Task 1 - Author Adithyakrishna Venkatesh Hanasoge

1. The `ThiefsAndCops` class is defined, which represents the problem-solving scenario. It takes three parameters during initialization: `grid`, `orientations`, and `fov`. These parameters represent the grid layout, the orientations of the policemen, and their field of view, respectively.

2. The `_make_grid` method is a private method that initializes the `gridX` and `gridY` matrices for further processing. It creates grid points corresponding to each cell in the input grid, considering the top left, top right, bottom left, and bottom right corners. It also handles the number of policemen.

3. The `findClosestSafeGrid` method is used to find the closest safe grid for the thief and determine the policemen who can see the thief. It performs the following steps:

    - Determines the thief's position.
    - Constructs matrices representing the row and column indices of each policeman's position.
    - Calculates the angles between each grid point and the policemen's positions.
    - Defines the lower and upper limits of the field of view for each policeman.
    - Identifies the visible grids based on the calculated angles and field of view limits.
    - Identifies the policemen who can see the thief based on the intersection of the visible grids and the thief's position.
    - Modifies a copy of the original grid to mark the visible and safe grids, as well as the closest safe grid.
    - Retrieves the safe grids and finds the closest safe grid using the Manhattan distance.
    - Updates the new grid to mark the closest safe grid and the thief's position.
    - Returns the list of policemen who can see the thief and the coordinates of the closest safe grid.

4. The `visualize` method is used to visualize the grid and the results. It uses matplotlib to plot the grid, thief, policemen, visible grids, safe grids, and the closest safe grid.

5. The `_start_timer` and `_stop_timer` methods are used to measure the execution time of the `findClosestSafeGrid` method.

6. The `main` function demonstrates the usage of the `ThiefsAndCops` class. It defines a grid, orientations, and field of view, creates an instance of the class, calls the `findClosestSafeGrid` method to solve the problem, visualizes the grid using the `visualize` method, and prints the execution time, the policemen who can see the thief, and the coordinates of the closest safe grid.

Overall, this code solves the problem of finding the closest safe grid for a thief while considering the orientations and field of view of the policemen. It provides a visualization of the grid and the results for better understanding and analysis.

The code has been vectorised using Numpy without the use of Python "for loops".

A separate README for Task 1 is included in the Task_1 folder for further implementation details and how to run the code.