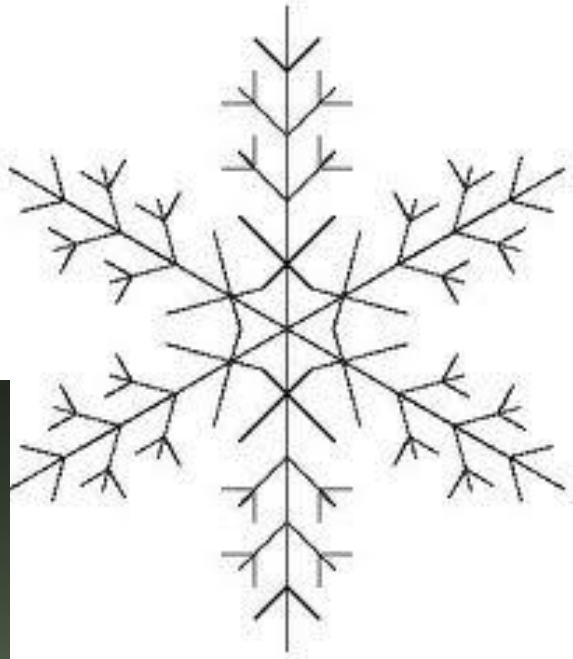
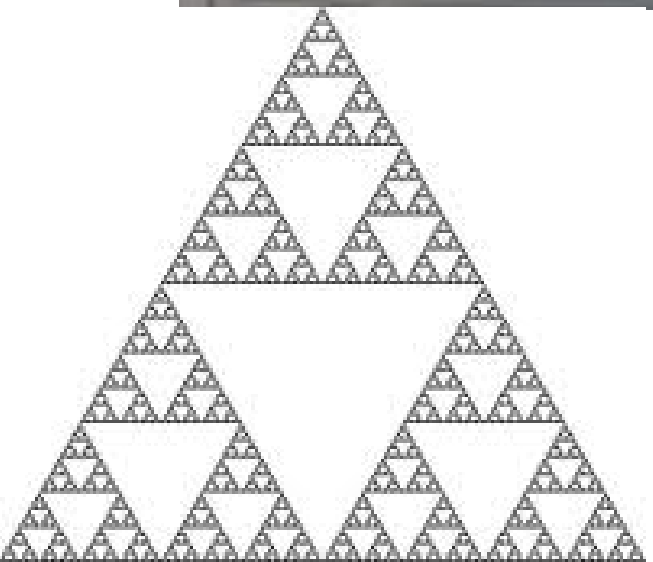
Java
10
Recursividade

Vitor Vaz da Silva





O definido entra na definição!

Factorial

- O factorial de n é o produto de n pelo factorial de n-1
- O factorial de 0 é 1

$$n! = n \times (n-1)!$$

$$0! = 1$$

Série de Fibonacci

- O enésimo elemento da série é o resultado da soma dos dois elementos anteriores.

$$F(n) = F(n-1) + F(n-2)$$

$$F(0)=0$$

$$F(1)=1$$

Factorial

```
long factorial(int n){
    long fact=1L;
    for( ; n>0; n--){
        fact*=n;
    }
    return(fact);
}
```

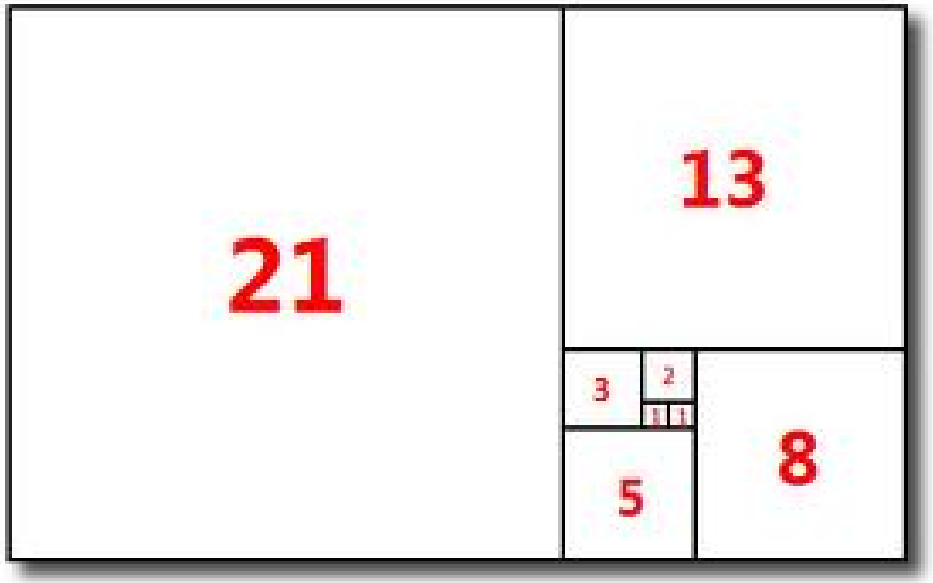
```
long factorial (int n){
  if(n==0) return(1L);
  return(n * factorial(n-1));
}
```

