Coding Assignment-1 :: Artificial Intelligence

Instructions:

1. Submit report with output screenshots and proper explanation.
2. Plagiarism is strictly prohibited, if found zero marks will be awarded.
3. Submit all the code in a single zipped folder.
4. Total marks = 20

A.) Dynamic Goal-Based Agent for Warehouse Logistics Optimization.

A robotic agent operates in a warehouse modeled as an N×M grid environment. The agent starts at a predefined loading dock and must deliver packages to multiple destinations marked on the grid while avoiding dynamically placed obstacles.

Take suitable values of the following parameters.
- **Warehouse dimensions**: N×M grid size (M,N between 5 and 10, inclusive)
- **Number of packages**: P (between 2 and 6, inclusive)
- **Number of obstacle**s: O (between 1 and 10, inclusive)
- **Package locations**: (X1, Y1), (X2, Y2), ... (XP, YP)
- **Drop-off locations**: (D1X, D1Y), (D2X, D2Y), ... (DPX, DPY)
- **Robot starting position**: S=(S_x,S_y), starts at a fixed cell but moves dynamically
- **Movement cost**: Each movement incurs a cost of 1 unit
- **Delivery reward**: Successfully delivering a package adds 10 units to the total reward
- **Obstacle penalty**: Hitting an obstacle results in a (-5) penalty

Note : Packages locations and drop-off locations should not overlap.

Q1. Represent the warehouse as an N×M matrix. Place the packages, drop-off points, and obstacles randomly. Display the initial warehouse configuration.
Q2. Implement a goal-based agent that can identify all goals, plan a sequence of actions to reach the goal, use a search algorithm (BFS, DFS, or UCS) to find optimal paths, deliver all packages, and calculate the total cost.
Q3. Choose a random seed value for the ease of reproducing the results. Your program should give outputs: the chosen path taken by the agent, total cost and rewards, final score based on penalties, movement costs, and successful deliveries.

B.) Suppose you and your friend live in different cities on a map and you both are planning for a common meetup place which is optimal for you both. Let’s find the common meetup with a little twist.

On every turn, you and your friend can simultaneously move to a neighboring city on the map. The amount of time needed to move from city i to neighbor j is equal to the straight line distance d(i, j)x2 between the cities, but on each turn the friend that arrives first must wait until the other one arrives (and calls the first on his/her cell phone) before the next turn can begin. The heuristic you are assuming is the straight line distance d(i,j). You both friends want to meet as quickly as possible.

1. Formulate this search problem and display the map of the city with heuristic (h) and transition cost (g).
2. Implement search strategies: a. Greedy Best First Search b. A* and provide the Search Cost (nodes generated, space and time taken for execution) for both of them.

3. Change the heuristic function d(i, j) from the straight-line distance between cities i and j to more realistic functions such as road or train route distance. And provide the analysis of the solution for different heuristic functions in terms of Search Cost (nodes generated, space and time taken for execution).

Note:

1. Use maps of India for the cities [Taluka level granularity is expected]. One city will be your current city and Your friend City should be at least in a different state. You can take lat, long of the Taluka center for calculating the distance.
2. Take necessary Assumptions.

Material:
Can use the following links for maps.
https://surveyofindia.gov.in/pages/political-map-of-india
Can use QGIS for initial preprocessing of the maps or Geopandas.