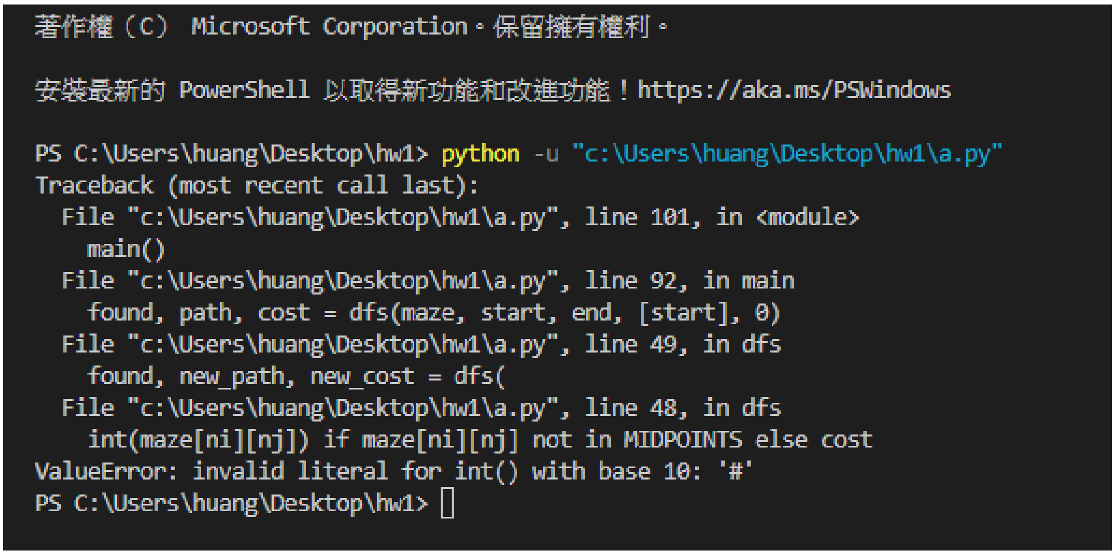
人工智慧

HW1

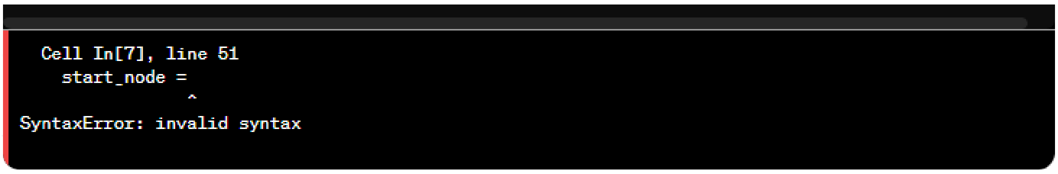
資訊三乙 11027222 黃彥霖

原本是以 Gpt4.0 做迷宮路徑，但他跑出來會有很多 ERROR，如以下的轉換問題，雖然後來自己有成功修正，但他的走出來的路徑非常奇怪，問了很多次都是一樣，沒辦法成功改出正確的步數，後來就用 GPT4.0。

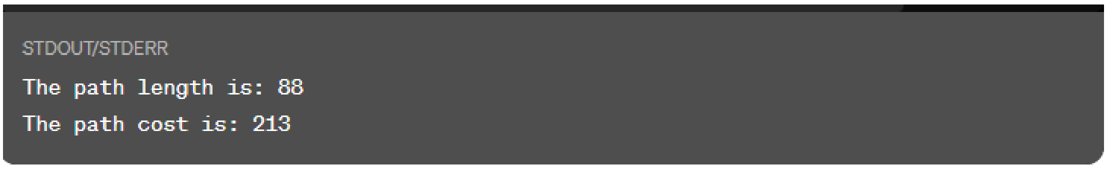


```
著作權 (C)  Microsoft Corporation。保留擁有權利。

安裝最新的 PowerShell 以取得新功能和改進功能！https://aka.ms/PSWindows

PS C:\Users\huang\Desktop\hw1> python -u "c:\Users\huang\Desktop\hw1\a.py"
Traceback (most recent call last):
  File "c:\Users\huang\Desktop\hw1\a.py", line 101, in <module>
    main()
  File "c:\Users\huang\Desktop\hw1\a.py", line 92, in main
    found, path, cost = dfs(maze, start, end, [start], 0)
  File "c:\Users\huang\Desktop\hw1\a.py", line 49, in dfs
    found, new_path, new_cost = dfs(
  File "c:\Users\huang\Desktop\hw1\a.py", line 48, in dfs
    int(maze[ni][nj]) if maze[ni][nj] not in MIDPOINTS else cost
ValueError: invalid literal for int() with base 10: '#'
PS C:\Users\huang\Desktop\hw1> 
```

