Homework 3: Blackjack

Cheng Zhong 16307110259

1.(a) The $V_{opt}(s)$ in 0,1,2 iterations are as follows:

Iteration	-2	-1	0	1	2
0	0	0	0	0	0
1	0	15	-5	26.5	100
2	0	14	13.45	23	0

(b) So, the resulting optimal policy for all non-terminal states is

	-2	-1	0	1	2
Action	\	-1	+1	+1	\

2.(a) Consider this kind of MDP:

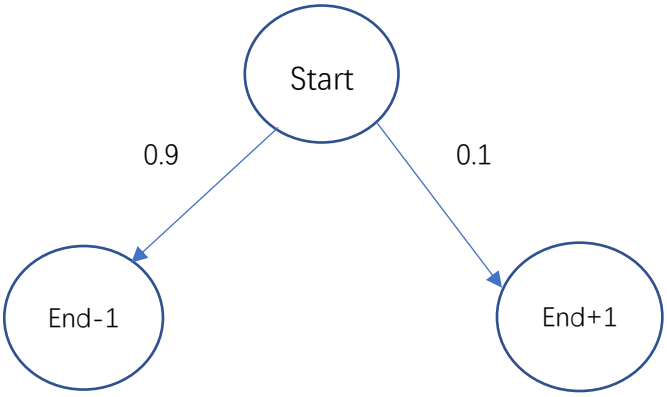

Figure 1 The Counter example MDP

We begin at "Start" node, which have 90% probability to get to "End-1" node and get reward -1, also, it can get to node "End-1" with probability 10% and reward +1. With discount=1, we can compute that $V_1(S_{start}) = -0.8$. If we add noise, we will have more probability to get to node "End+1" and get higher $V(S_{start})$, we can compute that $V_2(S_{start}) = -0.4$. In this situation, $V_2(S_{start}) > V_1(S_{start})$

(b) If the MDP is acyclic, we can simplify the entire MDP model into a tree by regarding the state as nodes and transition method as edges, and then use the recursive method to calculate the node from top to bottom. So that we can get the V_{opt} that only requires one pass over all edges (the (s,a,s') triples).

(c) we can define the new MDP as:

$$\begin{aligned} T'(s,a,s') &= \gamma T(s,a,s') \\ T'(s,a,o) &= 1-\gamma \\ \text{Reward}'(s,a,s') &= \frac{1}{\gamma}\text{Reward}(s,a,s') \\ \text{Reward}'(s,a,o) &= 0 \end{aligned}$$

4.(b)

It can be seen from the results of the program that Q-learning's strategy is not completely consistent with the result of Value iteration. Q-Learning does better on smallMDP and worse on largeMDP. Also, value iteration policy on largeMDP also takes more iterations to convergence

This is because the Q-Learning algorithm uses greedy strategies to generate policy, also, our feature extractor cannot describe the exact value at each state, which will cause the Q-Learning algorithm can not learn the Q-value more precisely in large state.

```
----- START PART 4b-helper: Helper function to run Q-learning simulations for question 4b.
ValueIteration: 5 iterations
MDP accuracy is:  0.7407407407407407
ValueIteration: 15 iterations
MDP accuracy is:  0.668488160291439
----- END PART 4b-helper [took 0:00:04.862118 (max allowed 60 seconds), 0/0 points]
```

Figure 2 The result of problem 4b (Small MDP result is above and the Large MDP is below)

4.(d)

It can be seen from the results of the program that we get lower rewards in FixedRLAlgorithm because the policy we get from FixedRLAlgorithm (value iteration) is from original MDP and it can not adapt for a new MDP problem.

However, if we run Q-Learning on new MDP problem, we can get higher reward because it can adapt to new MDP problems.

```
----- START PART 4d-helper: Helper function to compare rewards when simulating RL over two different MDPs in question 4d.
ValueIteration: 5 iterations
Rewards for value iteration:  [6, 6, 6, 6, 7, 10, 7, 7, 6, 7]
Rewards for Q-learning iteration:  [12, 12, 10, 12, 11, 12, 11, 11, 12, 5]
----- END PART 4d-helper [took 0:00:00.002001 (max allowed 60 seconds), 0/0 points]
```

Figure 3 The result of problem 4d