

TECNOLOGIA EM INFORMÁTICA

Estrutura de Dados

Prof. Glauco da Silva
contato: glaucoslv@gmail.com

Roteiro

- *Objetivos*
- *Recursão (Recursividade)*
- *Exercícios*

Roteiro

- ***Objetivos***
- *Recursão (Recursividade)*
- *Exercícios*

Objetivos

- *Apresentar o conceito de recursão e sua utilização em aplicações computacionais*
- *Mostrar como a recursão é útil no desenvolvimento de aplicações que necessitam da mesma função várias vezes*

Roteiro

- *Objetivos*
- ***Recursão (Recursividade)***
- *Exercícios*

Recursão

- *Técnica de programação onde um método chama a si mesmo*
- *É um processo de resolução de um problema, reduzindo o problema em sub-problemas menores, com as seguintes características:*
 - *idênticos ao problema original*
 - *resolução mais simples e conhecida*

Recursão

- *Exemplo*

→ *Escrever um método que receba como parâmetro um inteiro n e retorne a soma de todos os números inteiros entre 0 e n*

→ $n = 4$

→ $soma = 0 + 1 + 2 + 3 + 4 = 10$

Recursão

- *Exemplo*

```
int Soma(int n)
{
    int i, soma=0;
    for (i=1; i<=n; i++)
        soma += i;
    return soma;
}
```

Recursão

- *Exemplo*

→ *Escrever um método que receba como parâmetro um inteiro n e retorne a soma de todos os números inteiros entre 0 e n . Não é permitido utilizar nenhum laço de repetição (for, while, do while, etc...)*

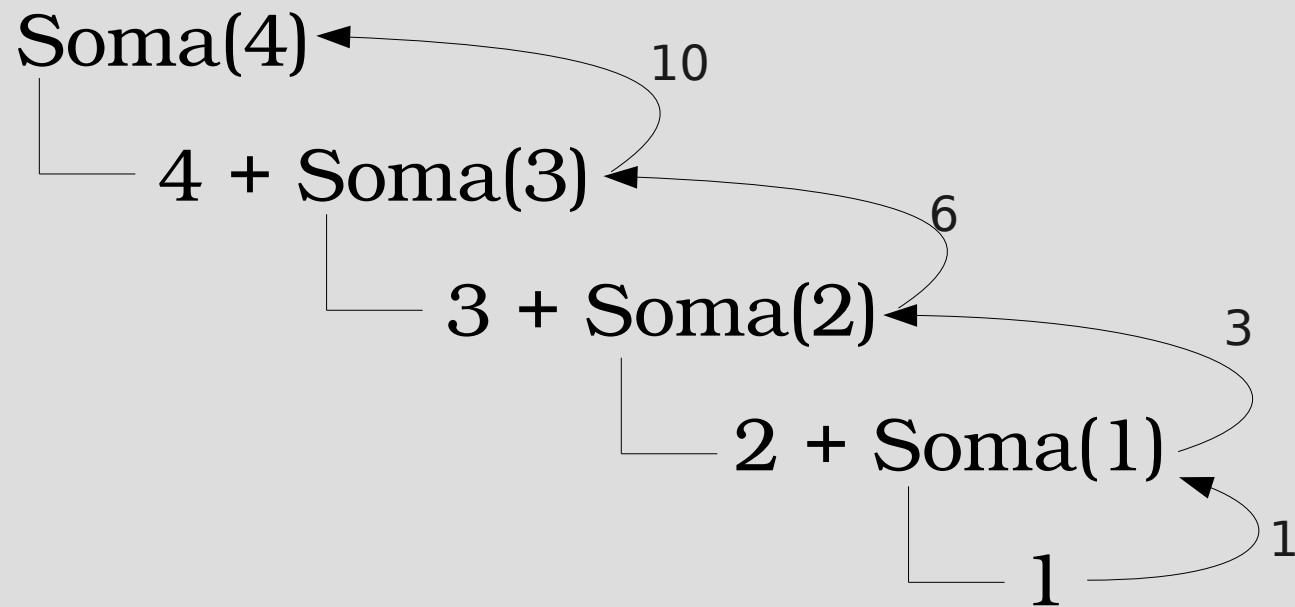
Recursão

- *Exemplo*

```
int Soma(int n)
{
    if (n == 1)
        return 1;
    else
        return (n + Soma(n-1));
}
```

