Homework 1 (Dynamic Programming)

(8 % of total grade)

1. Install Anaconda(or Minoconda)

1) Visit https://docs.conda.io/en/latest/miniconda.html and choose the Miniconda install for your platform to install Miniconda. Please choose the latest Python3.x version. If you already have Anaconda or Miniconda installed, you can skip this step.

2) We will create a new environment to run the code accompanying this book. Open a command terminal and type the following:

```
conda create -n rl_env
```

where rl_env is the name of the environment and you can change it to your own choice. Answer yes to all the prompts.

3) Switch to the new environment you created using the following:

```
conda activate rl_env
```

2. Create Gridworld environment. You define a grid world such as

start(0)	1	2	3	4	5	6
7	8	9	10	...		
	block					
					block	
block						
			final			

- must have one(or multiple) start state(s) and one(or multiple) final state(s).
- the size of column or row >5
- must have a few blocks(obstacles)
- moving out of the boundary or moving to a block cell stays in the current state.

1) Show the picture of your own Grid world including start/final state, blocks.

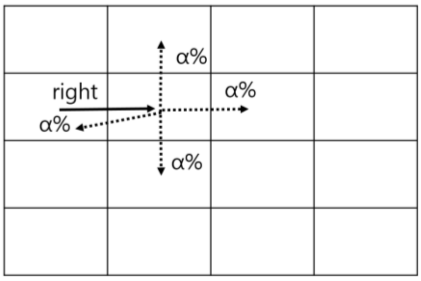
3. Complete the program code (refer to code.py file).

1) In (A)

Define your grid environment. Specifically, you have to define transition prob. and rewards.

1-1) deterministic transition: agent moves to its intended state with 100%. Complete transition prob array T

1-2) probabilistic transition: In probabilistic transition, and after taking an action, with prob α (α <=0.05), it moves to its neighboring state as shown below. Agent moves to its intended state with 1-4*α prob.



Complete transition prob array T

1-3) You can assume each move generates -1 reward. Also define discount factor(gamma) (e.g., 0.9)

2) In (B),

Suppose we are using random(uniform) policy.

2-1) initialize policy array with uniform policy

2-2) implement prediction (policy evaluation) algorithm in slide page 7.

2-3) implement policy improvement using greedy method. Greedy method of choosing action with V value is as follows.

$$\pi(a|s) = \underset{a}{\operatorname{argmax}} \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

2-4) implement policy iteration

2-5) implement value iteration

4. Run programs

4-1) run programs in deterministic transition

4-1-1) given a uniform policy, show the results of policy_eval.

4-1-2) run policy iteration and show the results

4-1-3) run value iteration and show the results. After that extract policy from v values.

4-2) repeat 4-1 with probabilistic transition