**NAME: KATO JOSEPH BWANIKA  
REG: 2023-B291-11709  
COURSE UNIT: ARTIFICIAL INTELLIGENCE**

**Assignment 1.**

**Qn1. Application (Problem Solving with Algorithms)  
Maze Problem**

**Given a 5x5 (or 10x10) grid maze, identify:  
a) Start state  
b) Goal state  
c) Possible actions from each cell  
d) A valid solution path using BFS**

**Solution:**

1. **Maze Problem (5x5 Grid)**

**a) Start State**

The start state is the initial position from where the path search begins. In our maze, this is at the top-left cell with coordinates (0, 0).

**b) Goal State**

The goal state is the destination cell we want to reach. In this maze, it is the bottom-right cell at coordinates (4, 4).

**c) Possible Actions from Each Cell**

From any position, the agent can move in four directions when not blocked by walls:

* Up: means moving to the cell above, which is one row less but in the same column.
* Down: means moving to the cell below, which is one row more but in the same column.
* Left: means moving to the cell on the left, which is in the same row but one column less.
* Right: means moving to the cell on the right, which is in the same row but one column more.

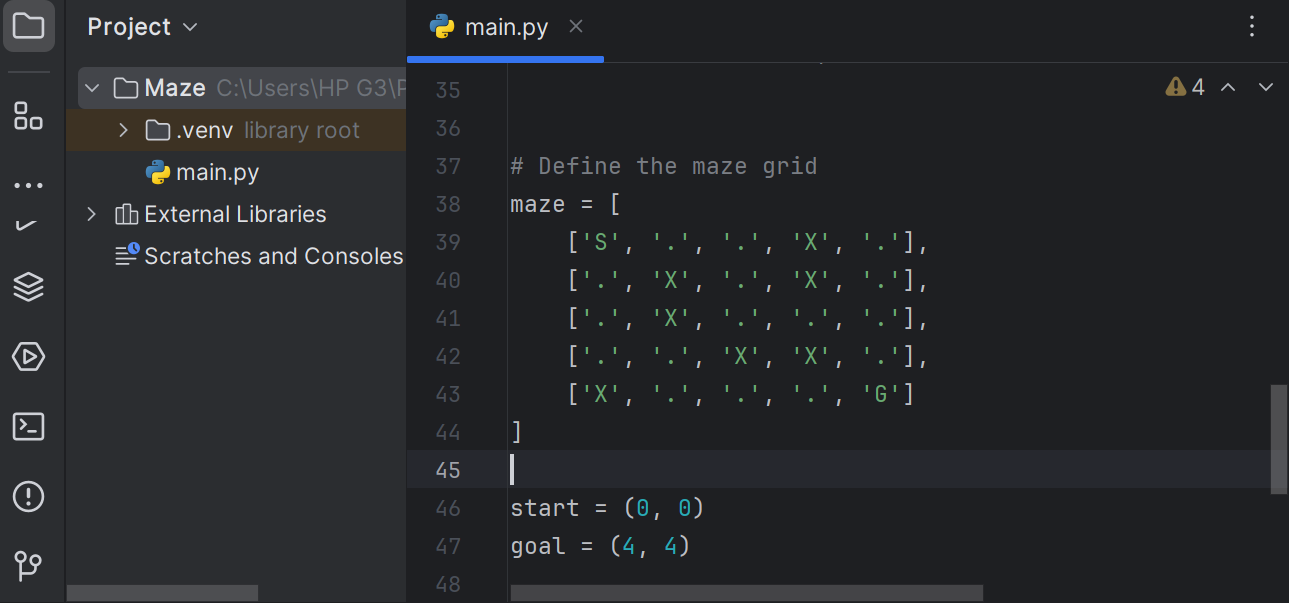
**d) Valid Solution Path Using BFS.**

Breadth-First Search explores all neighbors at the current depth before moving deeper, guaranteeing the shortest path length in a grid without weighted edges.

