1 2 3 g = 3
_ 4 _ h = 3
7 6 5 f = 6

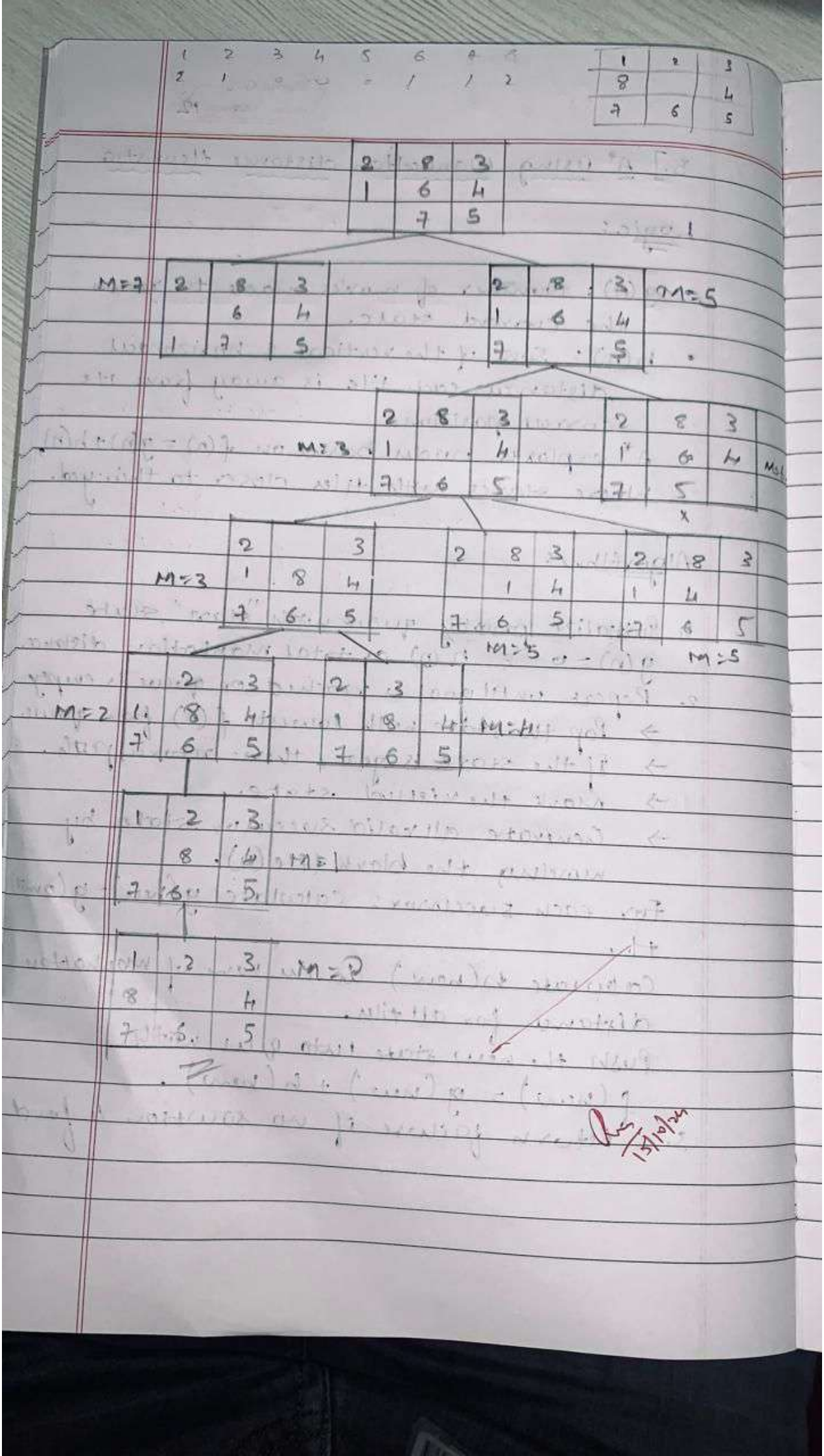
B.] A* using Manhattan distance Heuristic

Logic:

- g(n): Number of moves made to reach the current state.
- h(n): Sum of the vertical & horizontal distances each tile is away from its correct position.
- A* explores nodes based on f(n) = g(n) + h(n), where states with tiles closer to their goal.

Algorithm:

1. Initialize priority queue with "start" state. g(n) = 0 & h(n) as total Manhattan distance.
2. Repeat until goal is reached or queue is empty
   → Pop the state with lowest f(n) from queue.
   → If the state is goal, then return path.
   → Mark the visited state.
   → Generate all valid successor states by moving the blank tile (0).
   For each successor: calculate g(new) = g(omit) +1.
   Calculate h(new) as the sum of Manhattan distances for all tiles.
   Push the new state into queue with,
   f(new) = g(new) + h(new).
3. Return failure if no solution is found.


M=7
M=5
M=3
M=3
M=5
M=5
M=2
M=4
M=1
M=0