Matrix Solitaire

The key observation is that there is only one pair of tiles for each nominal value, so it is always best to remove it if it is free, regardless of other tiles. Hence, the problem lends itself to a greedy solution.

The algorithm is straightforward: as long as there are free identical pairs, remove them (in any order). There are various ways to implement it, the simplest of which is to use a queue.

Solution 1 — Queue

Let there be a single queue initially containing all free tiles and their respective position on the board, along with a flag indicating on which boundary it is located (left or right).

Then, as long as there are free tiles, extract one from the queue and do the following:

1. If it was not yet seen, save its position and flag into an indexed container.
2. Otherwise, match its pair and, for both tiles, add their neighboring tile (if there is one still unmatched, which will become free) from the same row to the queue.

Once the queue becomes empty, the answer will be positive if the number of matches equals the total number of pairs on the board; or negative otherwise.

Solution 2 — Deque

Let there be n double-ended queues, one for each row, initially containing all tiles in the respective row. Additionally, save the row indices where each tile appears.

Then, keep iterating over the queues and checking whether each end of the queue has a matching tile, using the row indices of the corresponding tile for quick lookup.

Once there are no more matches, the answer will be positive if the number of matches equals the total number of pairs on the board; or negative otherwise.