Recursão

- *Exemplo*



Recursão

- *Anagramas*

- *Combinação qualquer de um conjunto de caracteres, utilizando caracteres de uma string fornecida*

- *Exemplo: lado*

- *Anagramas: aldo, alod, adlo, adol, aold, aodl*

- lado, laod, ldao, ldoa, load, loda*

- dlao, dloa, dalo, daol, doal, dola*

- olad, olda, oald, oadl, odla, odal*

Recursão

- *Anagramas*

- *Algoritmo*

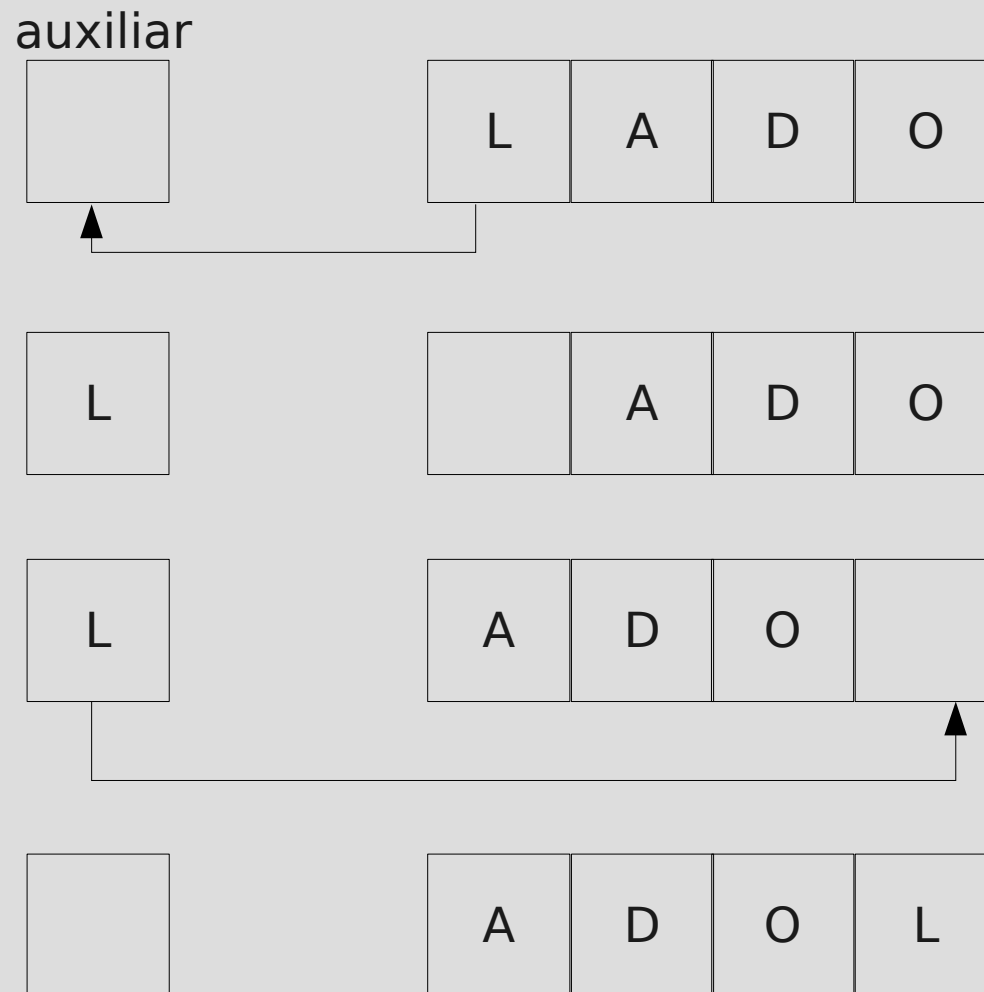
- Faça o anagrama das $n-1$ letras mais à direita*

- Gire todas as n letras*

- Repita essas etapas n vezes*

Recursão

- Anagramas*



Recursão

- *Anagramas*

- *Para realizar o anagrama das $n-1$ letras, devemos chamar o próprio método anagrama*

- *Cada vez que o método é chamado, ele trabalha com uma palavra com um caracter a menos*

- *Quando só existir uma letra, não há como gerar anagramas, portanto a mesma é retornada*

Recursão

- *Anagramas*

Anagrama (abc)

 Anagrama (bc)

 Anagrama (c)