比較 4.0 和 3.5，4.0 會自己執行過一次程式，確定不會有 compile error，都確定是對的之後，也能自動幫你驗證答案，GPT 可以自己分析他的答案對錯，減少你自己去 debug 的機會，雖然他沒辦法重新跑很多次，但他會想辦法達到你所要求的答案。

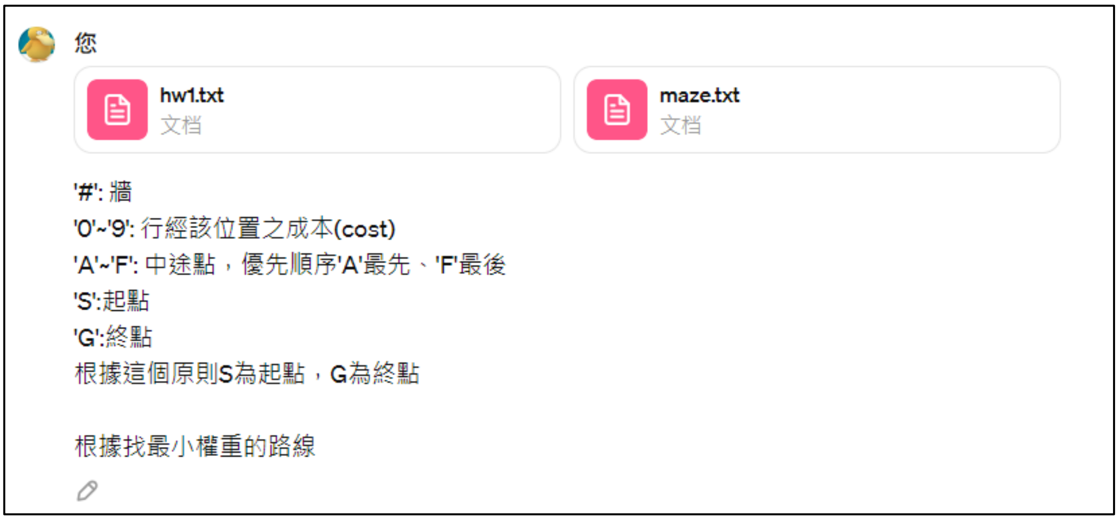


```
  Cell In[7], line 51
    start_node =
                ^
SyntaxError: invalid syntax
```



```
STDOUT/STDERR
The path length is: 88
The path cost is: 213
```

提問流程:



您

hw1.txt
文档

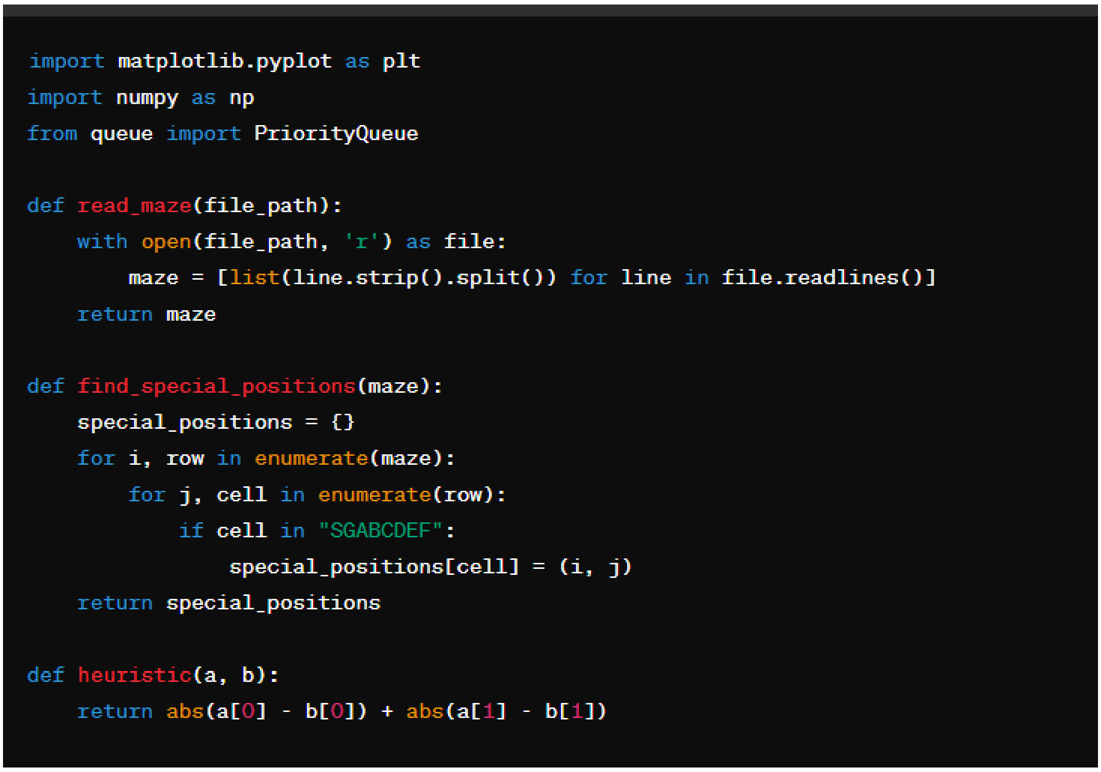
maze.txt
文档

'#': 牆
'0'~'9': 行經該位置之成本(cost)
'A'~'F': 中途點，優先順序'A'最先、'F'最後
'S':起點
'G':終點
根據這個原則S為起點，G為終點