**ii) Explanation of how the Breadth-First Search works.**

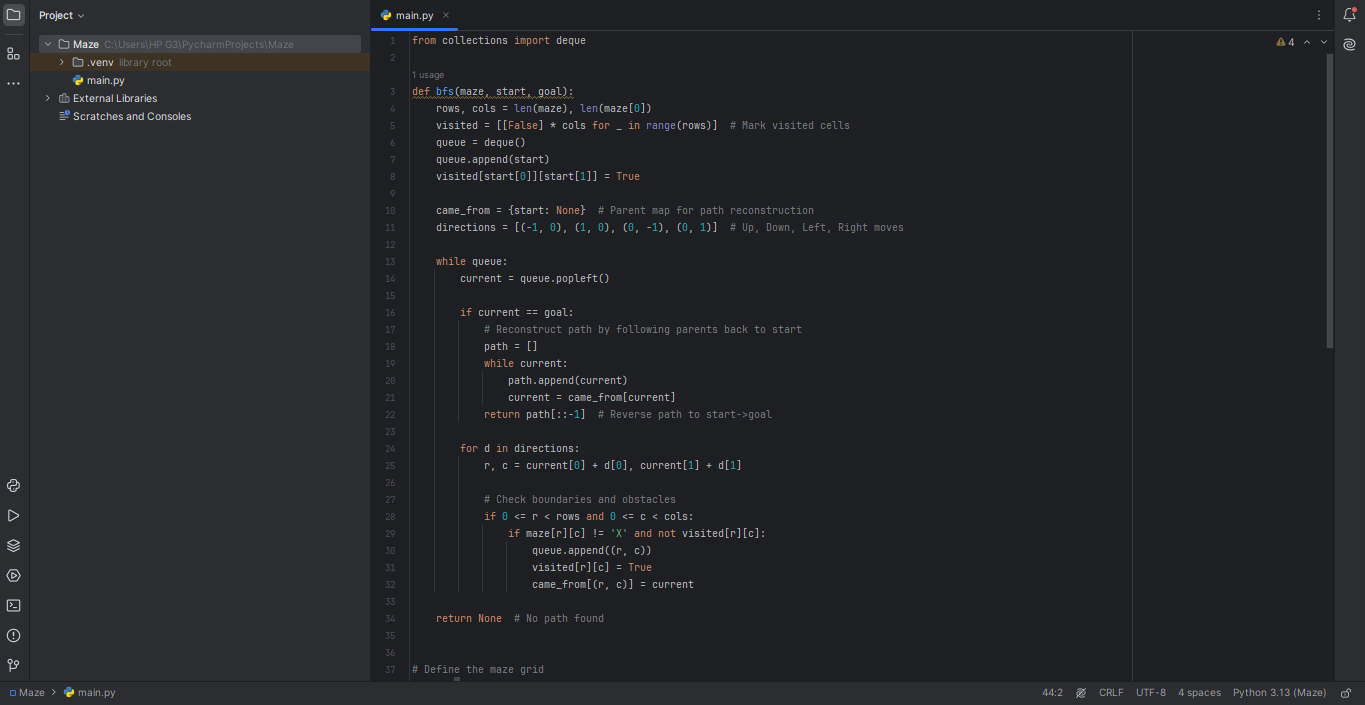
BFS works by starting at the beginning and putting that spot in a queue. It marks this spot as visited and remembers where each next step comes from. Then, it checks each spot in the queue one by one. If it finds the goal, it stops. If not, it adds all nearby open spots that haven’t been visited yet to the queue, marks them visited, and remembers their “parent” spot. Once the goal is found, it follows these parents backward to find the full path from start to goal.

