The code has four major section:

1. Checking plans for known start

   For this problem, a plan has been provided. Following the simulation of the vacuum cleaner's movement, the cleaned squares are enumerated. Subsequently, the missed squares are identified. If all the empty spaces have been cleaned, the plan is deemed satisfactory; otherwise, it is considered unsatisfactory, and the missed areas are listed.

2. Checking plans for unknown start

   For this problem, as there is no designated starting point, each empty square is considered a potential starting point. Therefore, we apply the previously mentioned solution to all empty squares. The final plan is then determined by aggregating all the uncleaned spaces.

3. Finding plans for known start

   For this problem, both the cave map and the starting point are provided. By employing a recursive depth-first search (DFS) algorithm, each step of the plan is systematically calculated.

4. Finding plans for unknown start

   For this problem, just the cave is provided and each empty square is considered a potential starting point. Therefore, the depth-first search on the cave map, creating a plan that covers all reachable squares. The dfs function is a recursive function that generates the cleaning plan as it explores the cave. The opposite_direction function is used to move back to the previous position in the cave after exploring a direction. This approach assumes that there is at least one empty cell in the cave to start the DFS. If the cave is entirely filled with walls, the function will return an empty plan. Additionally, this plan is designed to work from any starting position, as it ensures all reachable squares are cleaned.