 Anagrama (cb)

 Anagrama (b)

Anagrama (bca)

 Anagrama (ca)

 Anagrama (a)

 Anagrama (ac)

 Anagrama (c)

Anagrama (cab)

 Anagrama (ab)

 Anagrama (b)

 Anagrama (ba)

 Anagrama (a)

Recursão

- *Anagramas*

```
public static void GerarAnagrama(int novoTamanho){
    if (novoTamanho == 1)
        return;
    for (int i=0; i<novoTamanho; i++){
        GerarAnagrama(novoTamanho-1);
        if (novoTamanho == 2)
            Exibir();
        rotacionar(novoTamanho);
    }
}
```

Recursão

- *Anagramas*

```
public static void rotacionar(int novoTamanho){  
    int i;  
    int posicao = tamanho - novoTamanho;  
    char auxiliar = palavra[posicao];  
    for (i=posicao+1; i<tamanho; i++)  
        palavra[i-1]= palavra[i];  
    palavra[i-1] = auxiliar;  
}
```

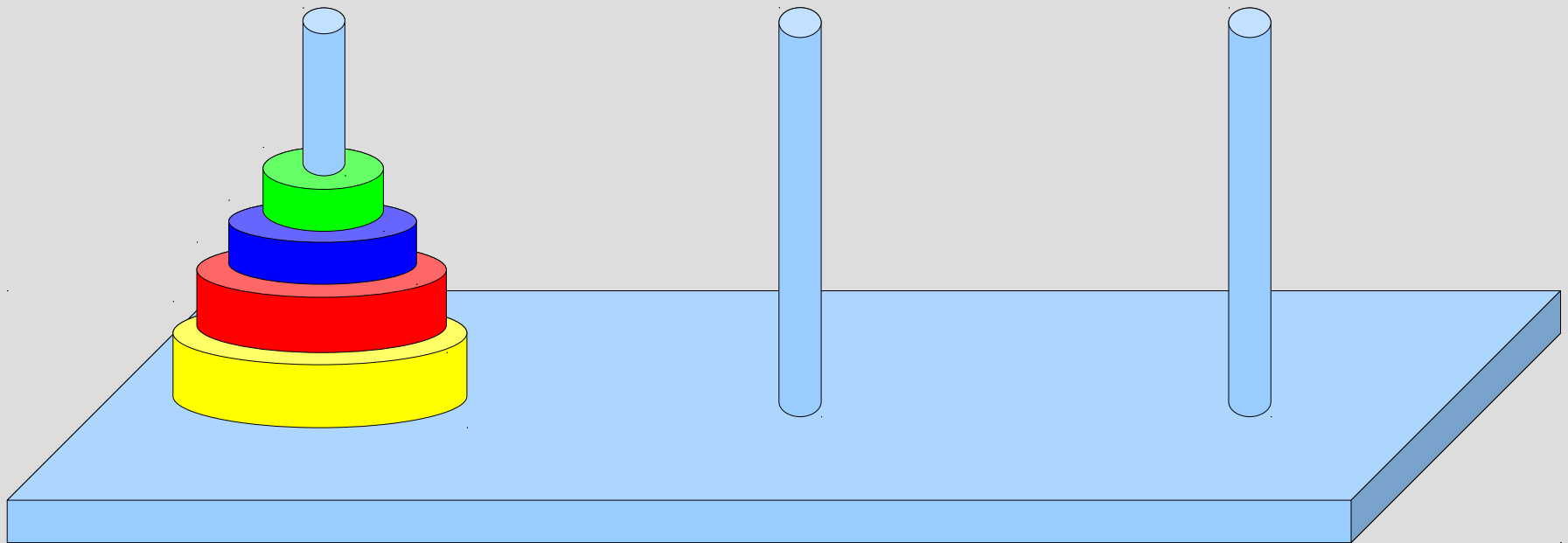
Recursão

- *Torres de Hanoi*

- *Quebra-cabeças que consiste de três hastes e diversos discos colocados nesta haste*
- *Os diâmetros dos discos são diferentes*
- *Todos os discos são iniciados na coluna A*
- *O objetivo é transportar todos os discos da coluna A para a coluna C, sendo que só um disco pode ser movido por vez e um disco só pode ser colocado em cima de um disco de diâmetro maior*

