

Homework 4 - Markov Decision Process

-- Course: *Intelligent Robotics* – Professor: *Qi Hao*

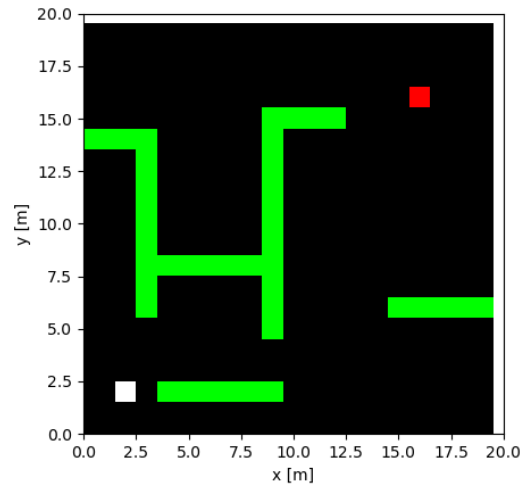
Coding Homeworks. Most of coding assignments will be done by Python(≥ 3.5) under a simple [robotics simulator](#). You can follow the [Coding instruction](#) to use this simulator to complete the coding part in question1-3. Your final submission should be a compressed package with extension .zip, which includes your codes and explanations (you need to know how to write the manuscript with Markdown or LATEX). Your code should be run step-by-step without any error. Real-time animation is also recommended.

Question

Please find the optimal path under a given grid map with reward using Markov Decision Process (MDP)

Note:

- white: the start position
- red: the goal position
- green: the obstacle
- black: ground
- obstacle reward: -10
- goal reward: 10
- others: -1
- over the bound: -5



Coding instruction

Install the intelligent robotics simulator

```
git clone -b edu https://github.com/hanruihua/intelligent-robot-simulator.git
cd intelligent-robot-simulator
pip install -e .
```

Note1: Please confirm that this repository is under the *edu* branch. You can use **git branch** to check current branch. If it is not under the *edu* branch, you can use **git checkout edu** to change current branch to *edu* branch.

Note2: The pycharm reduces the functionality of Matplotlib, which may lead to the failure of saving the gif animation. You can follow this [link](#) to solve this problem

Note3: If you have installed this simulator, you can use *git pull* to fetch the code update.

Code for question

There are five files for this question in the source folder, [question_run.py](#), [mdp.py](#), [grid_map.py](#), [map_matrix.npy](#), and [reward_matrix.npy](#)

- [question_run.py](#) is the main program you should run
- [mdp.py](#) is the file to perform Markov Decision Process. You should complete the functions include value iteration and policy iteration in this file for the coding task.
- [grid_map.py](#) is the file that defines the class about the grid map for you to use.
- [map_matrix.npy](#) and [reward_matrix.npy](#) define the map and the reward in each grid.

You should complete the file [mdp.py](#) and run `question_run.py` to show the simulation results. You can set the parameter `animation = True` in `question_run.py` to generate the animation such as the follows.

