

3) Solve the 8 puzzle problem using

- (a) Depth First Search
- (b) Manhattan Distance

Algorithm for the 8 puzzle problem using Depth First Search
& Manhattan Distances

Step 1: Initialize the start and goal states:

Refine the initial puzzle configuration and two goal configuration.

Step 2: Find the blank/empty tile:

Locate the position of the blank/empty tile.

Rough work:

8	2	6
3		7
1	4	5

Blank tile is in Number of moves
middle : 4
edge : 3
corner : 2

Step 3: Check if the current state is the goal state:

Comparison of the current puzzle state with the goal configuration. If they match, the puzzle is solved.

Step 4: Explore possible moves as shown in the rough work
(Right, down, left, up)

Step 5: Recursively Explore the new states:

DFS:

- Explore each branch before backtracking
- Generates moves by sliding tiles and explores possible moves
- Manhattan Distance:
 - Heuristic: calculates the sum of absolute differences between the current positions of the tile versus the final position tile

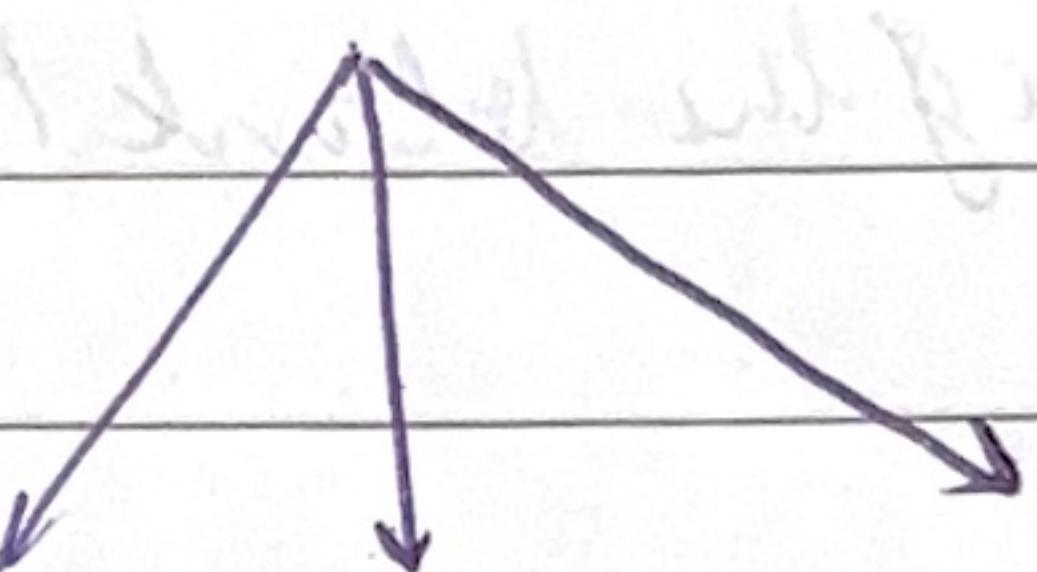
Step 6: Backtracking if and only if no solution is found

Step 7: End when the goal state is reached or all possibilities are exhausted.

Example:

Depth First Search

5	8	3
0	2	1
7	6	4



0	8	3
5	2	1
7	6	4

5	8	3
7	2	1
0	6	4

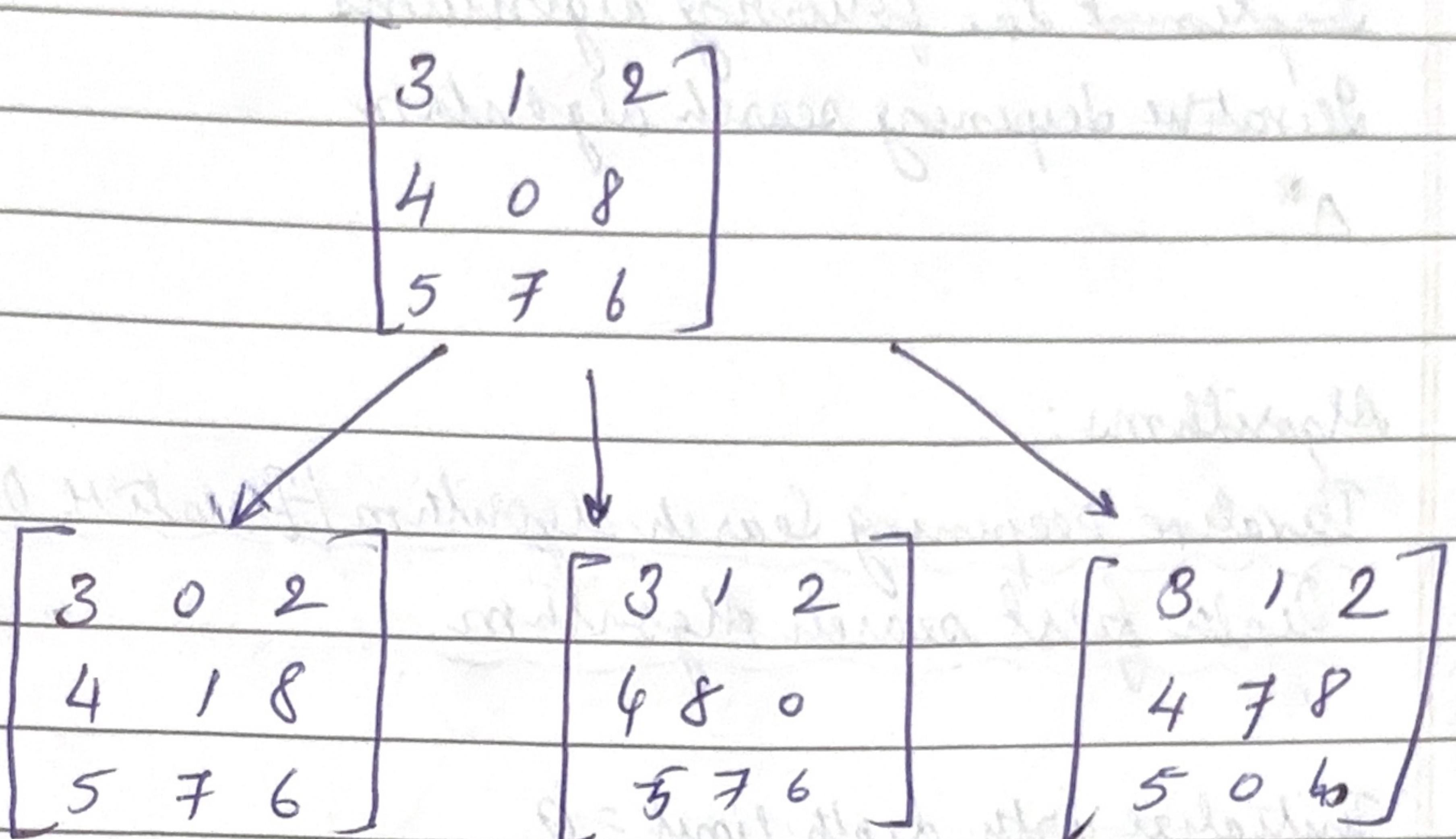
5	8	3
2	0	1
7	6	4

8	0	3
5	2	1
7	6	4

5	8	3
0	2	1
7	6	4

All possible cases are checked and verified.

Manhattan Distance



~~Some P/B~~