

## Task 1

0            H H **H T** T T

1        **H T** **H H** \_ \_ T T

2       H T \_ \_ \_ \_ T **T H H**

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

4            \_ \_ 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 4 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

Because we're no longer 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.

