RECURSIVIDADE:

RECURSIVIDADE É UMA TÉCNICA DE DIVISÃO E CONQUISTA.

Os problemas originais serão divididos em problemas menores que tem as mesmas características do problema original.

Por conta disso, os subproblemas poderão ser novamente subdivididos em problemas ainda menores que continuam a manter as características do problema original.

Deve existir um ou mais subproblemas que não poderão ser divididos. Esse(s) subproblema(s) deve(m) ter a solução conhecida, pois essa solução será "juntada" às soluções dos demais subproblemas para se chegar à solução do problema original.

O(s) subproblema(s) acima é (são) denominado(s) soluções terminais ou triviais.

Sobre os problemas:

1) Podem ser divididos em subproblemas que são semelhantes ao problema original, mas com um domínio menor (menor quantidade de elementos, por exemplo)
2) Existe um ou mais subproblemas, com uma quantidade mínima ou tamanho mínimo que não necessite mais ser dividido pois se conhece a solução desse problema.

Sobre a implementação:

1) Defina sua estratégia de divisão dos problemas (como será dividido? Em quantos pedaços?)
2) Defina os testes que verificam se o problema é ou não um problema terminal ou trivial.
3) Construa uma rotina que recebe os dados do problema, verifique se esse problema é terminal. Se for terminal, retorne a resposta conhecida, caso contrário, aplique a sua estratégia de divisão (passo 1)

<u>Exemplo 1: Soma dos N primeiros números Naturais.</u>

1, 2, 3, 4, 5, ...

Soma(N) = 1 + 2 + 3 + ... + N-1 + N = Soma(N-1) + N , se N>1

Soma(5) = $\underline{1 + 2 + 3 + 4}$ + 5 = Soma(4) + 5 = 10 + 5 = **<u>15</u>**

Soma(4) = 1 + 2 + 3 + 4 = Soma(3) + 4 = 6 + 4 = 10

Soma(3) = 1 + 2 + 3 = Soma(2) + 3 = 3 + 3 = 6

Soma(2) = 1 + 2 = Soma(1) + 2 = 1 + 2 = 3

Soma(1) = 1

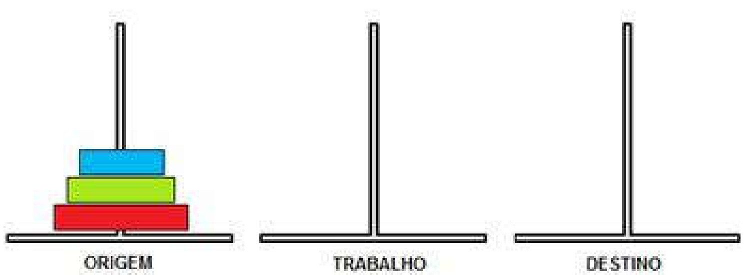
```
Inteiro Soma(N){
	Se (N > 1) retorna Soma(N-1) + N
	Senão retorna 1;
}
```

Exemplo 2: Conversão de números decimais para binário.

Exemplo 3: Torres de Hanói.

O objetivo do problema das Torres de Hanói é movimentar todos os anéis da torre origem para a torre destino, sob as seguintes restrições:

1) Pode-se movimentar apenas um anel de cada vez.
2) Não pode haver um anel com diâmetro menor sob um anel com diâmetro maior.

