



Prática 6 - Collatz

Para essa atividade prática, considera o algoritmo seguinte definido por números inteiros positivos:

1. Escolhe um número determinado
2. Se o número for par, divide-o por dois
3. Se o número for ímpar, multiplica-o por 3 e soma-lhe 1
4. Repetir todo o processo anterior até obter um 1

Por exemplo, se começarmos com o número 6 o processo é como o seguinte:

- 6, é par logo divide-se por 2 \Rightarrow 3
- 3, é ímpar logo multiplica-se por 3 e soma-se 1 \Rightarrow 10
- 10, é par logo divide-se por 2 \Rightarrow 5
- 5, é ímpar logo multiplica-se por 3 e soma-se 1 \Rightarrow 16
- 16, é par logo divide-se por 2 \Rightarrow 8
- 8, é par logo divide-se por 2 \Rightarrow 4
- 4, é par logo divide-se por 2 \Rightarrow 2
- 2, é par logo divide-se por 2 \Rightarrow 1

A conjectura (devida a Lothar Collatz, que foi o primeiro a estabelecer o problema em 1937) especifica que, independentemente do número com o que se inicia, acaba-se sempre com 1. A conjectura não pode ser formalmente confirmada, mas não se encontrou, até os dias de hoje, nenhum número que a invalide.



Dito isto, cria uma classe que inclua um método estático "collatz" com as seguintes características:

```
/**
 * Devolver a lista de Collatz para o valor n
 *
 * @param n valor inteiro positivo (>0)
 * @return Array de inteiros representando a lista de Collatz, os
valores restantes do array iniciam-se a zero
 *
 * Dado um número devolver um array com a série de números de Collatz
(incluindo na primeira posição o número
 * inicial. Para não ter de trabalhar com um array de tamanho variável
supor que o array tem um tamanho fixo
 * suficiente (p.ex. 300 elementos) para acomodar a série e que os
elementos não utilizados se fixem a zero.
 *
 * Por exemplo, para o número 6 iria devolver o seguinte array:
 * [6, 3, 10, 5, 16, 8, 4, 2, 1, 0, 0, 0, 0, ..., 0]
 */
public static int[] collatz(int n) { ... }
```