CS 580: Introduction to Artificial Intelligence

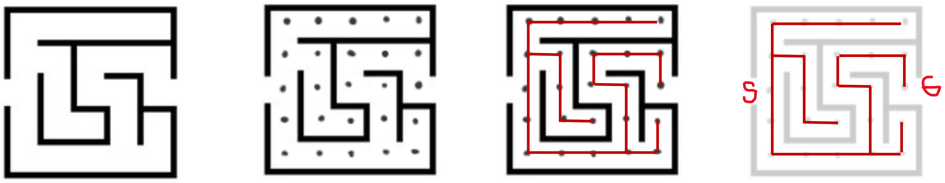
HW 1: Problem Solving as Search

Due: Sun, Feb 12 at 11:59 pm ET

Note: This homework is considered individual effort and the code of honor is applied when reviewing the implementation.

Introduction

A maze can be represented by a graph if each junction is considered as a vertex and a line is drawn to join those that are not blocked by a wall. See the example:



1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	**3,5**
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,5

After transforming the maze into a tree, we could implement A* and evaluate the cost of each node to select the one with the lowest cost.

- **Start**: Coordinate (3,1)
- **Goal**: Coordinate (3,5)

Instructions

In this homework you will implement **A*** in Python to solve a maze. To select a node, **A*** uses the minimum cost which is calculated by **f(n) = g(n) + h(n)**, where:

- **g(n)** is the cost to reach to reach the node (**n**) *from the start node*.
- **h(n)** is the heuristic cost to reach the goal node *from the current node (**n**)*.

For this implementation:

- **g(n)** is the Euclidean distance from the start **node** to **n**
- **h(n)** is the Manhattan distance from **n** to the **goal**, $d(p,q) = |x_1 - x_2| + |y_1 - y_2|$
- The starting point will always be the lower right corner and the goal the upper left corner.
- The size of the maze can vary from [2,100]

Implementation

Install the pyamaze module from https://pypi.org/project/pyamaze/ to visualize the maze.
In case you have issues installing the module, just copy the `pyamaze.py` file in the same working path of your implementation, for example, in the same folder you will have:

```
h1_AstarMaze_username.py
pyamaze.py
```

Include the next code in your `h1.py` file to import and visualize the maze:

```
from pyamaze import maze, agent, textLabel
#rows and cols must are variables, so factual values are needed to avoid errors
m = maze(rows,cols)
m.CreateMaze() #create a maze with one path
a = agent(m, footprints = True)
m.tracePath({a:m.path})
l=textLabel(m,'Path Length',len(m.path)+1) #display the cost of the solution
m.run()
```

To verify that everything is set, just run `h1.py` with this code and replace rows, cols with 10.

A* can be implemented as a Priority Queue, Python already has a module for this purpose, this module can be imported with the following line:

Here is a useful link explaining the main methods of the module:
https://www.linode.com/docs/guides/python-priority-queue/

Submission

Submit your solution as `h1_AstarMaze_<username>.py`, where `<username>` is your Mason account. Once the link for submission is closed, we do not accept resubmissions, so it is the responsibility of the student to verify that their file is the correct one and it can be extracted (i.e. it is not a corrupted file).