

ED5215: Introduction to Motion Planning

Assignment - 1

Due on 20th February **morning 9 AM**

Total marks: 25

Question 1: (Marks: 10)

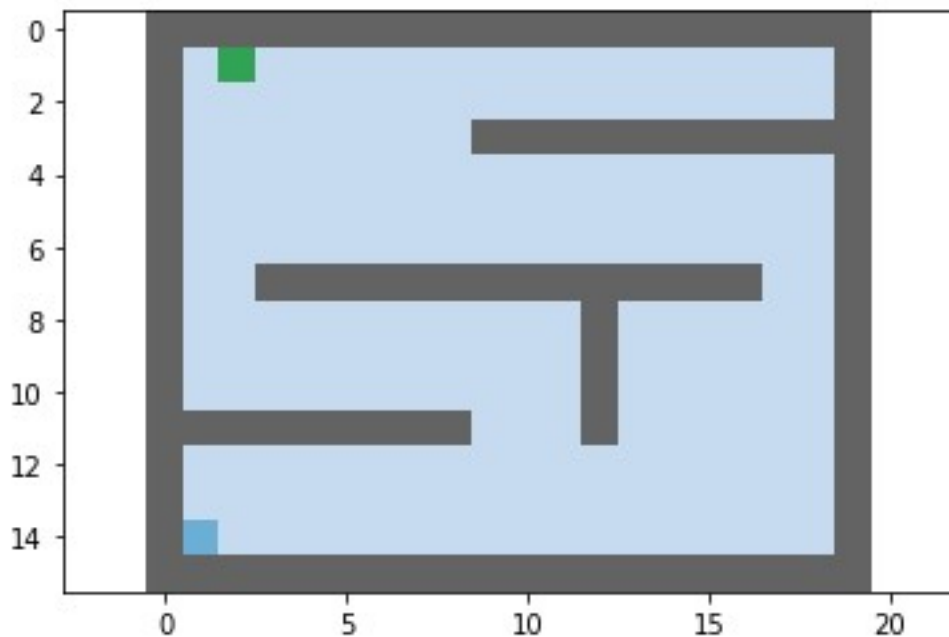


Figure: Maze_1

You are provided with the above maze. Blue denotes the start point, Green denotes goal point, Black are obstacles, Light blue denotes open space. The Search problem formulation has been done for you in file `maze.py`. The maze itself is defined in file `maze_maps.py`.

You are required to find the shortest path from start to goal by implementing:

1. Depth First Search (Marks: 5)
2. Breadth First Search (Marks: 5)

Implement your solution in the file `search.py`. To run the code, execute file `maze.py`. Explain your implementation in a report format with screenshots of the terminal showing results after running the code as well as screen shots of the final path produced by your implementation.

Question 2: (Marks: 10)