Recursão

- *Torres de Hanoi*



Recursão

- *Torres de Hanoi*

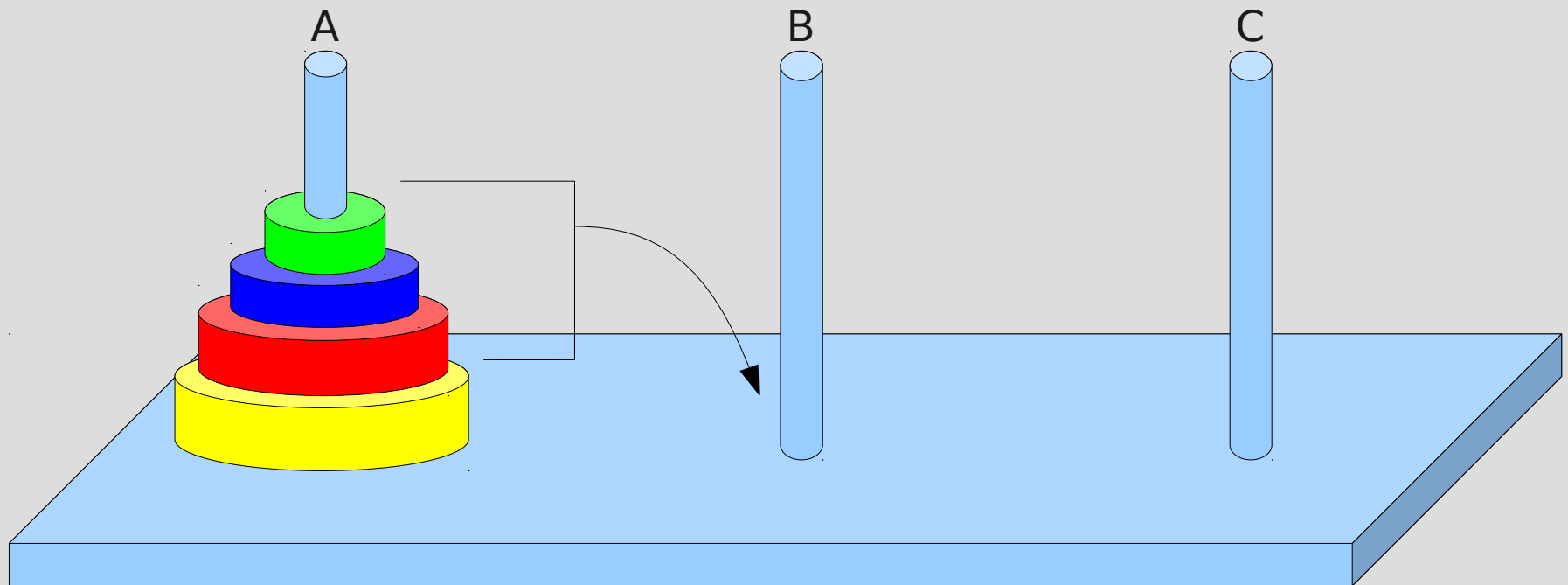
- *Mova a subárvore que consiste dos $n-1$ discos superiores de A para B*

