

Task 1

0 **H H** H T T T

1 _ _ H T T **T H H**

2 _ _ **H T** T _ _ H **T H**

3 _ _ _ _ T H **T H** T H

Yellow shows the pair that is about to get moved. The Bold is what was just moved. It is solvable in 3 moves.

Task 2

0 _ _ H H **H T** T

1 **H T** **H H** _ _ T

2 H T _ _ **H H T**

3 H T **H T** H _ _

Yellow shows the pair that is about to get moved. The Bold is what was just moved. It is solvable in 3 moves.

Efficiency (this was done with the $n, n - 1$ test).

Because we're not looping through the iteration, we weren't too sure how to count the actions/iterations. We included a num_actions counter for all actions that were updating coin positions, deleting coin positions and making changes to the various indexes for our algorithm. Our n is based on the number of coins.

The n, n algorithm also does not loop through the array. It knows the exact indexes to move every time.

