1. For exact solution, the DFS method is used. For each board and each red token, branch all possible moves of it. The DFS tree max height is bounded by the size of the greedy solution. The details are listed as code comments. A list of tuples is used to store the best and intermediate solutions.

1. We use a greedy solution to upper bound the search tree height. For the greedy solution, in each round, we find the blue and red token pairs that are closest to each other (use the spread action to cover some distance) and move the red token towards the blue one. For the test.csv, with greedy solution as upper bound, DFS solution terminates in 1s. Without it, the DFS does not finish in 1min.

3. With spawn action enabled, the complexity of DFS solution is drastically increased since possible moves are increased by the number of all empty cells. For greedy solution, considering spawn a token next to blue token with high power could be an interesting strategy.