Cairo University
Faculty of Engineering
Computer Engineering Department

CMP (N)402 Machine Intelligence Fall 2019

Python – Lab 1

Description:

- Squirrel is a game where you play as a squirrel collecting nuts and hiding them in stashes where every stash can only hold one nut.
- The input will be a 2D grid of characters where each character can denote the ground '.', a wall '#', the player '@', a nut 'N' or a stash 'X'.
- you can move in one of the four main direction ("UP", "DOWN", "LEFT", "RIGHT") to an adjacent cell on the grid as long as it is not a wall.
- If the you stand in a cell containing a nut, you can "TAKE" it with you (the nut will be removed from the grid and added to your inventory).
- If you have at least one nut and stand in a cell containing a stash, you can "HIDE" it in that stash (the stash will be removed from the grid and one nut will be removed from you inventory).
- Once you take a nut, you can no longer leave anywhere except in a stash. Also, once you hide a nut in a stash, you cannot use the same stash again nor can you take the nut back from the stash.
- The goal is to reach a state where every nut is hidden in a stash (no nut on the grid and no nut in your inventory).

Requirement:

1. Define the problem of Squirrel such that it can be solved by the supplied Breadth-First-Search Solver. The problem must inherit the "Problem" class. The only file you are allowed to modify is "squirrel.py". The output for all the test cases should match the expected output for BFS.

Note:

To run the lab code, you need python to be defined in the PATH environment variable since you have to call it from the command line (or terminal on Linux).
For example, to run test-1 in the maze problem, execute:

```
python solve.py maze.Maze search.bfs maze_tests/test1.txt
```

Similarly, to run test-1 in the squirrel problem, you should execute:

```
python solve.py squirrel.Squirrel search.bfs squirrel_tests/test1.txt
```

NOTE: PLEASE READ AND UNDERSTAND THE CODE BEFORE YOU START WRITING YOUR SOLUTION.

Cairo University
Faculty of Engineering
Computer Engineering Department

CMP (N)402 Machine Intelligence Fall 2019
Python – Assignment 3

Description:

- This assignment continues after Lab 1 (Squirrel Game).
- In this game, each action has a cost that depends on the amount of nuts you have in your inventory.
- Both “TAKE” and “HIDE” action has a cost of 1.
- However, The movement actions () has a cost of one only you have no nuts in your inventory. Each nut adds more cost to your movements; the first nut add 1, the second adds 2 and so on.
- For example, if you move one cell without any nuts, the cost is 1. If you have 1 nut, the cost is 1+1=2, if you have 2 nuts, it will be 1+1+2=4 and for 3 nuts, it will be 1+1+2+3=7 and so on. You can think of it as 1 plus an arithmetic series.

Requirement:

2. Change the squirrel problem such that the action cost follows the description above.
3. Design and implement a heuristic for this problem. The heuristic must be admissible and consistent. The better your heuristic is, the faster the A* will be. Your grade will depend on how good your heuristic will be.
4. Implement Uniform Cost Search, A* search and Greedy Best First Search. They must work with any problem that extends the “Problem” class.

Note:

While you must get the same cost as the expected output for Uniform Cost Search and A* and the same number of steps for Breadth first. Your answers could differ in the Greedy Best First. As for the number of "get_successor" calls, it can also not match the given values, but it should never be larger for A* than Uniform Cost Search and it should never be larger for Greedy Best first than A*.