CSCI 3005 – SPRING 2019 – PROGRAMMING ASSIGNMENT 3 – Breadth First

Your task is to develop a solution to the *MinSteps* problem using the breadth-first algorithm to search through the various possible options in the state space graph. The rules of the game are the same as before: starting with a pile of n tokens, remove tokens from the pile until only one token is left, subject to the following:

- If the number of tokens in the pile is divisible by 3, you can remove two-thirds of the tokens
- If the number of tokens in the pile is divisible by 2, you can remove half of the tokens
- It is always possible to remove exactly one token per move

The objective of the game is to reduce the number of tokens to 1 in as few moves as possible. Using breadth-first, the start state of the problem consists of the initial number of tokens. The children of each state correspond to the possible configurations that result from applying the relevant rules. By applying the breadth-first algorithm studied in class, your solution should discover the optimal sequence of moves. You may need to modify the algorithm so that it stops as soon as the solution state (1 token) is found.

Your assignment is to write a Java class named **MinBFSteps** which provides public methods with the following signatures and functionality:

```
MinBFSteps(int n)        // initializes a game with n initial tokens
int solutionSteps()      // computes minimum number of steps using breadth-first search
int solutionNodes()      // returns total number of distinct states generated using breadth-first
String getBFMoves()      // returns sequence of moves required (see format below)
```

You should initially test your methods using the `MinBFStepsTest` class provided. Here is a sample sequence of method calls and return values:

```
MinBFSteps game = new MinBFSteps(8);
game.solutionSteps();          // returns 3
game.solutionNodes();          // returns 7 (see figure below)
game.getBFMoves();             // returns "8 --> 4 --> 2 --> 1"
```

Once you are satisfied that your solution works, the source code for **MinBFSteps.java** should be submitted to Mimir for testing. <u>Note that a solution that computes the results by any method other than the breadth-first algorithm will receive zero points for testing and design.</u>