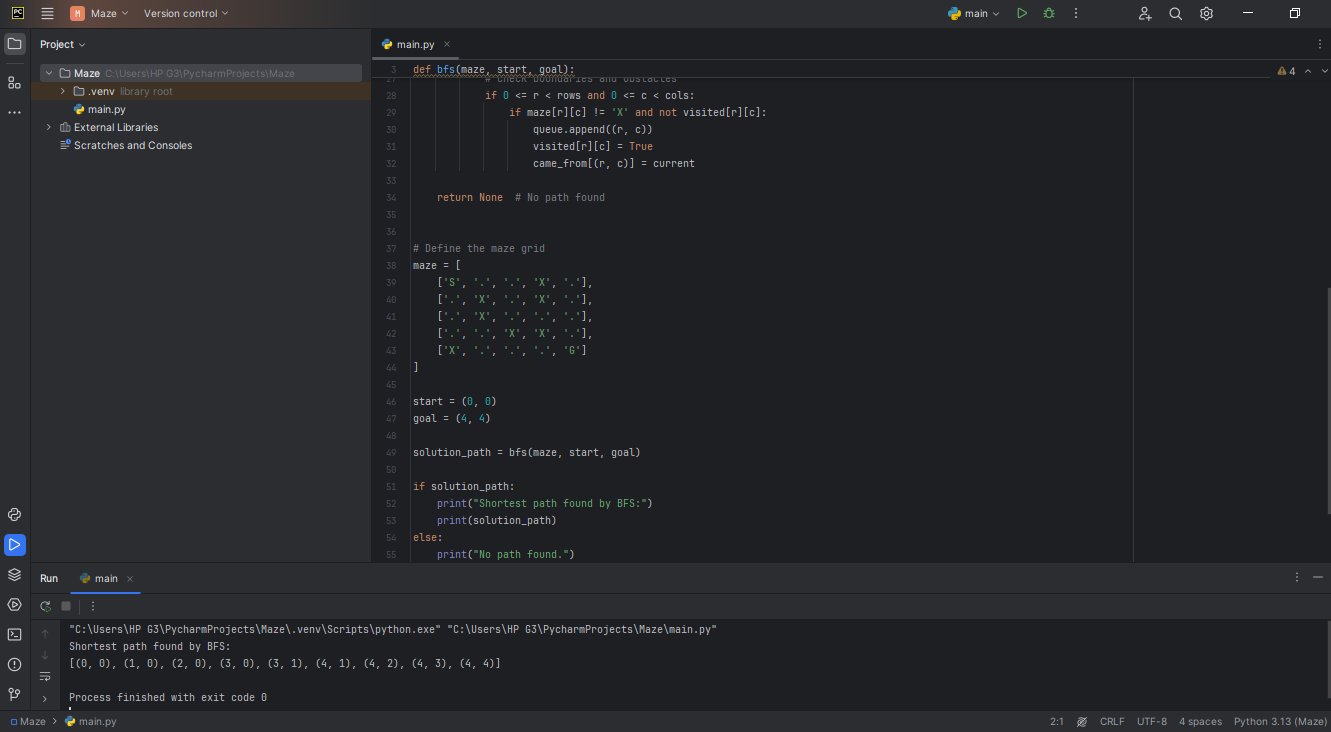
**iii) Maze Grid**



* 'S' stands for Start at (0,0)
* 'G' stands for our Goal (4,4)
* '.' stands for an Open cell
* 'X' stands for a Wall

**iv) Code showing the shortest path.**





**Shortest path found by BFS in our code snippet is**; [(0, 0), (1,0), (2,0), (3,0), (3,1), (4,1), (4,2), (4,3), (4, 4)]

**Qn2. You are designing an AI system for ambulance dispatch in a city. Define the state space, actions, goal, and path cost for this problem. Suggest a suitable search strategy and justify your choice.**

**Solution:**

1. **State Space**

The state space represents all possible arrangements of ambulances, emergency requests, and other relevant factors across the city at any given time. It includes the locations and statuses of all ambulances such as available, busy, or en route, the locations of emergency calls waiting for response, the positions and availability of hospitals, and real-time traffic and road conditions that affect travel times.

1. **Actions**

Actions are the possible decisions or moves made from one state to another, such as dispatching a specific ambulance to an emergency location, re-routing an ambulance based on updated traffic information, returning an ambulance to a hospital or standby location after a call, and prioritizing emergency calls for ambulance allocation.