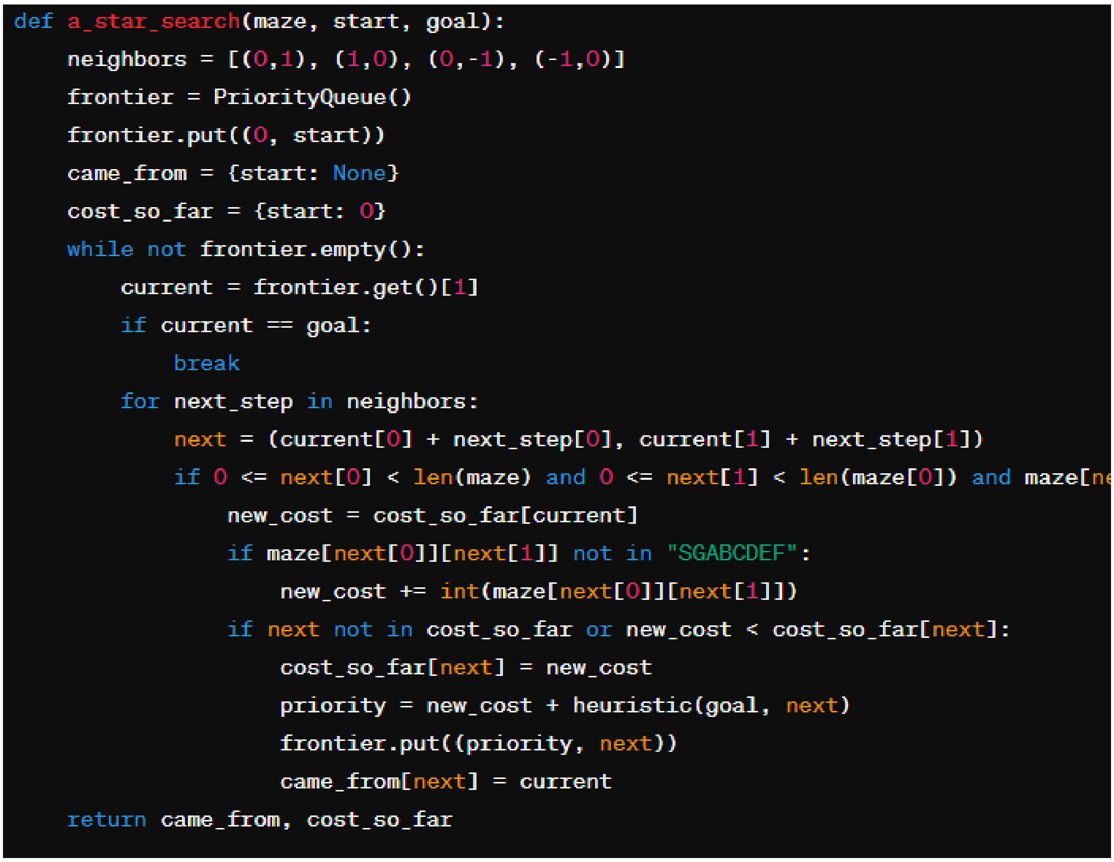
根據找最小權重的路線

因為用過 3.5 和 4.0，3.5 大概花了兩小時做出來，而且跟他說有錯誤的地方，但他一直不會去改。而 4.0 約用了 20 分鐘，而且可以丟入檔案讓他更方便的去做判斷。