Série de Fibonacci

```
long fibonaci (int n){
     long fib=0L;
     if(n==0) return(fib);
     long fib_1=fib;
     fib=1L;
     if(n==1) return(fib);
     long fib_2=fib_1;
     fib_1=fib;
     for( ; n>1; n--){
          fib= fib_2 + fib_1;
          fib_2=fib_1;
          fib_1=fib;
     }
     return(fib);
}
```

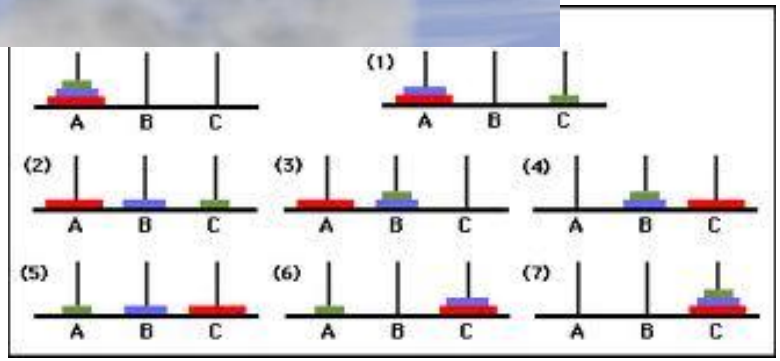
Série de Fibonacci

```
long fibonacci (int n){
    if(n==0) return(0L); //if(n<=1) return(n);
    if(n==1) return(1L);
    return(fibonacci(n-1)+fibonacci(n-2));
}
```



Torres de Hanoi

- Lenda da Torre de Brama



Torre de Hanoi

- 2^n-1 movimentos para n discos
- A peça mais pequena tem de ser colocada no local de destino total caso o número de peças a movimentar seja ímpar, ou para a outra caso seja par.
- Fazer os movimentos como se fossem apenas 3 peças, para depois libertar a que está por baixo.

Solução

- Torre com n discos (n>1)
- Movimento n-1 discos
- Movimentar último disco (a base, disco maior)
- Movimento os n-1 discos

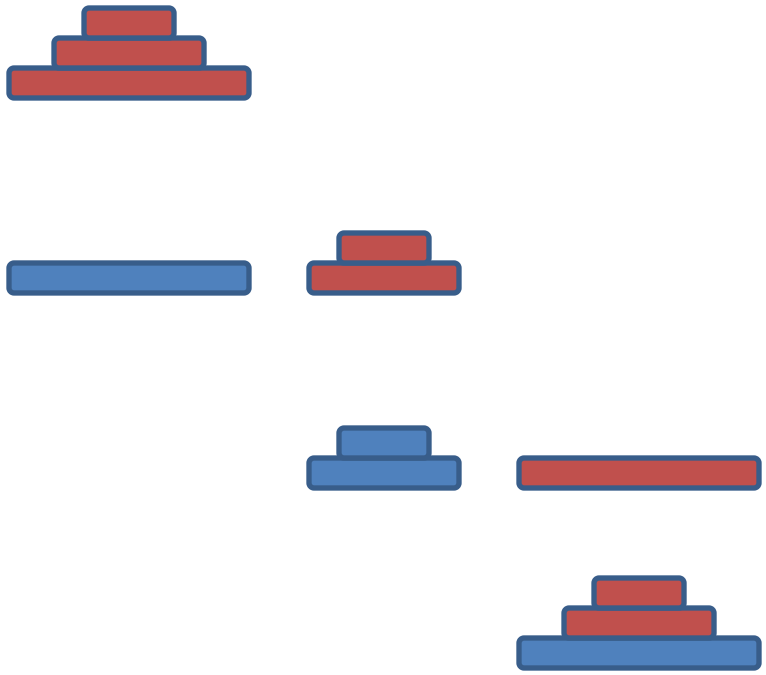
```
                                a       b       c
mudaDisco(num, pilha, livre, destino){

   if(num>0){
      mudaDisco(num-1, pilha, destino, livre);


      destino=pilha.primeiro;


      mudaDisco(num-1, livre, pilha, destino);
   }

}
```



Recursividade

- Utilizar quando:
  - O algoritmo fica mais compreensível
  - O algoritmo fica mais simples
  - Há recursos suficientes
- Efeitos
  - Utiliza mais contexto, stack
  - Pode demorar mais tempo

Exercícios – utilizar recursividade

- Calcular o maior número num array
- Preencher um array com números pares
- Somatório de uma série cujos elementos estão num array