Artificial Intelligence

COMS 4701 – Fall 2014

`http://www1.ccls.columbia.edu/~ansaf/4701/`

Home Work n°1: Search Agents

Due Sunday September 28^{th}, 2014 @11:59pm

Problem 1: Robotic Path Planning

The aim of this homework is to build a search agent for a robotic path planning. You will be implementing and comparing several search algorithms and evaluating their performances.

More specifically, we would like to build *Mark1*, a search agent robot for our AI class. We want Mark1 to be able to navigate in a given map with obstacles.

The first important task of the robot is assigned to you. To make crafting this first prototype easier and manageable, the robot leader has allowed for various assumptions to hold in the map and the robot. These are described as follows:

Assumptions

1. The map is a rectangular arena of size $m \times n$ which is bounded by walls on the four sides.
2. An obstacle in the map is marked by an "o", and an empty positions is marked by " " (blank space) in the map.
3. The robot always starts from one exact position marked "s" in the map.
4. The robot has to reach one and only one goal position marked "g" in the map.
5. The robot is allowed to move only in one of the four directions (UP, DOWN, LEFT and RIGHT), one move at a time.
6. The cost of moving from one point to any neighbour is the same and equal to one. Thus, the total cost of path is equal to the number of moves made from start to the goal position.
7. Though the robot is autonomous, it is assumed that the robot is always aware of the complete map and its current location w.r.t to the map at anytime.

Example

m = 7, n = 9

Cost of path = 12 (in the below case), the path is shown with "*"

$M =$

		g					
		o	o	o	o	o	
	o						
		o	s			o	

$\Longrightarrow$

		g	*	*	*	*	*
		o	o	o	o	o	*
	o				*	*	*
		o	s	*	*	o	

Questions

1. (25 points) Given the map M, implement a **Breadth-First Search** to obtain a path from the start to the goal position.
2. (25 points) Given the map M, implement a **Depth-First Search** strategy to obtain a path from the start to the goal position.
3. (25 points) Given the map M, implement an **A* Search** to obtain a path from start to the goal position.
4. (15 point) Summarise and compare the results of the above three approaches w.r.t. to:
   (a) cost of the path (exact numbers),
   (b) memory requirements of each approach (exact numbers/general view), and
   (c) running time (exact numbers).
5. (5 points) Which of the above methods will fail / not work effectively if the goal position is not given right at the start of the algorithm?
6. (5 point) Name one algorithm that can be applied if the cost for moving from one position to the neighbouring is not all equal. You can view the problem as a graph problem with positions as vertices and costs as weights on edges between the vertices.
7. (25 points) **(Optional)** Implement an Iterative Deepening A* search (IDA*) and compare it the other three methods implemented above.

Files provided

```
hw1_UNI.py                - Template Python file - Python 2.7.x ONLY allowed
arena1.txt                - Input arena map
hw1_output_example.txt    - Example output format to submit (the paths and costs are examples)
```

arena1.txt
It contains the arena map as discussed earlier, it has 'm' lines each with 'n' characters (either blank space / g / s / o)
Run the following commands to understand the template code for this HW1.

```
>> python hw1_UNI.py -h
>> python hw1_UNI.py -all -m arena1.txt
```

Files to submit

```
hw1_UNI.py           - Completed Python code
hw1_output1.txt      - Output of arena1.txt
hw1_output2.txt      - Output of arena2.txt (surprise map given 1-3 days before the deadline)
hw1_output3.txt      - Output of arena3.txt (surprise map given 1-3 days before the deadline)
hw1_report_UNI.pdf   - Completed PDF Report. Use of LaTeX is optional but preferred.
```

The file hw1_output1.txt is expected to contain the following: solution to each algorithm with the cost of the path, in the following order BFS, DFS and A* search (e.g., see hw1_output_example.txt)