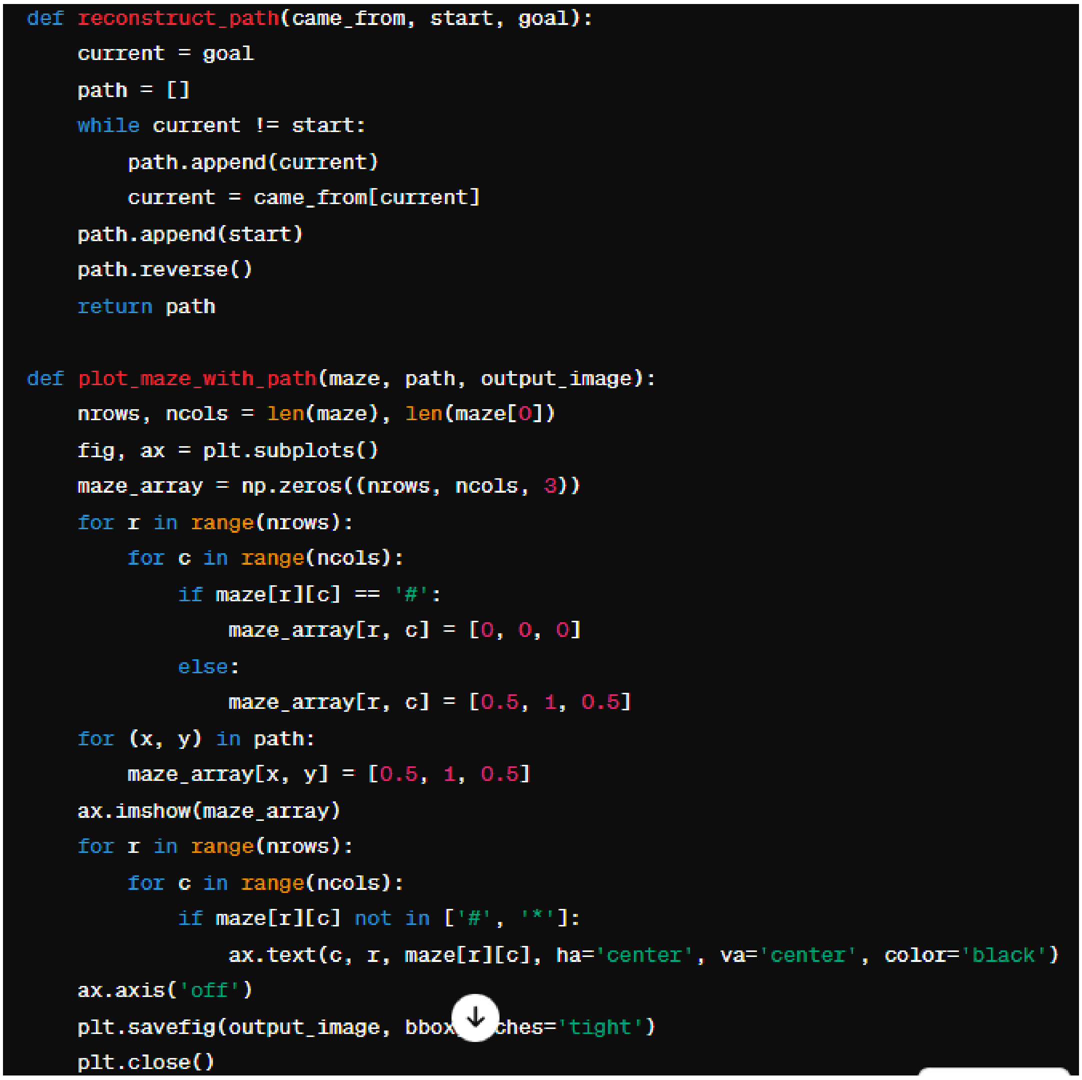
以下為 GPT 直接分析出的結果，且答案正確



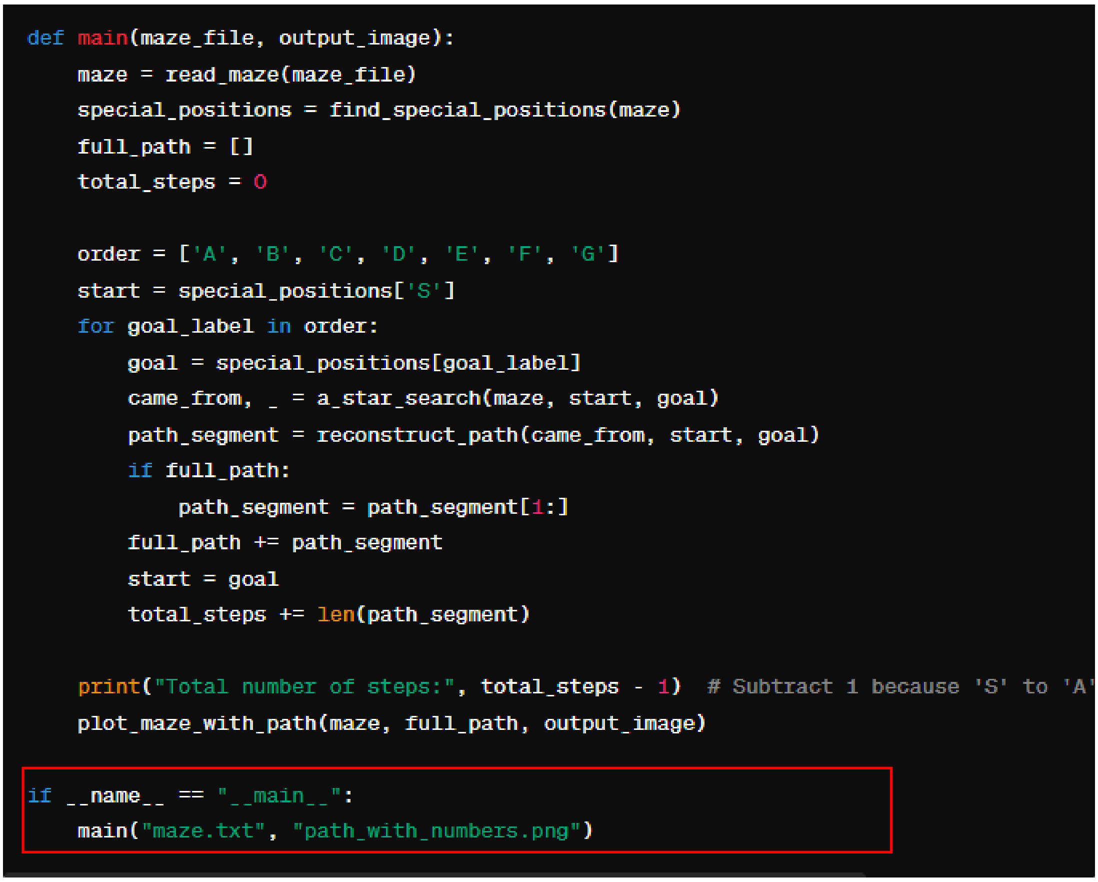
```python
import matplotlib.pyplot as plt
import numpy as np
from queue import PriorityQueue

def read_maze(file_path):
    with open(file_path, 'r') as file:
        maze = [list(line.strip().split()) for line in file.readlines()]
    return maze

def find_special_positions(maze):
    special_positions = {}
    for i, row in enumerate(maze):
        for j, cell in enumerate(row):
            if cell in "SGABCDEF":
                special_positions[cell] = (i, j)
    return special_positions

def heuristic(a, b):
    return abs(a[0] - b[0]) + abs(a[1] - b[1])
```



```
def a_star_search(maze, start, goal):
    neighbors = [(0,1), (1,0), (0,-1), (-1,0)]
    frontier = PriorityQueue()
    frontier.put((0, start))
    came_from = {start: None}
    cost_so_far = {start: 0}
    while not frontier.empty():
        current = frontier.get()[1]
        if current == goal:
            break
        for next_step in neighbors:
            next = (current[0] + next_step[0], current[1] + next_step[1])
            if 0 <= next[0] < len(maze) and 0 <= next[1] < len(maze[0]) and maze[ne
                new_cost = cost_so_far[current]
                if maze[next[0]][next[1]] not in "SGABCDEF":
                    new_cost += int(maze[next[0]][next[1]])
                if next not in cost_so_far or new_cost < cost_so_far[next]:
                    cost_so_far[next] = new_cost
                    priority = new_cost + heuristic(goal, next)
                    frontier.put((priority, next))
                    came_from[next] = current
    return came_from, cost_so_far
```

