CS 540: Introduction to Artificial Intelligence
Homework Assignment # 11

Assigned: 4/26
Due: No due; served as exercise

Hand in your homework:

If a homework has programming questions, please hand in the Java program. If a homework has written questions, please hand in a PDF file. Regardless, please zip all your files into hwX.zip where X is the homework number. Go to UW Canvas, choose your CS540 course, choose Assignment, click on Homework X: this is where you submit your zip file.

Late Policy:

All assignments are due at the beginning of class on the due date. One (1) day late, defined as a 24-hour period from the deadline (weekday or weekend), will result in 10% of the total points for the assignment deducted. So, for example, if a 100-point assignment is due on a Wednesday 9:30 a.m., and it is handed in between Wednesday 9:30 a.m. and Thursday 9:30 a.m., 10 points will be deducted. Two (2) days late, 25% off; three (3) days late, 50% off. No homework can be turned in more than three (3) days late. Written questions and program submission have the same deadline.

Assignment grading questions must be raised with the instructor within one week after the assignment is returned.

Collaboration Policy:

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas with classmates, TAs, and instructor in order to help you answer the questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers
- not to copy answers or code fragments from anyone or anywhere
- not to allow your answers to be copied
- not to get any code on the Web

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we suggest that you specifically record on the assignment the names of the people you were in discussion with.

Question 1: MDP

Consider state space $S = \{s_1, s_2\}$ and action space $A = \{left, right\}$.

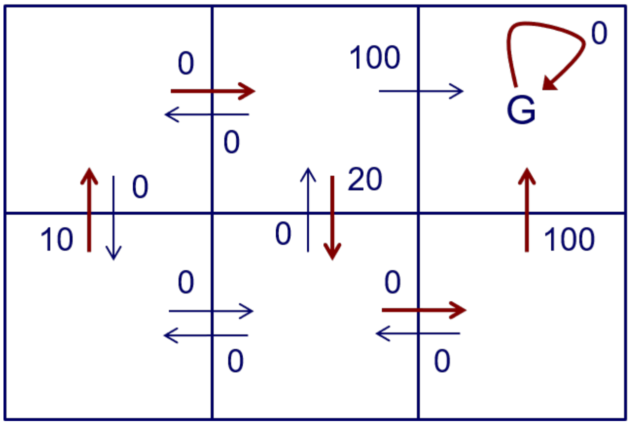
s_1	s_2

In s_1 the action "right" sends the agent to s_2 and collects reward $r = 1$. In s_2 the action "left" sends the agent to s_1 but with zero reward. All other state-action pairs stay in that state with zero reward. With discounting factor γ, what is the value $v(s_2)$ under the optimal policy?
Answer: The optimal policy is $\pi(s_2) = left, \pi(s_1) = right$.

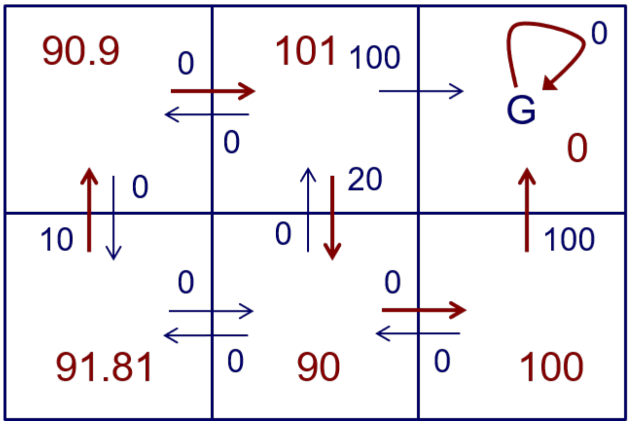
$$v(s_2) = 0 + \gamma \cdot 1 + \gamma^2 \cdot 0 + \gamma^3 \cdot 1 + \ldots = \gamma + \gamma^3 + \gamma^5 = \gamma/(1\gamma^2).$$

Question 2: Value function

Suppose a policy π is shown by red arrows, the discount factor $\gamma = 0.9$. Compute the value function $V^\pi(s)$ for all states s.



