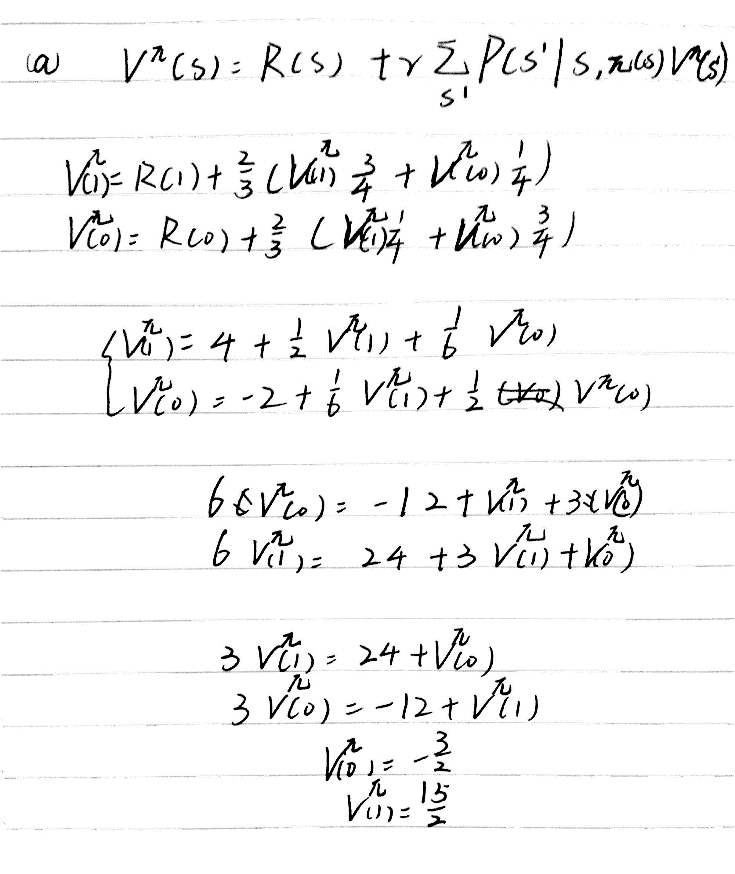
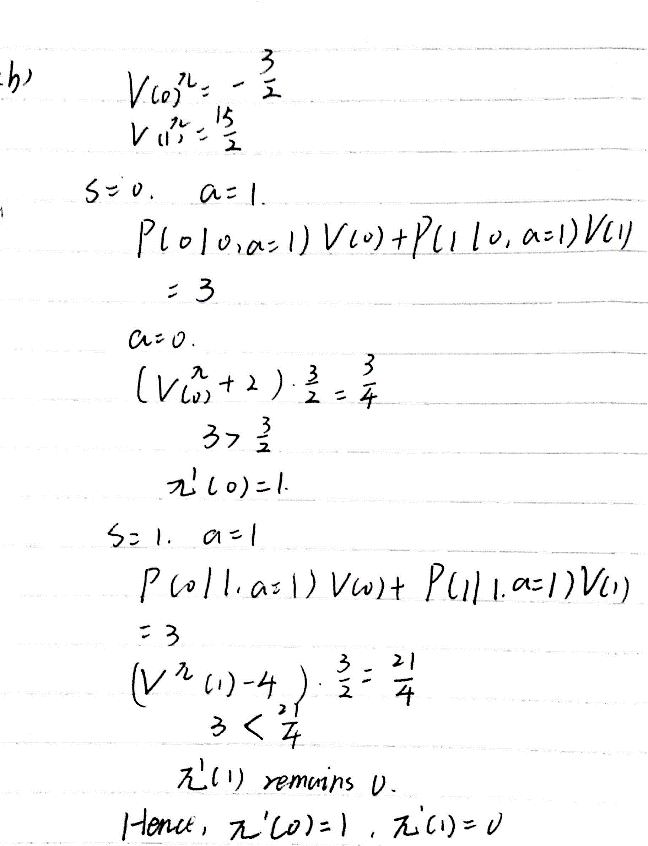
**CSE150 PA5 Report**

**Yunfan Kang A14282934**

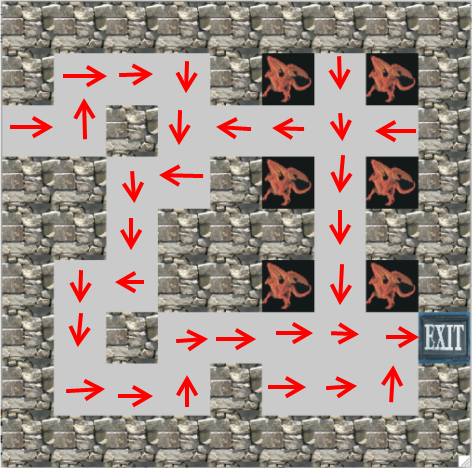
1(a)



b.



2.



For problem 2, I wrote a file named MDP.py to give the definition of MDP class. I set the transitionFunction as a defaultdict(list) with keys being (s, action) and values being lists of (s’, action). Object mdp is initialized in both value\_iteration.py and policy\_iteration.py with the data in the files provided. The following description will only focus on the functions.

1. Value iteration:

Create two lists U and U2, both of size 81, and initialize all their entries to be 0.

Set maxChange = 0

While True:

Copy the entries of U2 to U.

Set maxChange = 0

For every state s of the mdp:

For every action:

Get all (s2, prob) pairs through transitionFunction[(s, action)], calculate prob\*U[s2] and put the sum into a list listPU. Finally assign R(S) + discountFactor\*max(listPU) to U[s].

If |U2[s] – U[s]| > maxChange:

maxChange = |U2[s] – U[s]|

if maxChange == 0, break the while loop (Since on piazza TA respondes “stop iterating when delta = 0 for all states”).

Return U.

b.

After calculating U, go through each state in mdp.states and deter the action greedily by always choosing the action that leads to the maximum utility for s and store it to a dictionary pi with s being the key.

Print all (s, U[s], pi[s]) pairs where U[s] > 0.

c.

First implement policy\_evaluation(pi, mdp):

Since everytime we call this function the policy is given as a parameter, the set of equations become linear and can be solved by calling numpy.solve(). This function will return a list U with U[s] being the evaluated utility for state s.

policy\_iteration(mdp):

Generate a random policy pi vector indexed by state.

Initialize U as an empty.

While true:

U = policy\_evaluation(pi, mdp)

Set unchanged = True

Like in value\_iteration, pick the action that maximizes the value of the Bellman equation, store it as maxAction and compare it with pi[s], if maxAction is better, pi[s] = maxAction and set unchanged = False

If unchanged == True, break the while loop.

Return (pi, U)

d.

This assignment is done individually.