# Task: Solving Mazes Using the Dijkstra Algorithm

Objective: Implement the Dijkstra algorithm in Python to find the shortest path through a maze represented by a matrix of pixels.

You will be provided with two maze images and code to open and display them.

Your task is to implement the Dijkstra algorithm and then apply it to find the shortest path from the maze's start point to its end point.

Instructions:

1. Understanding the Maze:

* You will be given two maze images in a file format that can be opened using the provided code.
* Study the maze images and identify the start point and the end point.
* The maze is represented by a grid of pixels, where each pixel can be either black (representing a wall) or white (representing an open path).

2. Converting the Maze Image to a Matrix:

* Use the provided code to open the maze image files and display them, and to convert the maze image into a matrix representation, where each element in the matrix corresponds to a pixel in the maze.

3. Implementing the Dijkstra Algorithm:

* Implement the Dijkstra algorithm in Python to find the shortest path from the start point to the end point in the matrix representation of the maze.
* Utilize the **heapq** library in Python, specifically the heap data structure, to optimize the Dijkstra algorithm implementation.
* Remember to consider the walls (matrix elements with that smaller than 255) as obstacles that cannot be crossed.

4. Displaying the Path:

* The code to display the path you choose in your algorithm is provided to you.
* After finding the shortest path using the Dijkstra algorithm, apply the provided code to display the maze image with the path highlighted.