1. **Goal**

The goal of the system is to ensure that all emergency calls are served efficiently by dispatching an ambulance that reaches the scene in minimum response time, optimizing ambulance availability so that no calls remain unassigned, and minimizing total response times or maximizing patient survival chances.

1. **Path Cost**

The path cost models the cost associated with transitions between states and could be defined as the total time taken for ambulances to reach emergency calls, the sum of travel times including detours due to traffic, a weighted combination of factors like time, distance, and ambulance utilization, or the cumulative travel or response time across actions taken.

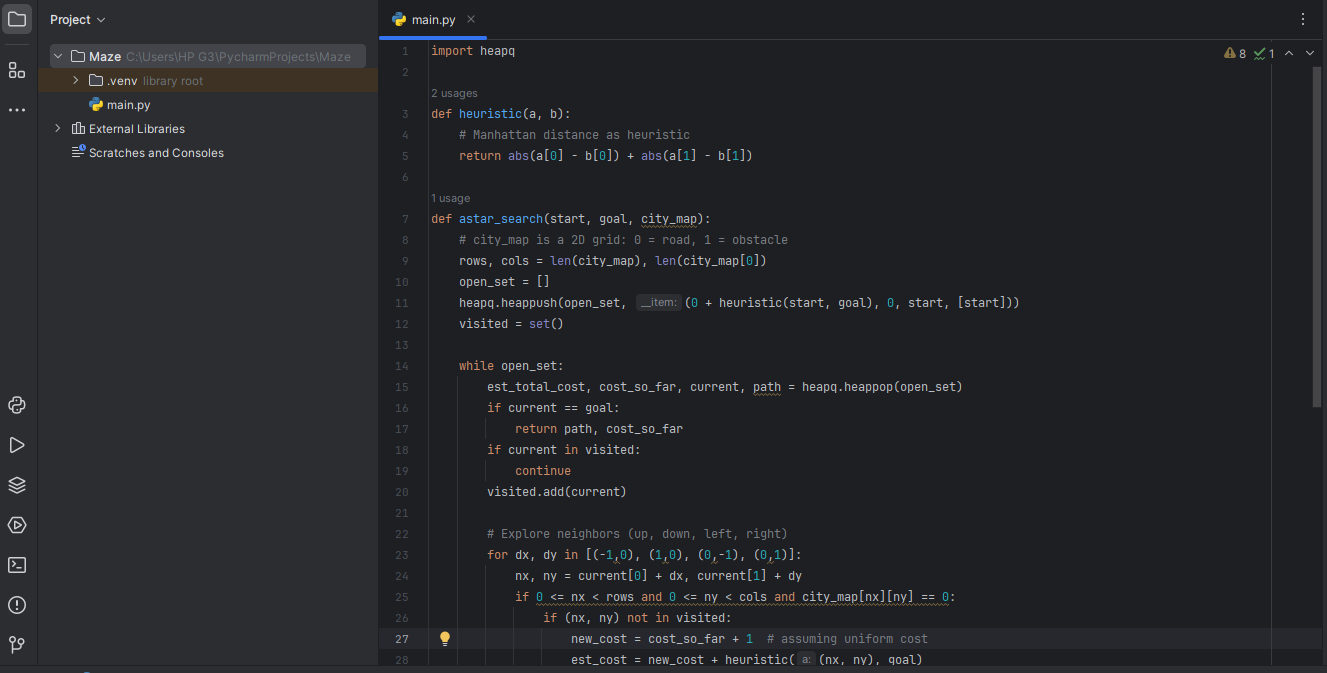
**Suggested Search Strategy:**

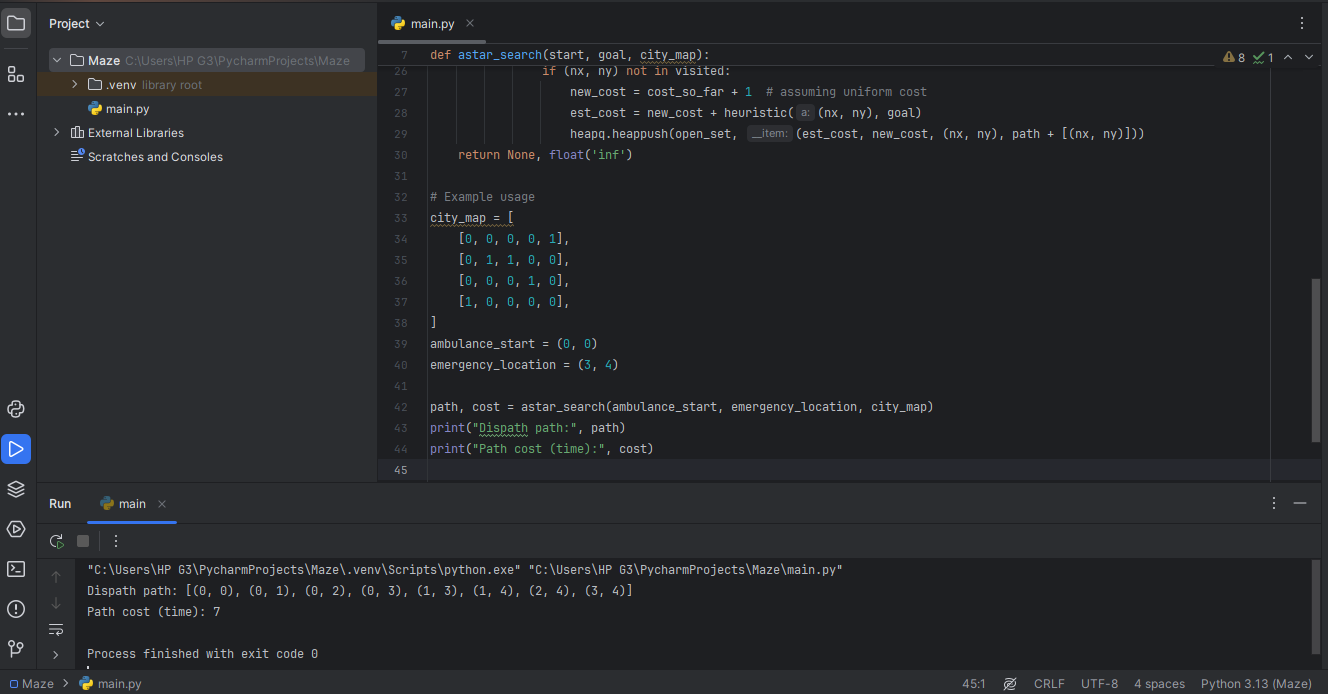
1. **A\* search algorithm.**

* The most suitable search strategy for the ambulance dispatch problem is the A\* search algorithm.
* This algorithm is particularly useful because it combines the actual cost incurred so far with a heuristic estimate of the remaining cost to the goal.
* In the context of ambulance dispatch, this means evaluating both the time already spent traveling and an estimate of how much longer it will take to reach the emergency location.
* This dual consideration helps A\* focus on the most promising routes first, making it efficient even in large and complex city maps. Additionally, A\* can easily incorporate real-time information such as traffic conditions or road closures into its heuristic, allowing the system to dynamically adapt and find the quickest possible response routes.

1. **Justification**

* A\* is preferable over uninformed search methods like breadth-first search or depth-first search, which would waste valuable time exploring many irrelevant paths.
* The algorithm’s ability to prioritize routes by combining known costs and intelligent guesses ensures more effective and timely ambulance dispatch decisions. Importantly, as long as the heuristic used does not overestimate the actual remaining travel time, A\* guarantees finding the shortest path to the emergency.
* This guarantee of optimality and efficiency is crucial when every second counts in emergency medical response. Therefore, A\* offers a balanced approach that is both practical and theoretically sound, making it ideally suited for real-world ambulance dispatch applications.

****

****

* **Path Cost Time;** 7
* **Shortest path found by BFS in our code snippet is**; [(0, 0), (0,1), (0,2), (0,3), (1,3), (1,4), (2,4), (3,4)]