

如何运行

1. **环境准备:** 确保你的 Python 环境安装了必要的库:

```
pip install matplotlib numpy
```

2. **准备输入文件:** 创建一个文本文件 (例如 `maze_input.txt`) , 按照以下格式描述迷宫:

- 第一行包含两个整数 N 和 M , 分别代表迷宫的行数和列数。
- 接下来 N 行, 每行包含 M 个整数 (0 或 1) , 用空格分隔。0 代表通路, 1 代表障碍物。

示例 `maze_input.txt` :

```
5 5
0 1 0 0 0
0 1 0 1 0
0 0 0 1 0
0 1 1 0 0
0 0 0 0 0
```

3. **执行脚本:** 在终端中, 使用重定向将输入文件传递给相应的 Python 脚本:

```
python maze_DFS.py < maze_input.txt # 运行 IDDFS
python maze_BFS.py < maze_input.txt # 运行 BFS
python maze_dijkstra.py < maze_input.txt # 运行 Dijkstra
python maze_A_star.py < maze_input.txt # 运行 A*
```

或者, 你可以直接运行 `python <name.py>` , 然后在终端中粘贴迷宫数据并按 `Ctrl+D` (Linux/macOS) 或 `Ctrl+Z` 然后按下 `Enter` 按键 (Windows) 来表示输入结束。

输出说明

脚本执行后会:

1. 在终端打印两行信息:
 - **路径长度:** X (表示路径包含的节点数减 1, 即步数)
 - **实际距离 (路径总代价) :** Y (考虑了斜线移动代价 ($\sqrt{2}$) 的总路径成本)
2. 弹出一个 `matplotlib` 窗口, 动态展示搜索过程和最终路径:
 - 灰色背景代表迷宫通路, 黑色方块代表障碍物。
 - 蓝色小点代表算法访问过的节点, 按访问顺序列出。
 - 红色粗线和圆点代表最终找到的从起点 (0, 0) 到终点 ($N-1, M-1$) 的路径。