

CSC520 Fall 2021 Assignment 1

Due August 30th at 11:59pm

This assignment consists of two questions which involve written answers and code. In order to complete the assignment you must submit a written report in pdf form detailing your answers to the questions as well as your code. As discussed in class all work *must* be your own. You may not use third party libraries or example code to complete the assignment. All reports must be clear and well written. All code must be clear, readable, and well-commented.

1 Question 1 Agents and Environments (40pts)

Consider the four problem scenarios listed below. For each scenario answer the following questions and provide a justification for your answer.

1. What is the appropriate PEAS specification for an agent that solves the problem and what are the features of the environment?
2. What is the *least* powerful agent that can solve the problem, and why?

Scenario 1: *Robocup* Assume we have a robot soccer player which can run and shot. This robot play a traditional soccer game with other five Robot Soccer Players. The robots do not know the locations of each other but their sensors can feel other robots if they are close enough. Meanwhile, there is no connection between robots, but if they are close enough, they can also detect if they are in the same team(cooperation) or not(conflict). Every robots also have sensors to tell where is the ball and the goal. The agent is assigned a task of making goals as many as it can for winning the game.

Scenario 2: *Deep Sea Drone* A deep sea research drone designed to measure subsurface ocean currents by diving to a pre-specified depth, recording salinity and temperature information at a set location, and then returning to the surface to transmit data.

Scenario 3: *Intelligent Monitor* can be deployed in some storehouses. This monitor will stay in standby mode under normal conditions. But it will take videos when it detect moving target by sensors. The monitor and sensors can cover all place in the storehouse.

Scenario 4: *Intelligent Infrastructure Management* In large organizations many online resources exist that go unused most of the time. These include virtual machines or other system resources, and soft-switch hardware resources such as networking kits. Assume that you have an intelligent agent that is tasked with managing these resources to maintain minimal power usage and fast user responses.

b

	0	1	2	3
0	2	1	1	3
1	2	1	2	3
2	1	1	2	3
3	3	G	3	1

Solution: 0,0; 2,0; 2,1; 3,1

Figure 1: Number Maze Example.

Question 2 Search (60 points)

A number-maze is a matrix representation for a maze problem. Each cell in the maze contains a number or a "G" indicating the goal state. A sample number maze is shown in Figure 1. Solvers start the maze at the entry point at cell 0,0. At each cell they can move up, down, left, or right, the number of cells listed in the current cell, unless they hit an edge. Thus in the example shown below they can move down or to the right by 2 cells to cells 0,2 or 2,0 respectively. A number maze is complete when you have traversed through it to find the goal state marked "G".

For the purposes of this assignment you have been given two sample mazes, the 4x4 maze shown below, and the 6x6 maze along with their ideal answers. You have also been given two additional mazes 8x8 and 12x12 without answers. For this question you should:

1. Define an admissible or consistent heuristic for the maze using one of the techniques in class and justify that here. Identify any limitations of your algorithm.
2. Implement a maze solver using state-space search that implements DFS, Best First, and A* search with your chosen heuristic. Your code must take a single argument specifying the maze file and must output an answer file in the same format supplied. The dimensions of the puzzle and other information must be read from the file itself. Note that your code will be tested on other unseen mazes.
3. Use the implemented solvers to solve each maze, including the example mazes. For each maze provide the following information:
 - (a) The shortest path.
 - (b) The number of states expanded by each algorithm when finding the path.
 - (c) The *total* number of unique paths through the graph for the 6x6.
 - (d) The number of states expanded in finding the paths.

All code must be clearly documented and be your own work. You are not permitted to use third party libraries, shared code, or demonstration code.