Figure 1: Shows points representing individual runs of the puzzle solver. This graph is comparing number of boards checked and time.

Issues of asymptotic complexity: The primary issue with this type of puzzle solver, is the fact that over a large number of moves, the number of searches can increase drastically. For each move along a puzzle that doesn’t lead to a solution, could lead to up to four more puzzles to be checked. So basically, when looking at board moves in a tree structure, each node could contain four branches. So, the complexity of this tree, in a worst-case scenario could be O(4^n-1).

Factors of runtime efficiency: Time to search for a previously seen board, O(1) in the hashmap, time to check for legal next moves O(n), time to make all of the moves O(n), time to search the tree O(4^n-1).

Benefits of including a previous position variable: Including a variable that eliminates the possibility of checking the previous position without checking the hashmap, cuts the time by an order of O(n), where n is the number of boards.