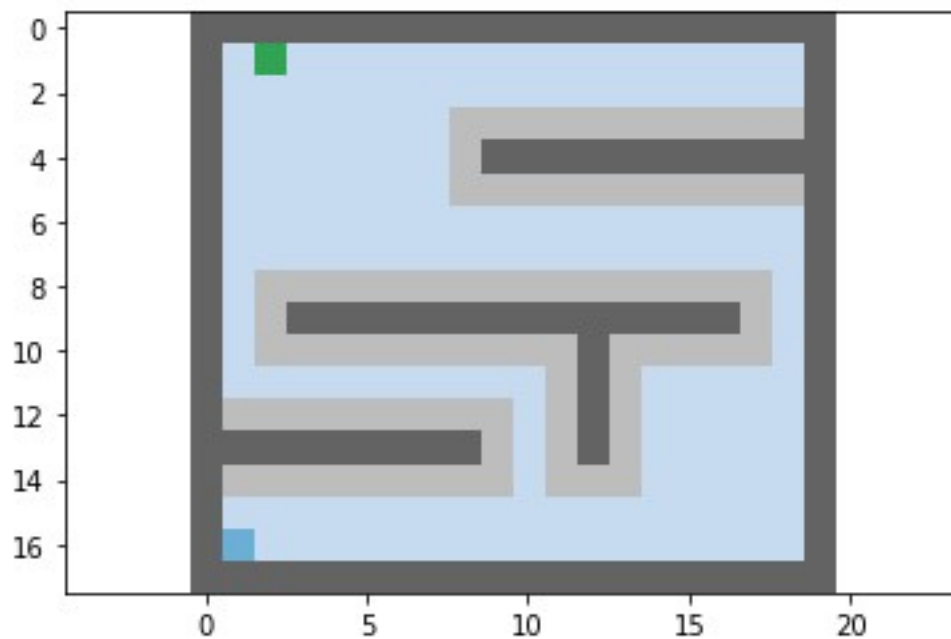


Figure: Maze_2

You are provided with a new maze as shown above. Here, Gray denotes cells that are closer to obstacle and therefore at a higher cost than the cells in Light Blue colour. The Search problem formulation has been done for you in file `maze.py`. The maze itself is defined in file `maze_maps.py`.

To access the new map, change line 123 in the file `maze.py` to

```
current_maze = Maze(2)
```

You are required to find the path of minimum cumulative cost from start to goal by implementing:

1. Uniform Cost Search (Marks: 10)

Implement your solution in the file `search.py`. Explain your implementation in a report format with screenshots of the terminal showing results after running the code as well as screen shots of the final path produced by your implementation.

Question 3: (Marks: 5)

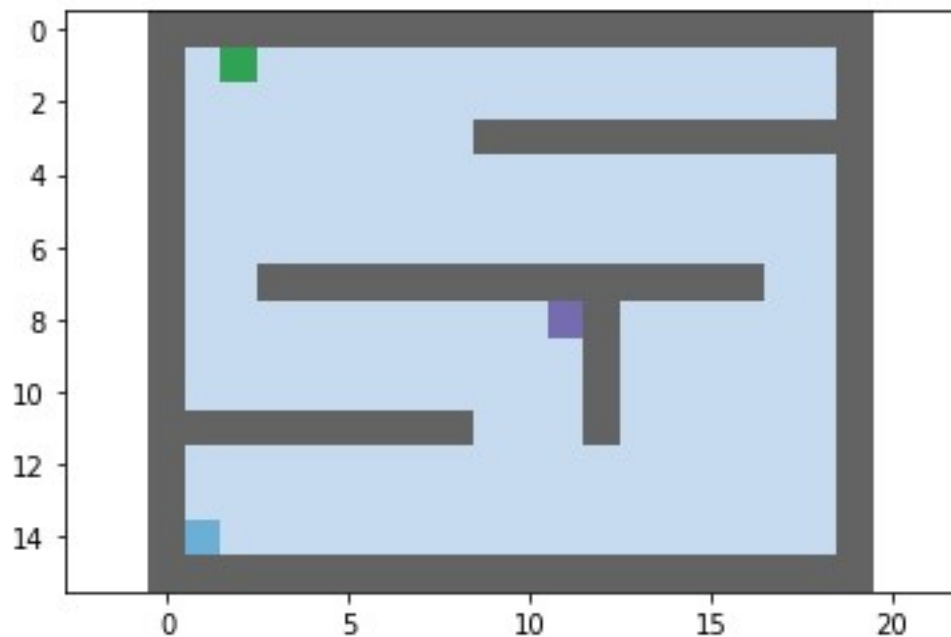


Figure: Maze_3

You are provided with a new maze as shown above. Here, Violet denotes a beacon that you must visit on your path to goal. The Search problem formulation has been done for you in file `maze.py`. The maze itself is defined in file `maze_maps.py`.

To access the new map, change line 123 in the file `maze.py` to

```
current_maze = Maze(3)
```

You are required to find the shortest path from start to goal that also visits the beacon on its way, by using your earlier implementation of Breadth First Search from question 1. To do this you must edit the `getSuccessors(state)` function defined on, line 78 of file `maze.py`.

Rename the file `maze.py` as `maze_q3.py` and implement your changes. Explain your implementation in a report format with screenshots of the terminal showing results after running the code as well as screen shots of the final path produced by your implementation.

Submission details:

Put your implementation of search.py, maze_q2.py and the report together in a zip file and upload to Moodle.

Note:

1. You are not allowed to use any python packages other than the ones already defined in the files. Instructions:
2. Type the report explaining your solution in Word or Latex and upload in PDF format.
3. Do not upload pictures of handwritten notes.
4. -1 marks per day for late submission.