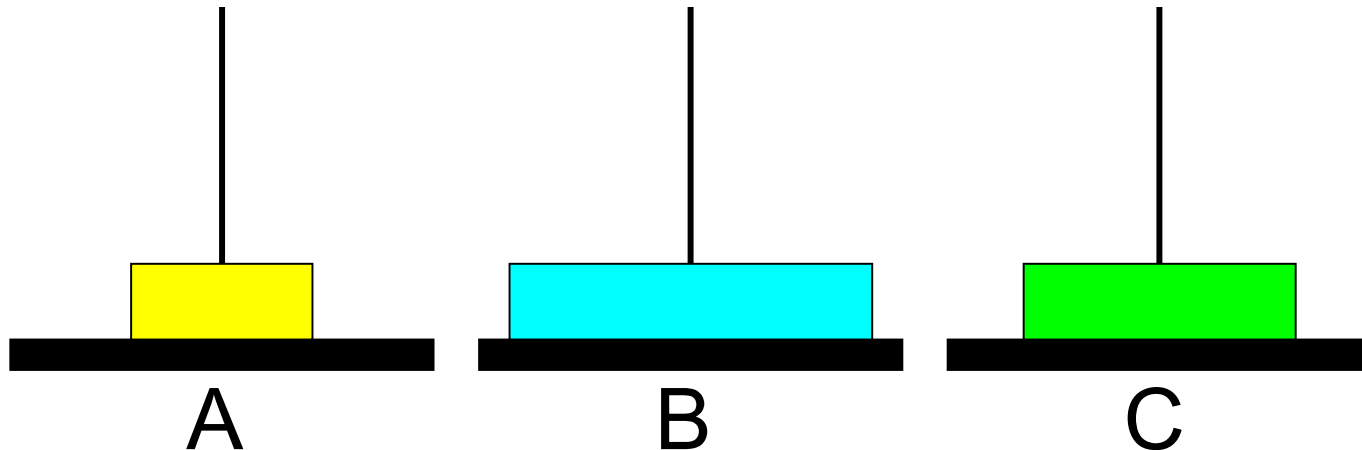
Funções Recursivas

- **Exemplo: Torre de Hanoi**
 - 4º passo: Mover de A para B.



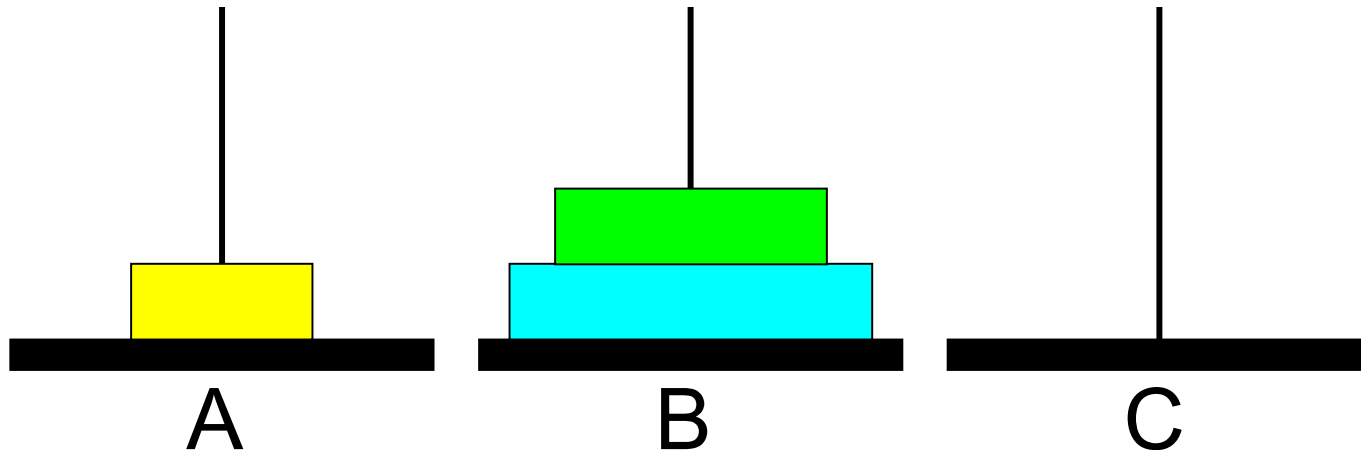
Funções Recursivas

- **Exemplo: Torre de Hanoi**
 - 5º passo: Mover de C para A.



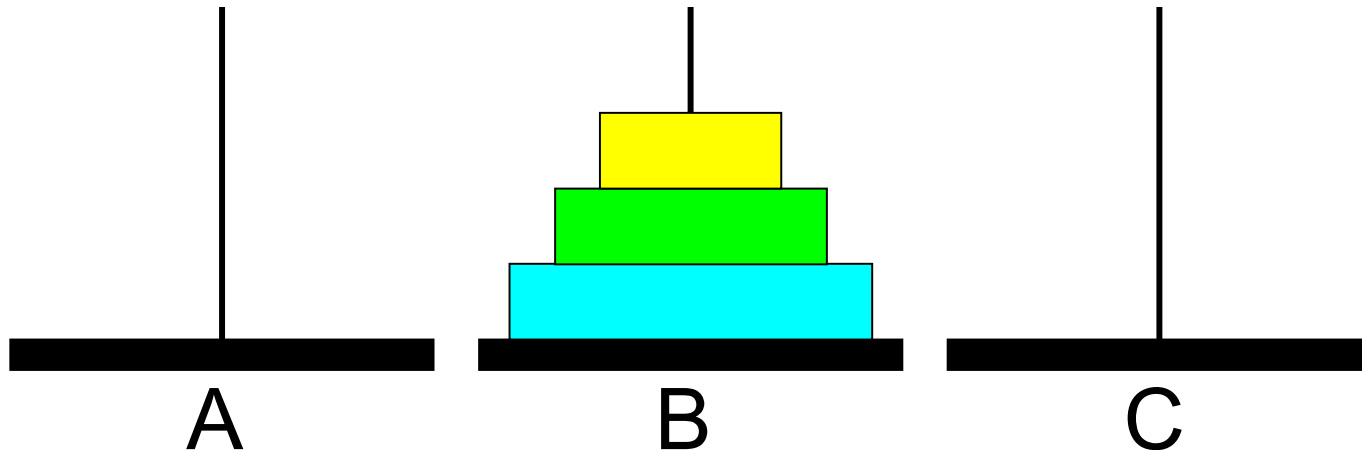
Funções Recursivas

- **Exemplo: Torre de Hanoi**
 - 6º passo: Mover de C para B.



Funções Recursivas

- **Exemplo: Torre de Hanoi**
 - 7º passo: Mover de A para B.



Funções Recursivas

- Exemplo: Torre de Hanoi

```
#include <stdio.h>
void hanoi(int n, char orig, char dest, char aux) {
    if(n==1)
        printf("1 -> %c\n",dest);
    else{
        hanoi(n-1, orig, aux, dest);
        printf("%d -> %c\n",n,dest);
        hanoi(n-1, aux, dest, orig);
    }
}
int main() {
    int n;
    printf("Número de discos: ");
    scanf("%d",&n);
    hanoi(n, 'A', 'B', 'C');
    return 0;
}
```