Solution Summary

Approach:

In Assignment 1.0 you will clean up the Wumpus cave with a vacuum cleaner robot. This assignment is composed of two different tasks. The first one is to go through the instructions whether it cleaned the cave completely or not. The other is to discover the directions to clean the cave. For the first one we need to follow a coordinated based approach and for the second one we must work upon the BFS approach.

Key Insights:

- The first solution was based on static set of instructions for cleaning a grid world (Wumpus cave) before that they did have search-based channel support (an example of a set of instructions are the coordinate-based approaches). Part of this is because the cave (map) gets trickier once you have portals and gates to account for.
- The second solution makes use of Breadth-First Search (BFS) among the searching algorithms to get the best path or instruction to clean the Wumpus's cave. It will also come across some issues with portals and spawn sites. Once starting positions have been found, BFS is applied to obtain the optimal solutions. BFS follows that to operate smoothly all the cleaned places would be visited.
- Several algorithms, including DFS, A*, and Dijkstra, were used to determine the best path. BFS provides robustness, efficiency, and adaptability while following closely to the objectives and restrictions of the robotic cleaning operation
- DFS, A*, Dijkstra algorithm cannot solve all problems due to its completeness and optimality.
- Optimizing data structures, such as priority queues in BFS or memorizing techniques, can enhance performance by reducing unnecessary calculations.

Design Decisions:

- Both solutions for locating the beginning of the map utilize different mechanisms. The check_plan_a_c.py iterates through the grid to locate an 'S', but if an 'S' cannot be found in the map, the find_instruction_d_f.py selects an empty space at random.
- The check_plan_a_c.py has a pre-defined set of instructions/moves while find_instruction_d_f.py uses BFS to identify path for the cleaning the map. As BFS follows the above steps it will give you an optimal solution to clean up everything.
- If a portal is detected (check_plan_a_c.py), it will redirect the movement to another portal site. The find_instruction_d_f.py script looks at portals and regards them as valid places to explore, consequently ports are incorporated in the BFS traversal.
- Output file: the solutions to the problems to run are saved on output files. In the first task, feedback is given to whether the provided plan is of very high quality to clear all the empty space, with support files. Overall, the second task has an optimum path which is the solution file of cleaning the map with various difficulties.

Conclusion:

These methodologies are shown to generalize well across different maps and handle the cleaning efficiently provided the requirement. Although the investigated technique with the coordinates is simple, BFS based strategy allows to improve optimality, enhance adaptability as well as make easier to implement more robust system with portals. In general, the solutions present a range of robotic cleaning approaches in grid-based applications.