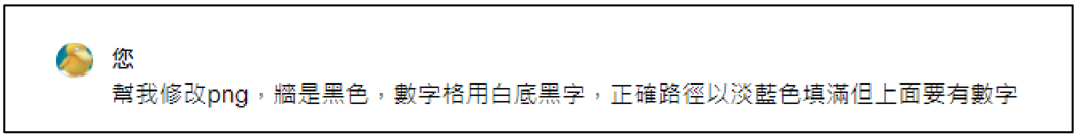


```
def reconstruct_path(came_from, start, goal):
    current = goal
    path = []
    while current != start:
        path.append(current)
        current = came_from[current]
    path.append(start)
    path.reverse()
    return path

def plot_maze_with_path(maze, path, output_image):
    nrows, ncols = len(maze), len(maze[0])
    fig, ax = plt.subplots()
    maze_array = np.zeros((nrows, ncols, 3))
    for r in range(nrows):
        for c in range(ncols):
            if maze[r][c] == '#':
                maze_array[r, c] = [0, 0, 0]
            else:
                maze_array[r, c] = [0.5, 1, 0.5]
    for (x, y) in path:
        maze_array[x, y] = [0.5, 1, 0.5]
    ax.imshow(maze_array)
    for r in range(nrows):
        for c in range(ncols):
            if maze[r][c] not in ['#', '*']:
                ax.text(c, r, maze[r][c], ha='center', va='center', color='black')
    ax.axis('off')
    plt.savefig(output_image, bbox[illegible]ches='tight')
    plt.close()
```



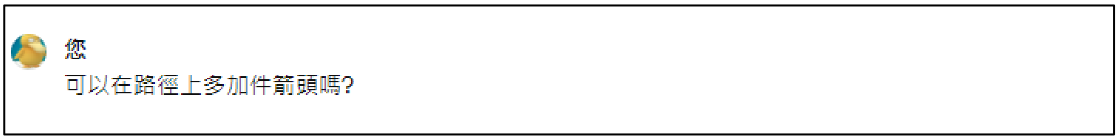
```
def main(maze_file, output_image):
    maze = read_maze(maze_file)
    special_positions = find_special_positions(maze)
    full_path = []
    total_steps = 0

    order = ['A', 'B', 'C', 'D', 'E', 'F', 'G']
    start = special_positions['S']
    for goal_label in order:
        goal = special_positions[goal_label]
        came_from, _ = a_star_search(maze, start, goal)
        path_segment = reconstruct_path(came_from, start, goal)
        if full_path:
            path_segment = path_segment[1:]
        full_path += path_segment
        start = goal
        total_steps += len(path_segment)

    print("Total number of steps:", total_steps - 1)  # Subtract 1 because 'S' to 'A'
    plot_maze_with_path(maze, full_path, output_image)

if __name__ == "__main__":
    main("maze.txt", "path_with_numbers.png")
```

