School of Data and Computer Science, Sun Yat-sen University

E01 Maze Problem

18340052 何泽

2020 年 10 月 13 日

目录

1 Task

- Please solve the maze problem (i.e., find the shortest path from the start point to the finish point) by using BFS or DFS (Python or C++)
- The maze layout can be modeled as an array, and you can use the data file `MazeData.txt` if necessary.
- Please send `E01_YourNumber.pdf` to `ai_2020@foxmail.com`, you can certainly use `E01_Maze.tex` as the LaTeXtemplate.

2 Codes

```
def DFS(map, x, y, used):
        global res
        if len(res) and len(used) > len(res):
                return
        if x == -1 or y == -1 or x == len(map) or y == len(map[0]) or
           map[x][y] == '1' or (x, y) in used:
                return
        elif map[x][y] == 'E':
                print('\033[1;30;46m %d \033[0m' % (len(used) + 1), end=
                   "")
        if len(used) + 1 < len(res) or len(res) == 0:
                res = used
                res.append((x, y))
                return
        else:
                used.append((x, y))
                DFS(map, x + 1, y, used[:])
                DFS(map, x - 1, y, used[:])
                DFS(map, x, y + 1, used[:])
                DFS(map, x, y - 1, used[:])
        used.remove(used[-1])


def print_result(map, path):
        print(' ')
        print('\033[1;30;46m                    最短路径长度:  %d
                              \033[0m' % len(path))
```

```
25          print('图示：')
26          for i in range(len(map)):
27                  for j in range(len(map[i])):
28                          if (i, j) in path[1:-1]:
29                                  print('\033[1;32;43m  \033[0m', end="")
30                          elif map[i][j] == "1":
31                                  print('\033[1;33;44m  \033[0m', end="")
32                          elif map[i][j] == "S":
33                                  print('\033[1;30;41mS \033[0m', end="")
34                          elif map[i][j] == "E":
35                                  print('\033[1;30;45mE \033[0m', end="")
36                          else:
37                                  print("  ", end="")
38                  print("")
39          print("")
40
41
42  if __name__ == "__main__":
43          print('\033[1;30;46m            何泽-18340052-人工智能实验一：Maze
                    \033[0m')
44          print('\033[1;30;44m  蓝色是墙  \033[0m', end="")
45          print('\033[1;30;41m  红色是起始点  \033[0m', end="")
46          print('\033[1;30;45m  紫色是终点  \033[0m', end="")
47          print('\033[1;30;43m  黄色是最短路径  \033[0m')
48          print('\033[1;30;46m  历史路径长度:  \033[0m', end="")
49          Maze = []
50          res = []
51          with open("./MazeData.txt", 'r') as Maze_og:
52                  for i in Maze_og.readlines():
53                          if i[0] == '1' or i[0] == '0':
54                  Maze.append(i[:-1])
55
56          for i in range(len(Maze)):
57                  for j in range(len(Maze[i])):
58                          if Maze[i][j] == 'S':
59                                  DFS(Maze, i, j, [])
60                                  break
61
```

```
62          print_result(Maze, res)
```

3 Results

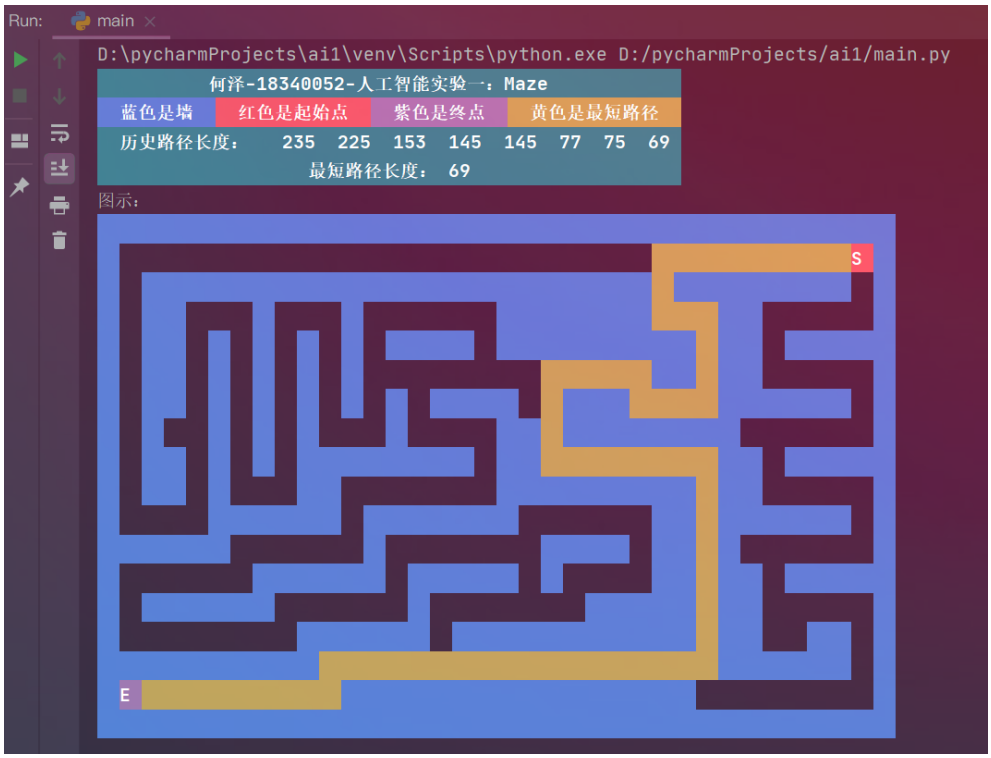

图 1: Result

- 首先说明了迷宫各种颜色的含义
- 我的算法采用了深度优先搜索
- 每找到一条路径，就会记录下来当前路径长度并打印，并给出最短路径长度
- 在图示中用色块画出了迷宫和最短路径，可以看出找出的路径确实是最短的