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

E01 Maze Problem

19214808 Yikun Liang

October 8, 2020

Contents

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.

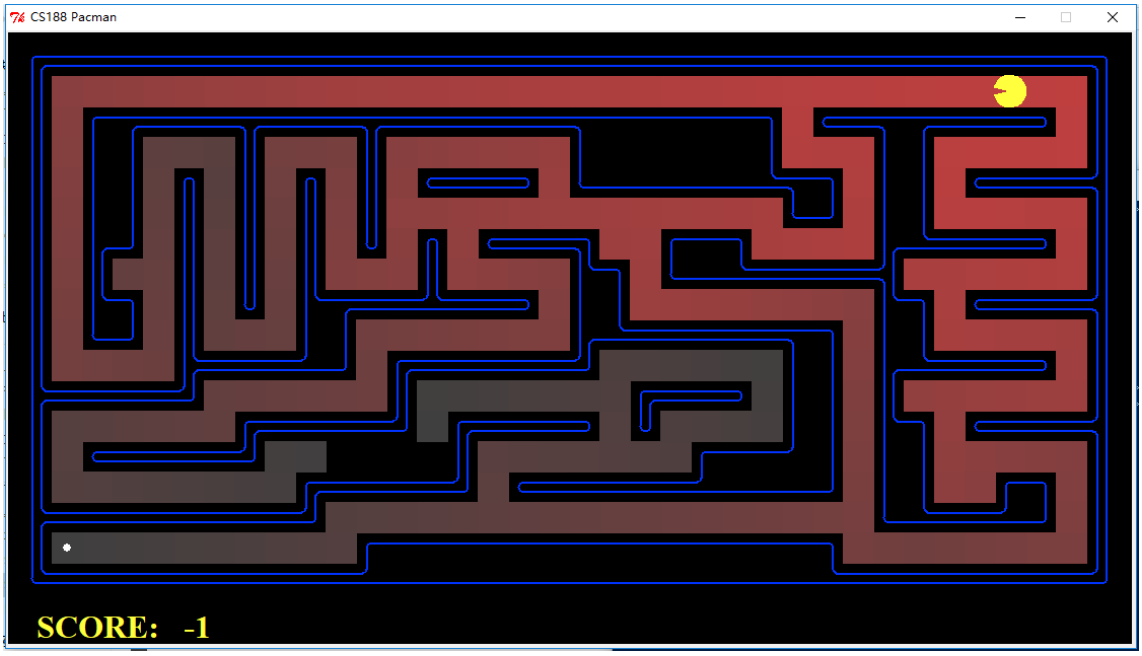

Figure 1: Searching by BFS or DFS

2 Codes

```
import queue
directions = ((-1, 0), (1, 0), (0, -1), (0, 1))
def bfs(q, maze):
    while True:
        cur = q.get()
        for direction in directions:
            next = (cur[0] + direction[0], cur[1] + direction[1])
            # This is used to avoid crossing the boundary.
            try:
                next_val = maze[next[0]][next[1]]
            except:
                continue
            ''' This is used to log last point's position,
            in order to prevent the process from visiting a point twice,
```

```
            and to jot down paths.'''
            if next_val == '0':
                maze[next[0]][next[1]] = cur
                q.put(next)
            elif next_val == 'E':
                maze[next[0]][next[1]] = cur
                return next
def read_maze(maze):
    with open("MazeData.txt") as f:
        while True:
            line = f.readline()
            if not line:
                break
            maze.append(list(line))
def init_queue(q):
    for i in range(len(maze)):
        for j in range(len(maze[i])):
            if maze[i][j] == 'S':
                q.put((i, j))
def print_path(maze):
    path = list()
    path.append(end)
    while True:
        cur = maze[path[-1][0]][path[-1][1]]
        if cur == 'S':
            break
        path.append(cur)
    while path:
        cur = path.pop()
        print(cur, end = '')
maze = list()
read_maze(maze)
q = queue.Queue()
init_queue(q)
end = bfs(q, maze)
print_path(maze)
```

3 Results

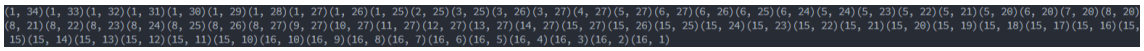

Figure 2: Results