但是他第一次雖然跑出來的結果是對的，但是他輸出的圖很糟糕，無法直觀的判斷路徑。直接向 GPT 請求要求更改即可。程式中唯一有修改的路徑只有以上框起來的部分，就是把路徑和檔名改成自己資料夾路徑而已。程式沒有其他的瑕疵。



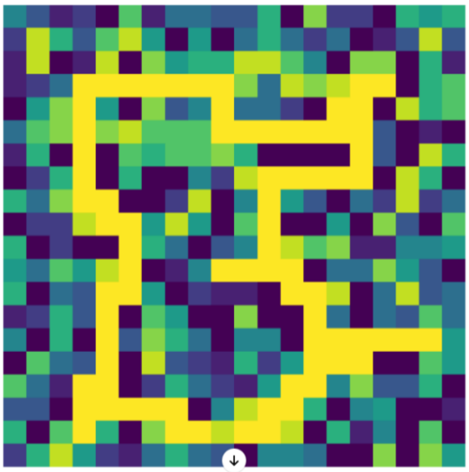
您

幫我修改png，牆是黑色，數字格用白底黑字，正確路徑以淡藍色填滿但上面要有數字

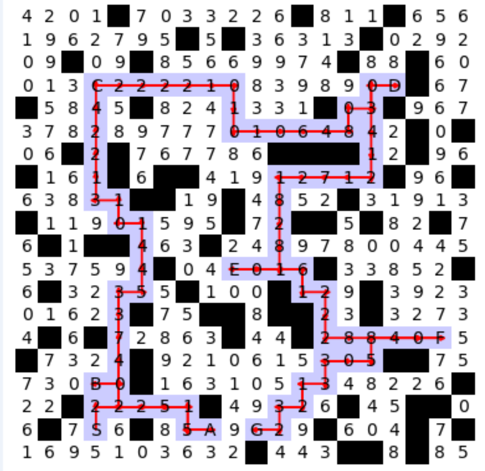


您

可以在路徑上多加件箭頭嗎?



前



4	2	0	1	■	7	0	3	3	2	2	6	■	8	1	1	■	6	5	6
1	9	6	2	7	9	5	■	5	■	3	6	3	1	3	■	0	2	9	2
0	9	■	0	9	■	8	5	6	6	9	9	7	4	■	8	8	■	6	0
0	1	3	C	2	2	2	2	1	0	8	3	9	8	9	0	D	■	6	7
■	5	8	4	5	■	8	2	4	1	3	3	1	■	0	3	■	9	6	7
3	7	8	2	8	9	7	7	7	0	1	0	6	4	8	4	2	■	0	■
0	6	■	2	■	7	6	7	7	8	6	■	■	■	■	1	2	■	9	6
■	1	6	1	■	6	■	■	4	1	9	1	2	7	1	2	■	9	6	■
6	3	8	3	1	■	■	1	9	■	4	8	5	2	■	3	1	9	1	3
■	1	1	9	0	1	5	9	5	■	7	2	■	■	5	■	8	2	■	7
6	■	1	■	■	4	6	3	■	2	4	8	9	7	8	0	0	4	4	5
5	3	7	5	9	4	■	0	4	E	0	1	6	■	3	3	8	5	2	■
6	■	3	2	3	5	5	■	1	0	0	■	1	2	9	■	3	9	2	3
0	1	6	2	3	■	7	5	■	■	8	■	■	2	3	■	3	2	7	3
4	■	6	■	7	2	8	6	3	■	4	4	■	2	8	8	4	0	F	5
■	7	3	2	4	■	9	2	1	0	6	1	5	3	0	5	■	■	7	5
7	3	0	B	0	■	1	6	3	1	0	5	1	3	4	8	2	2	6	■
2	2	■	2	2	2	5	1	■	4	9	3	2	6	■	4	5	■	■	0
6	■	7	S	6	■	8	5	A	9	G	2	9	■	6	0	4	■	7	■
1	6	9	5	1	0	3	6	3	2	■	4	4	3	■	■	8	■	8	5

後