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** add *pebble* to Q
8: **else***blackPebbles* $\leftarrow$ *blackPebbles* $+ 1$
9: T $\leftarrow$ result of preforming the flipPebbles procedure on Q and *blackPebbles*.
10: anyBlackRemaing $\leftarrow$ True if number of black remaining in T > 0, False otherwise
11: R $\leftarrow$ number of iterations in T and anyBlackRemaining
12: **return** R

Algorithm 2 Flip Pebbles: Belongs to Solver Class

1: **procedure** FLIPPEBBLES
2: **Input:** Queue of pebbles to flip, Q, and number of remaining black pebbles, *blackPebbles*.
3: **Output:** Number of iterations that pebble representations are replaced and the remaining number of black pebbles.
4: **if** Q is empty **then**
5: **return** 0 (number of iterations) and remaining number of black pebbles
6: $R \leftarrow \emptyset$ (Queue)
7: *remaining* $\leftarrow$ *blackPebbles*
8: **for** each pebble p in Q **do**
9: Change color of p's black neighbors to white and add them to R.
10: Decrement *remaining* by 1 for each black neighbor flipped
11: T $\leftarrow$ result of performing the flipPebbles procedure on R and *remaining*.
12: Add 1 to number of number of iterations in T
13: **return** T