26 Puzzles Problem

Prerequisites

- Python must be installed on your system. The program is tested with Python 3.x.
- No additional libraries are required outside of the Python Standard Library.

Program Files

- `main.py`: The main Python script containing the source code of the puzzle solver.
- `Input1.txt`, `Input2.txt`, `Input3.txt`: Input files containing the puzzle configurations.

Running the Program

1. Place the `main.py` script and the input files in the same directory.
2. Open a terminal or command line interface.
3. Change the directory to where the script and input files are located with `cd path/to/directory`.
4. Execute the script by running the command: `python main.py`.

Running custom files

1. To run your own puzzles, put them in the file and name it as you like.
2. Demonstration of the script is presented in the end of the source code in the block:

```
if __name__ == "__main__":
    # Driver code that solves puzzles from files and writes solutions to new files.
    res = solve("Input1.txt")
    PuzzleFileIO.write_solution_to_file("Input1_solution.txt", res)
    res2 = solve("Input2.txt")
    PuzzleFileIO.write_solution_to_file("Input2_solution.txt", res2)
    res3 = solve("Input3.txt")
    PuzzleFileIO.write_solution_to_file("Input3_solution.txt", res3)
```

3. I encourage you to use provided `Solution` and `PuzzleFileIO` files that are created to improve user experience.

    a. `Solution` class has implemented `__repr__` method which allows us to print the solution without saving it to the file using `print(res)` command, where `res` is the returned value of `solve(str)` function.

4. `solve(str)` is the only function in the code that is created to combine all the classes and return the instance of solution class that can be accessed and all the information extracted.

Expected Output

- The program will read the puzzle configurations from the input files.
- It will process each puzzle using the A* search algorithm to find the solution.
- The solution for each puzzle will be written to `Input1_solution.txt`, `Input2_solution.txt`, and `Input3_solution.txt` in the same directory as the script.
- Each solution file will contain **ONLY** and **EXACTLY** information required by the task.

Solution Files

- `Input1_solution.txt`
- `Input2_solution.txt`
- `Input3_solution.txt`

These files will be created or overwritten in the same directory as the script when the program is run with the default settings.

Source Code

Please refer to the `main.py` file for the full source code. The code is well-commented for clarity and understanding of the logic used. Each class and methods have pydoc comments that explain their purpose.

Inputs and Outputs to Provided Problems

Input

```
1 2 3
4 0 5
6 7 8

9 10 11
12 13 14
15 16 17

18 19 20
21 22 23
24 25 26

1 2 3
4 13 5
6 7 8

9 10 11
15 12 14
24 16 17

18 19 20
21 0 23
25 22 26
```

Output

```
1 2 3
4 0 5
6 7 8

9 10 11
12 13 14
15 16 17

18 19 20
21 22 23
24 25 26

1 2 3
4 13 5
6 7 8

9 10 11
15 12 14
24 16 17

18 19 20
21 0 23
25 22 26

6
30
D W S D E N
8 7 8 9 8 7 6
```

Input

```
1 2 3
4 0 5
6 7 8

9 10 11
12 13 14
15 16 17

18 19 20
21 22 23
24 25 26

1 10 2
4 5 3
6 7 8

9 13 11
21 12 14
15 16 17

18 0 20
24 19 22
25 26 23
```

Output

```
1 2 3
4 0 5
6 7 8

9 10 11
12 13 14
15 16 17

18 19 20
21 22 23
24 25 26

1 10 2
4 5 3
6 7 8

9 13 11
21 12 14
15 16 17

18 0 20
24 19 22
25 26 23

13
77
E N W D S W D S E E N W N
16 17 16 15 14 15 16 15 16 15 16 15 14 13
```

Input

```
1 2 3
4 0 5
6 7 8

9 10 11
12 13 14
15 16 17

18 19 20
21 22 23
24 25 26

0 2 3
1 7 14
6 8 5

12 9 10
4 13 11
21 16 17

18 19 20
22 25 23
15 24 26
```

Output

```
1 2 3
4 0 5
6 7 8

9 10 11
12 13 14
15 16 17

18 19 20
21 22 23
24 25 26

0 2 3
1 7 14
6 8 5

12 9 10
4 13 11
21 16 17

18 19 20
22 25 23
15 24 26

16
135
S E N D N W W S D E S W U N U N
18 19 20 19 20 19 18 17 18 19 20 21 20 19 18 17 16
```