Approach

The best result was obtained after using the A* algorithm with the Manhattan Distance + Linear Conflict heuristic. We know that A* finds a solution with less number of nodes expanded as the heuristic function acts as a lower bound for the actual cost to the goal state. Apart from that there were many other optimizations done to increase the efficiency of the code. They are as follows:

1. State was represented by a string in the code to take advantage of the internal optimizations of string operations. The string was of length 16 and consisted of all the tiles merged together in Row-major order. All the operations like heuristic calculation and expanding nodes were done using the string only.
2. When calculating the possible moves, the move opposite (Left -> right) was not expanded as it would lead to the same state and won't be a part of the optimal solution.
3. The path followed to reach that state was also stored along with the state, so that the final path can be found out easily and it also helps in Point 2.
4. Instead of calculating the heuristic value for each state, the heuristic value was only calculated for the initial state and after that the change in heuristic value(delta in the code) was calculated based on each move.

Explanation for optimality of the approach

We know that A* algorithm is complete and optimal when the heuristic function is admissible and consistent. The heuristics used was Manhattan Distance(MD) + 2* Linear Conflict(LC) . MD is the path cost to a relaxed version of the puzzle where a tile can move to any adjacent tile. So as MD is a valid solution to a relaxed version of the puzzle, it is also a valid heuristic for the main puzzle as it always underestimate the real cost(admissible) and as it is a actual cost of a relaxed game, it follows triangle inequality,hence it is consistent. One thing Manhattan distance fails to account for is the LC between tiles. LC occurs when two tiles that are in the same row or column, and their goals are also in the same row or column, but they are in the wrong order. Let’s assume 4576 is present in the second row, as per manhattan distance, tiles 7 and 6 are just 1 move away from their actual positions. But it is impossible to swap these two tiles in just 2 moves. Each LC takes at least 2 additional moves than the manhattan distance to reach their goal. So the heuristic MD + 2*LC is also admissible as it underestimates the actual cost.

Approach	Test case 1		Test case 2		Test case 3		Test Case 4	
	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes
Manhattan	0.001 s	32	1.0878 s	111182	14.7387 s	1363848	9.1654 s	890864
Manhattan optimised	0.0004 s	32	0.5251 s	111182	7.9653 s	1363848	4.9107 s	890864
Manhattan with Linear Conflict	0.0017 s	32	0.8203 s	21162	5.3972 s	139933	7.2048 s	186238
Manhattan with LC optimised	0.0003 s	32	0.1285 s	21162	0.9366 s	139933	1.2525 s	186238

Manhattan : Manhattan distance used as heuristic with heuristic calculation done at every state

Manhattan Optimised: Manhattan distance used as heuristic with heuristic calculation done using delta from previous state

Manhattan with Linear Conflict: Manhattan distance + 2* LC used as heuristic with heuristic calculation done at every state

Manhattan with LC optimised: Manhattan distance + 2*LC used as heuristic with heuristic calculation done using delta from previous state