```
1BM22CS335_Yashraj Sinha_Vacuum Cleaner
Starting Vacuum Cleaner Simulation
Current Grid State:
0 1
0 0


Cell at (1, 1) is already clean.
Vacuum moved left to position (1, 0)
Current Grid State:
0 1
0 0


Cell at (1, 0) is already clean.
Vacuum moved right to position (1, 1)
Current Grid State:
0 1
0 0


Cell at (1, 1) is already clean.
Vacuum moved up to position (0, 1)
Current Grid State:
0 1
0 0


Cleaned cell at position (0, 1)
Vacuum moved down to position (1, 1)
Current Grid State:
0 0
0 0


All cells are clean! Total moves taken: 4


...Program finished with exit code 0
Press ENTER to exit console.
```

```
1BM22CS335_Yashraj Sinha_8-puzzle_bfs_dfs

Solving 8-puzzle using BFS:
Step 0: [[1, 2, 3], [4, 5, 6], [0, 7, 8]]
Step 1: [[1, 2, 3], [0, 5, 6], [4, 7, 8]]
Step 2: [[1, 2, 3], [4, 5, 6], [7, 0, 8]]
Step 3: [[0, 2, 3], [1, 5, 6], [4, 7, 8]]
Step 4: [[1, 2, 3], [5, 0, 6], [4, 7, 8]]
Step 5: [[1, 2, 3], [4, 0, 6], [7, 5, 8]]
Step 6: [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
Solution found using BFS!

Solving 8-puzzle using DFS:
Step 0: [[1, 2, 3], [4, 5, 6], [0, 7, 8]]
Step 1: [[1, 2, 3], [0, 5, 6], [4, 7, 8]]
Step 2: [[0, 2, 3], [1, 5, 6], [4, 7, 8]]
Step 3: [[2, 0, 3], [1, 5, 6], [4, 7, 8]]
Step 4: [[2, 5, 3], [1, 0, 6], [4, 7, 8]]
Step 5: [[2, 5, 3], [1, 7, 6], [4, 0, 8]]
Step 6: [[2, 5, 3], [1, 7, 6], [0, 4, 8]]
Step 7: [[2, 5, 3], [0, 7, 6], [1, 4, 8]]
Step 8: [[0, 5, 3], [2, 7, 6], [1, 4, 8]]
Step 9: [[5, 0, 3], [2, 7, 6], [1, 4, 8]]
Step 10: [[5, 7, 3], [2, 0, 6], [1, 4, 8]]
Step 11: [[5, 7, 3], [2, 4, 6], [1, 0, 8]]
Step 12: [[5, 7, 3], [2, 4, 6], [0, 1, 8]]
Step 13: [[5, 7, 3], [0, 4, 6], [2, 1, 8]]
Step 14: [[0, 7, 3], [5, 4, 6], [2, 1, 8]]
Step 15: [[7, 0, 3], [5, 4, 6], [2, 1, 8]]
Step 16: [[7, 4, 3], [5, 0, 6], [2, 1, 8]]
Step 17: [[7, 4, 3], [5, 1, 6], [2, 0, 8]]
Step 18: [[7, 4, 3], [5, 1, 6], [0, 2, 8]]
Step 19: [[7, 4, 3], [0, 1, 6], [5, 2, 8]]
Step 20: [[0, 4, 3], [7, 1, 6], [5, 2, 8]]
Step 21: [[4, 0, 3], [7, 1, 6], [5, 2, 8]]
Step 22: [[4, 1, 3], [7, 0, 6], [5, 2, 8]]
Step 23: [[4, 1, 3], [7, 2, 6], [5, 0, 8]]
Step 24: [[4, 1, 3], [7, 2, 6], [0, 5, 8]]
Step 25: [[4, 1, 3], [0, 2, 6], [7, 5, 8]]
Step 26: [[0, 1, 3], [4, 2, 6], [7, 5, 8]]
Step 27: [[1, 0, 3], [4, 2, 6], [7, 5, 8]]
Step 28: [[1, 2, 3], [4, 0, 6], [7, 5, 8]]
Step 29: [[1, 2, 3], [4, 5, 6], [7, 0, 8]]
Step 30: [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
Solution found using DFS!


...Program finished with exit code 0
Press ENTER to exit console.
```