- *Mova o disco maior (restou em A) de A para D*

- *Mova a subárvore de B para D*

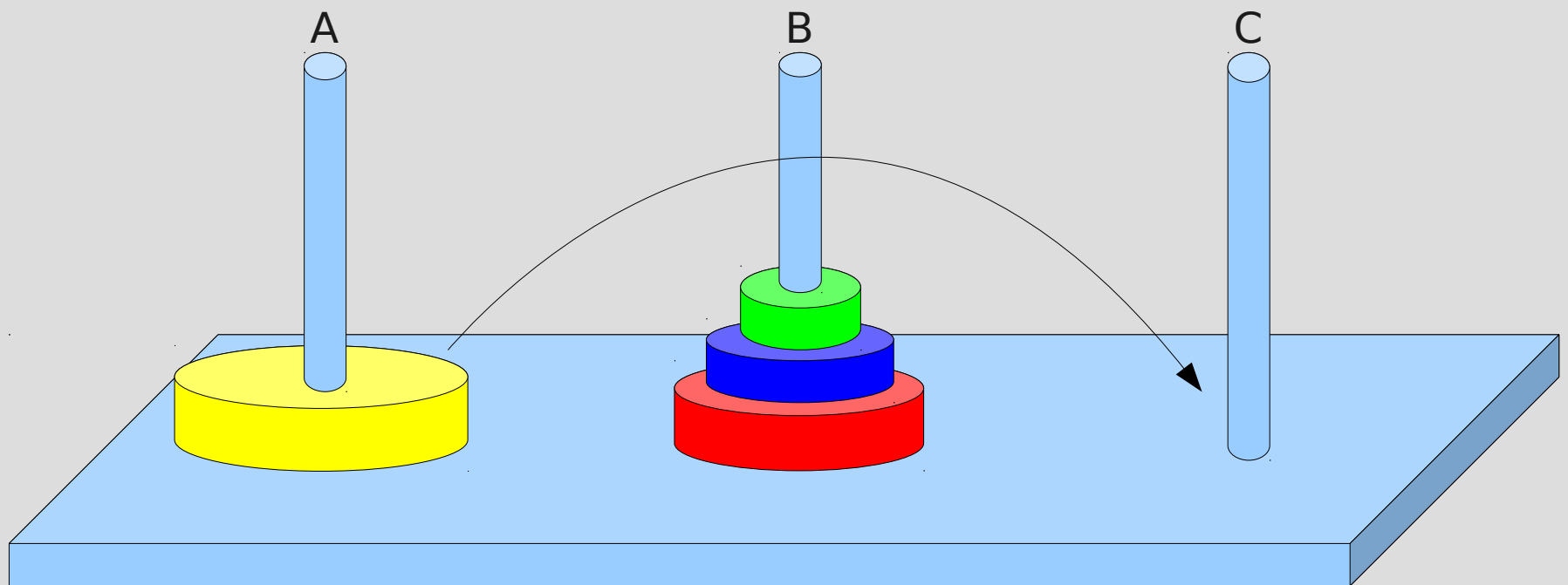
Recursão

- *Torres de Hanoi*



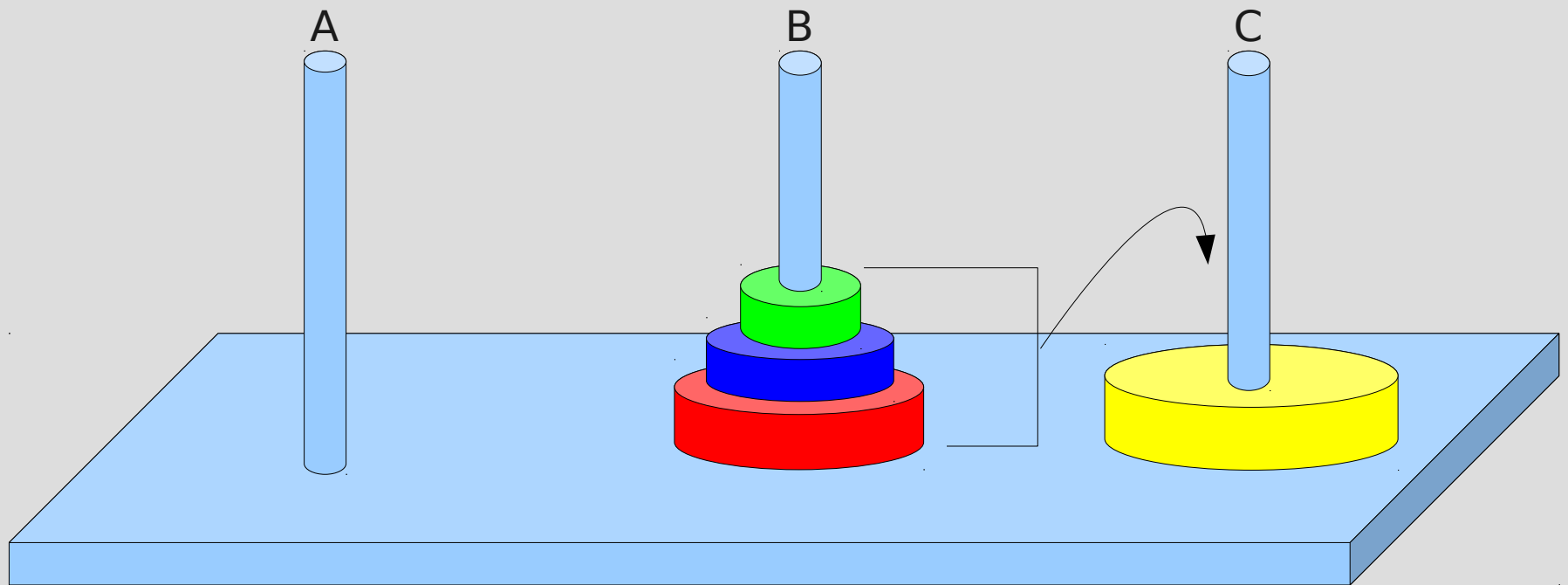
Recursão

- *Torres de Hanoi*



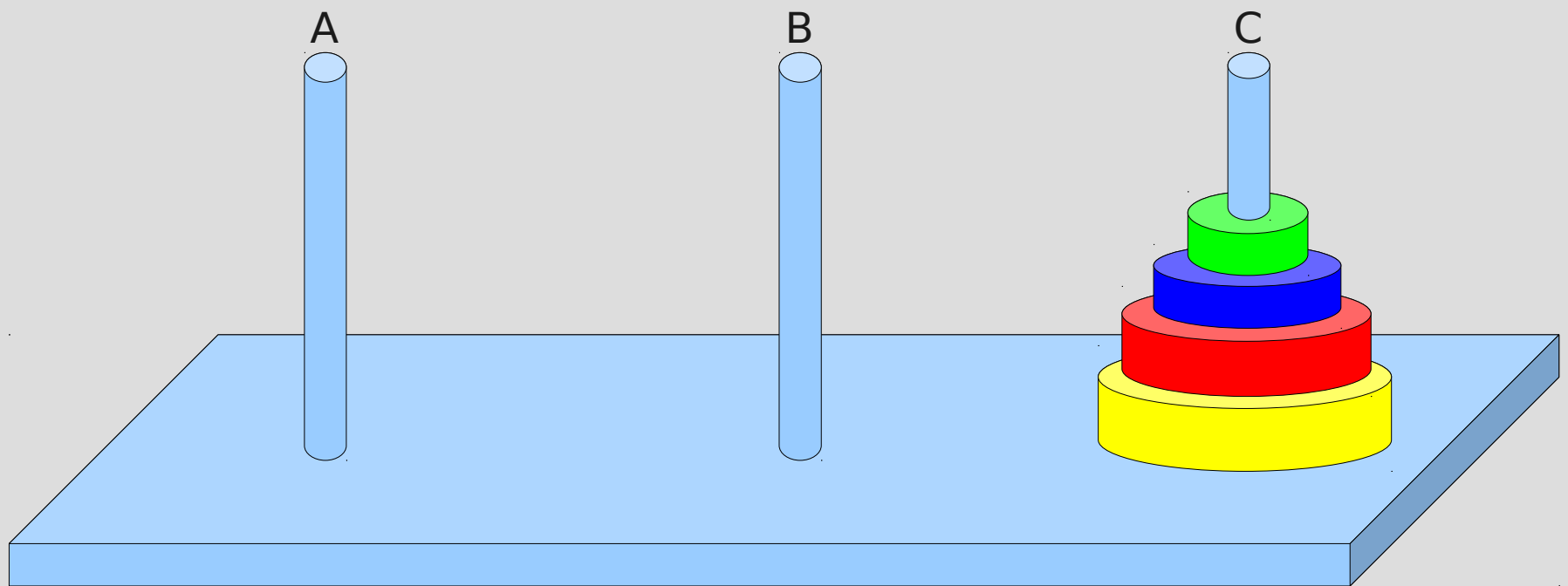
Recursão

- *Torres de Hanoi*



Recursão

- *Torres de Hanoi*



Recursão

- *Torres de Hanoi*

Método Hanoi (int num, char esq, char meio, char dir)
se num = 1 então
 Exibe “Disco 1 movido de esq para dir”
caso contrário
 Hanoi (num-1, A, C, B)
 Mover o disco do topo de A para B
 Hanoi (num-1, C, B, A)

Roteiro

- *Objetivos*
- *Recursão (Recursividade)*
- ***Exercícios***

Exercícios

- *Implementar um programa que calcula o fatorial de um inteiro n fornecido pelo usuário. O método fatorial deve ser recursivo.*
- *Implementar um programa que eleve um número n a uma potência m de forma recursiva.*
- *Implemente um programa recursivo para calcular o produto de dois inteiros, m e n , usando apenas adição e subtração.*
- *Implemente um programa para resolver o problema das Torres de Hanói, utilizando o algoritmo apresentado em aula.*