## Introduction to the CS188 codebase

A. Write an MDP class for the grid that the Udacity course uses as its primary example. (It’s also the grid that the CS188 course uses. They call it the “BookGrid.” To refresh your memory about how it looks, here are some [value iteration steps](https://docs.google.com/presentation/d/1TklBwZdeqmIuBZ_y9Zeju3lxDyW-99sJRMm1Oe4tnz0/edit?usp=sharing) on that grid.)

Here is the version [updated version](https://drive.google.com/file/d/1vhGKjrvugYQ3KlVV-6DtFTv289bQyOQr/view?usp=sharing) that include both value iteration and q-value iteration. It also accommodates bith the Sutton Example 3.5 and “BookGrid.” Define your class so that it is easy to change the rewards. (Recall the various reward configurations we used a couple of weeks ago in the [Udacity quiz](https://classroom.udacity.com/courses/ud600/lessons/4100878601/concepts/6512308700923).)

Here are [value iteration and q-value iteration steps](https://drive.google.com/file/d/1rnODV4Yt7pZPMp7R85lRZUTkehx5z3n2/view?usp=sharing) on the two grids.

B. This week we start [Project 3](https://edge.edx.org/courses/course-v1:BerkeleyX+CS188+2018_SP/courseware/817468c509ef4b64bcb74cd38766ac44/41c847d3794141a7acb4f6b1e5904215/?child=first) from Berkeley’s CS188. (This and the following are all on *edX*. You may have to create an account and “sign up” for the course to get access to the material)

* [**Read p3\_rl\_introduction**](https://edge.edx.org/courses/course-v1:BerkeleyX+CS188+2018_SP/courseware/817468c509ef4b64bcb74cd38766ac44/41c847d3794141a7acb4f6b1e5904215/?child=first). The instructions ask you to download a zip archive. That archive is written in Python 2.7. You can use it if you like. I’ve [translated it to Python 3](https://drive.google.com/file/d/1LJxpJNAu2K_FCq9KLAMhgrrCxcjtg18N/view?usp=sharing). That’s the version I use. I recommend that you use it also.
* [**Read p3\_rl\_welcome**](https://edge.edx.org/courses/course-v1:BerkeleyX+CS188+2018_SP/courseware/817468c509ef4b64bcb74cd38766ac44/41c847d3794141a7acb4f6b1e5904215/?child=first). After downloading the code, play around with it. The assignment offers some suggestions. For example if you run

python gridworld.py -m

you will be able to move around on the grid using the arrow keys.

python gridworld.py -h

shows you the available command-line options.

* [**Do the assignment at p3\_rl\_q1\_value\_iteration**](https://edge.edx.org/courses/course-v1:BerkeleyX+CS188+2018_SP/courseware/817468c509ef4b64bcb74cd38766ac44/41c847d3794141a7acb4f6b1e5904215/?child=first)**.** This assignment asks you to write some code. In particular you are asked to write the functions:
  + computeActionFromValues(state) computes the best action according to the value function given by self.values. (We discussed this briefly in class.)

This function corresponds to the *max* function used by Sutton and Charles/Michael. That is, it finds *the best action* to take.

* + computeQValueFromValues(state, action) returns the Q-value of the (state, action) pair given by the value function given by self.values.

Given a (state, action) pair, its Q-value is the value resulting from taking action when in state.

This corresponds to Sutton’s equation (3.13) on p. 58.

As the *edX* assignment page suggests, if your code works properly, when you run

python gridworld.py -a value -i 5

you should get [these results](https://docs.google.com/presentation/d/1TklBwZdeqmIuBZ_y9Zeju3lxDyW-99sJRMm1Oe4tnz0/edit?usp=sharing).