Solver Class:

- Instance Variables:
  - Board
- Methods:
  - solveCurrentBoard

Board Class:

- Instance Variables:
  - 2D-Array of Pebbles
- Methods:
  - addPebble - add pebble to specified location on board

Pebble Class:

- Instance Variables:
  - Color
  - Iteration Value
  - Left Pebble
  - Right Pebble
  - Up Pebble
  - Down Pebble
- Methods:
  - flipColor - if black, set Color to white, else set Color to black

Algorithm 1 Solve Current Board: Belongs to Solver Class

1: **procedure** SOLVECURRENTBOARD
2: **Input:** A board representation that contains a 2 dimensional array of pebble representations.
3: **Output:** Number of iterations that pebble representations are replaced and whether or not there are any black pebble representations remaining.
4: $Q \leftarrow \emptyset$ (Queue)
5: *blackPebbles* $\leftarrow 0$
6: **for** each *pebble* on board **do**
7: **if** *pebble* is white **then**
8: add *pebble* to Q
9: set *pebble* iteration level to *0*
10: **else**
11: *blackPebbles* $\leftarrow$ *blackPebbles* $+ 1$
12: *currentIteration* $\leftarrow 0$
13: **while** Q is not empty **do**
14: *currentPebble* $\leftarrow$ pebble deqeued from Q
15: *currentIteration* $\leftarrow$ *currentPebble*'s *iterationLevel*
16: Change color of *currentPebble*'s black neighbors to white and enqueue them to Q with iteration level of one greater than currentPebble's iteration level.
17: Decrement *blackPebbles* by 1 for each black neighbor flipped
18: **return** *currentPebble*'s iteration level and True if blackPebbles > 0, False otherwise

Runntime:

- First loop iterates through each pebble at most once, this O(n)
- Second loop goes through entirety of queue, which holds at most all the pieces on the board since pieces cannot be added to the queue more than once, this is O(n)
- O(n) + O(n) = O(n), where n is the number of pieces on the board