CODE FOR SHORTEST PATH

from collections import deque

def shortest\_path(grid):

rows, cols = len(grid), len(grid[0])

directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]

queue = deque([(0, 0)])

visited = set([(0, 0)])

path = {}

while queue:

row, col = queue.popleft()

if (row, col) == (rows - 1, cols - 1):

break

for dr, dc in directions:

new\_row, new\_col = row + dr, col + dc

if 0 <= new\_row < rows and 0 <= new\_col < cols and grid[new\_row][new\_col] == 'Y' and (new\_row, new\_col) not in visited:

visited.add((new\_row, new\_col))

queue.append((new\_row, new\_col))

path[(new\_row, new\_col)] = (row, col)

if (rows - 1, cols - 1) not in path:

return None # No path found

# Reconstruct path

current = (rows - 1, cols - 1)

shortest\_path = [current]

while current != (0, 0):

current = path[current]

shortest\_path.append(current)

return shortest\_path[::-1]

grid = [['Y', 'B', 'Y', 'Y', 'B'],

['Y', 'B', 'Y', ‘Y’, 'Y'],

['Y', 'Y', 'Y', 'Y', 'Y'],

['Y', 'B', 'Y', 'B', 'Y'],

['Y', 'Y', 'Y', 'B', 'Y']]