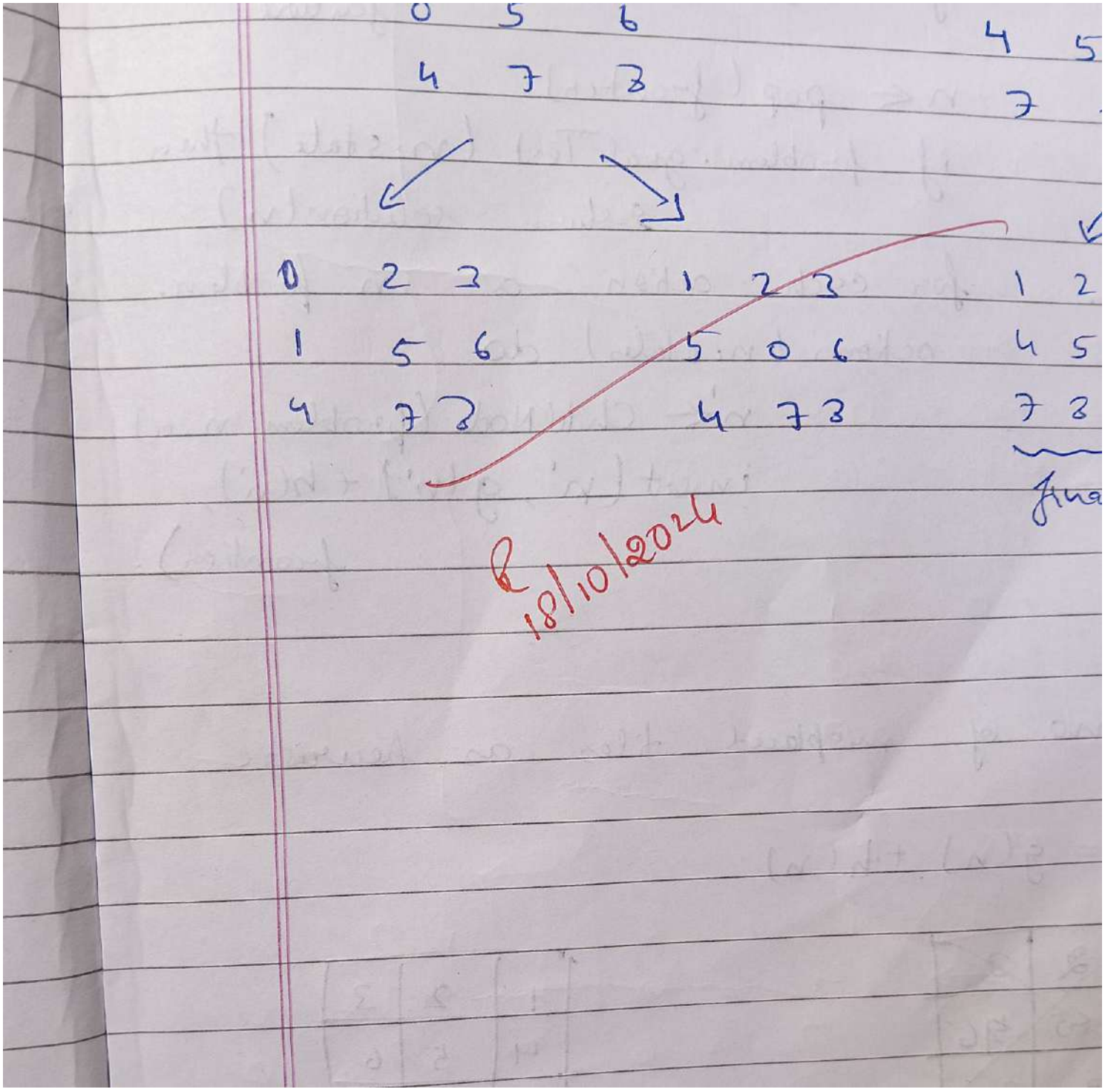
left, right), to see
any are dirty.

ii. Move to the next
applying strategy
BFS, DFS or other

d. If no dirty cell ar
stop.

3) End.

18/10/2024


0 5 6
4 7 3
4 5
7
0 2 3
1 5 6
4 7 3
1 2 3
5 0 6
4 7 3
1 2
4 5
7 3
18/10/2024