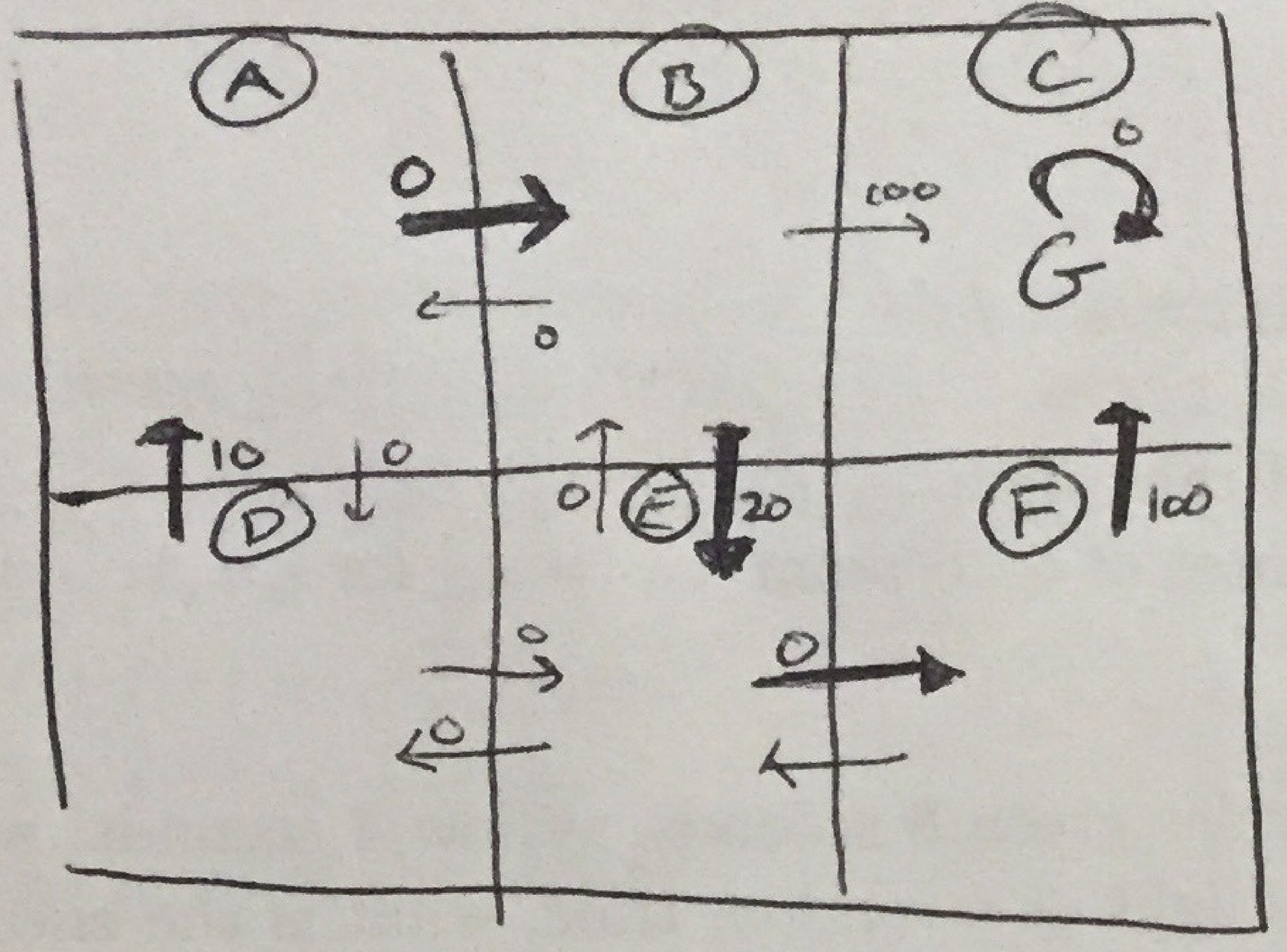
Answer:



Question 3: Q-learning

A robot initializes Q-learning by setting $q(s, a) = 0$ for all state s and action a. It has a learning rate α, and discounting factor γ. The robot senses that it is in state s_{105} and decides to performs action a_{540}. For this action, the robot receives reward 100 and arrives at state s_{7331}. What value is $q(s_{105}, a_{540})$ after this one step of Q-learning?

Answer: $q(s_{105}, a_{540}) = \alpha \cdot 100$. All other things zero out.



Bold arrows are those taken by the policy (π)

$\gamma = .9$

$V^\pi(s) = \sum_0^t \gamma^t \cdot E(r_t)$

t = a state that will be reached by s via an arrow in π

~~Alright~~

easiest to work backwards from the goal

C: can only reach itself with one action, which has a reward of 0.

I labelled these in a bad unintuitive way. ↓ sorry

$V^\pi(c) = 0.$

F: $V^\pi(F) = (.9)^0(100)$ ← accessible from this state

$= 100$

E: $V^\pi(E) = (.9)^0(0) + (100)(.9)^1$

$= 90$

B: $V^\pi(B) = (.9)^0(20) + (.9)^1(0) + (100)(.9)^2$

$= 81 + 20 = 101$

A: $V^\pi(A) = (101)(.9)^1 = 90.9$

↖ lazy, but the same as adding up all successor states down the line

D: $V^\pi(D) = \cancel{101}\ (90.9)(.9)^1 + (10)(.9)^0 = 91.81$