Assignment 5 Report for Part A
Billy Lin

0a. $Q^*(s,a) = \Sigma_{s'} T(s,a,s') [R(s,a,s') + \gamma V^*(s')]$
$V^*(s) = \max_a Q^*(s,a)$

0b. $Q_{k+1}(s,a) \leftarrow \Sigma_{s'} T(s,a,s') [R(s,a,s') + \gamma V_k(s')]$
$V_{k+1}(s) \leftarrow \max_a Q_k(s,a)$

1a.
How many iterations of VI are required to turn 1/3 of the states green? (i.e., get their expected utility values to 100).
4 iterations

1b.
How many iterations of VI are required to get all the states, including the start state, to 100?
8 iterations

1c.
From the Value Iteration menu, select "Show Policy from VI". (The policy at each state is indicated by the outgoing red arrowhead. If the suggested action is illegal, there could still be a legal state transition due to noise, but the action could also result in no change of state.) Describe this policy. Is it a good policy? Explain.
With no discounting, no living reward, and no noise, the policy is determined purely randomly. Every state has the same value, so there is no way of comparing the successors of the current state. The policy generated also includes some illegal moves that result in no-ops. Therefore, this policy is not optimal and thus not a good policy.

2a.
How many iterations are required for the start state to receive a nonzero value.
8 iterations

2c.
At this point, view the policy from VI as before. Is it a good policy? Explain.
Now with noise, the value for each state becomes comparable as the transitional function gives unequal distribution to the rewards of the actions. With the unequal values, we can determine the optimal policy with by choosing the max values among the successors. Therefore, this is a good policy.

2d.
Run additional VI steps to find out how many iterations are required for VI to converge. How many is it?
56 iterations

2e.

After convergence, examine the computed best policy once again. Has it changed? If so, how? If not, why not? Explain.

No, the policy didn't change because the policy has converged before the value converges.

3a.

3a. Run Value Iteration until convergence. What does the policy indicate? What value does the start state have? (start state value should be 0.82)

The policy prioritizes exiting at the goal state with R=10 because the discounting factor is too high to spend many actions to get to the other goal state with R=100.

3b.

Reset the values to 0, change the discount to 0.9 and rerun Value Iteration until convergence. What does the policy indicate now? What value does the start state have? (start state value should be 36.9)

The policy avoids exiting at the goal state with R=10 because the discounting factor is low, so it's worth it to spend more actions to get to the goal state with R=100. Also, since the start value is already larger than the goal R=10, the policy will not intend to go to that goal state.

Iteration	Go off plan	Arrive in goal	Away from goal	Parts not visited
1	0	1	0	Upper space
2	0	1	0	
3	0	1	0	
4	1	1	0	
5	1	1	0	
6	1	1	0	
7	1	0	1	
8	1	1	0	
9	1	1	0	
10	0	1	0	

4a.

In how many of these simulation runs did the agent ever go off the plan?

6

4b.

In how many of these simulation runs did the agent arrive in the goal state (at the end of the golden path)?

9

4c.

For each run in which the agent did not make it to the goal in 10 steps, how many steps away from the goal was it?

1 step away

4d.

Are there parts of the state space that seemed never to be visited by the agent? If so, where (roughly)?

The upper part of the state space was never visited

5a.

Since it is having a good policy that is most important to the agent, is it essential that the values of the states have converged?

No, the policy will converge before the values converge, so there's no need to wait for the values to converge before determining the policy.

5b.

If the agent were to have to learn the values of states by exploring the space, rather than computing with the Value Iteration algorithm, and if getting accurate values requires re-visiting states a lot, how important would it be that all states be visited a lot?

Not important because we can always prioritize visiting the states that can give us larger rewards. If we learn a value for a state that has reward lower than the current state, we don't need to explore that state and its successors because we know they will end up having lower rewards.