Here is a game that gave inspiration for this project:
Move it! A sliding block puzzle game, by Optime Software LLC:
https://apps.apple.com/bg/app/move-it/id363162684

The game provides hints, but not solutions.

Algorithms that come to mind are breadth first search and depth first search. In a sense, this type of puzzle is a maze. The difficult thing though is that the walls are moving and the path is not clear. There can also be multiple solutions for some levels but just one solution for others.

Breadth first search and depth first search are common mapping algorithms for path finding. They are optimal, I just need to figure out how I would incorporate a similar algorithm into a solver.

The time complexity of both breadth and depth first search algorithms is O(V+E) where V is the vertices and E is the edges in the map. I am not dealing with a proper maze, however, so I am not 100% sure what the time complexity will be.