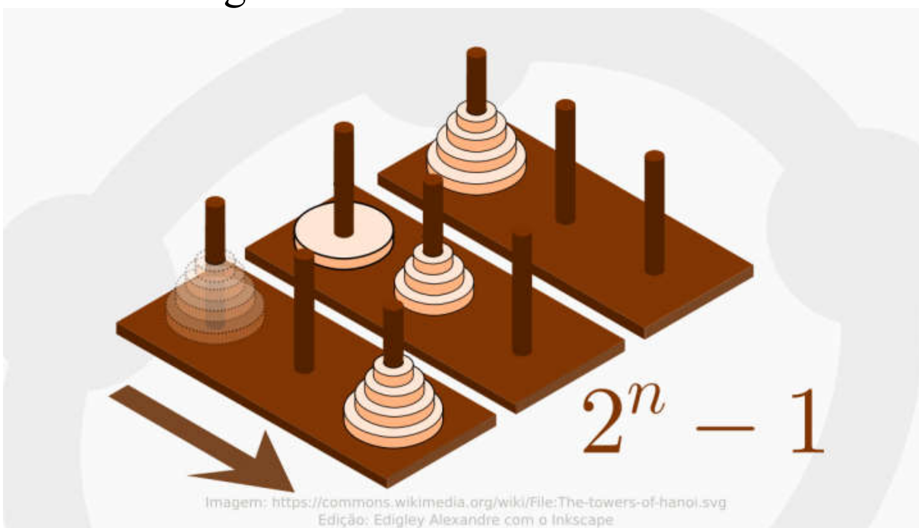


A torre trabalho é usada para ajudar a movimentação, respeitando as restrições.

Para modelar o problema, cada anel será associado a um número inteiro, sendo o menor deles o 1, o segundo menor o 2, até o maior que será associado ao valor N.

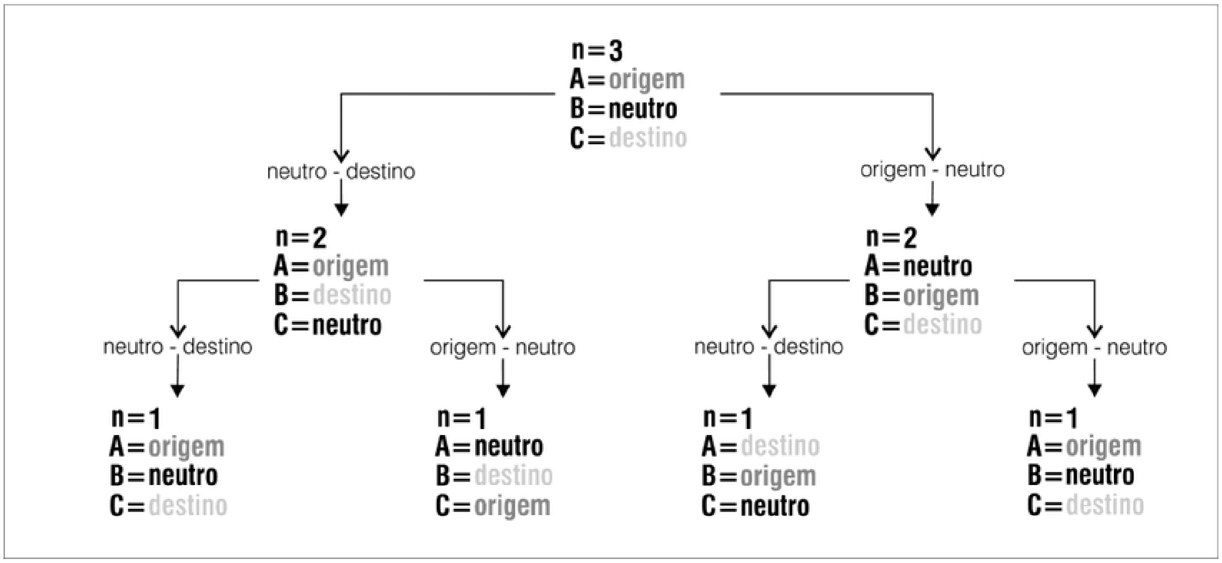
Para aplicar recursividade na solução do problema das Torres de Hanói, precisamos definir uma estratégia de divisão do problema original em problemas menores que tem a mesma característica do problema original.

Observe a figura abaixo:



Hanói(4, O, D, T)
    Hanói(3,O,T,D) -> Hanói(2,O,D,T) ->Hanói(1, O,T,D)
    Mover o anel 4 da Torre O para a Torre D
    Hanói(3,T,D,O) -....

```
Hanói(N,O,D,T)
    Se(N>1)
        Hanói(N-1, O, T, D);
        Mover anel N da Torre O para a Torre D
        Hanoi(N-1, T, D, O);
    Senão
        Mover anel 1 da Torre O para a Torre D
```



n (número de discos)	**número de movimentos**
1	1
2	3
3	7
4	15
5	31
6	63
8	256
10	1.023
15	32.767
20	1.048.575
30	1.073.741.823
64	18.446.744.073.709.551.615