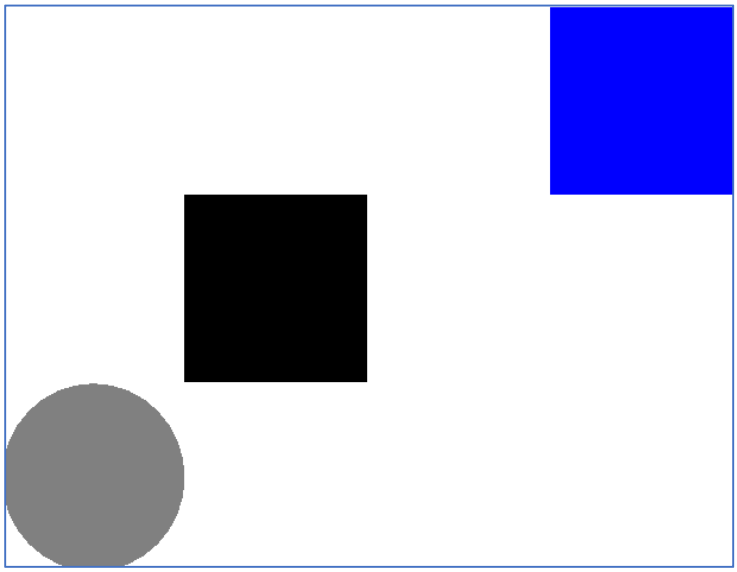
Assignment 4 Analysis **gth836x; 901929700**
Myles Lefkovitz **mlefkovitz@gatech.edu**

MDP Selection

Small MDP

I selected a simplified gridworld based on the example shown in lecture for my first MDP. An image is provided below. In the image, the gray circle represents the agent's start position, the black square represents the wall, and the blue square represents the terminal/reward state.



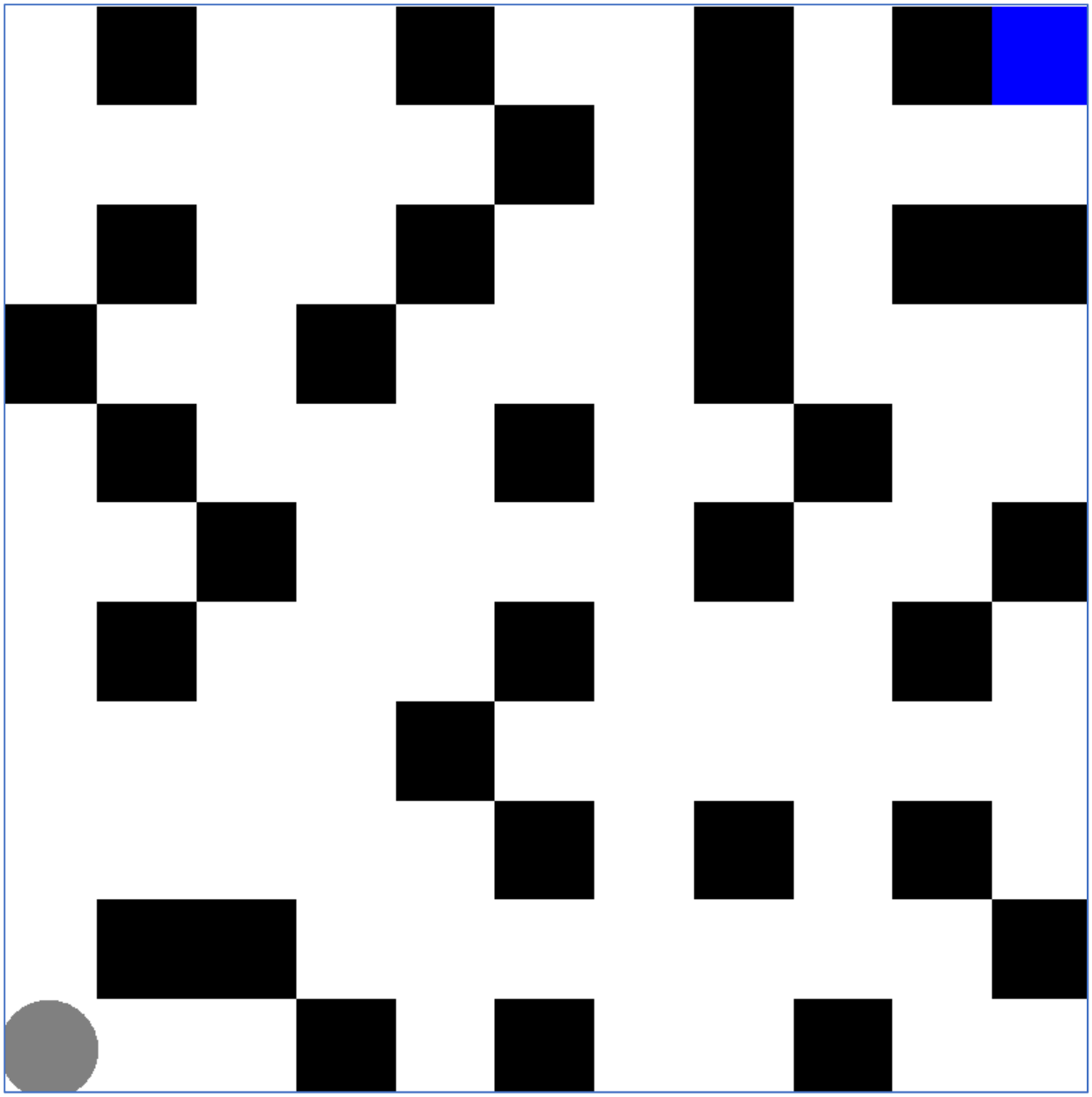
Like the example in class, the gridworld has 4 columns and 3 rows with only one space used as a wall. Unlike the example in class, there is only a single terminal/reward state in this gridworld (same location as the positive reward state in the lecture example).

This gridworld example is interesting because it allows me to directly extend my understanding from the lessons taught in lecture. It's one of the simplest examples of a gridworld, but still allows for probabilistic movement, includes a wall, and has all of its reward function based on a single terminal state.

Transitions in the MDP are defined probabilistically. If an agent intends to move in a given direction, there is an 80% chance that movement is made in that direction and a 20% chance that movement is made (or attempted, in the presence of a wall) in one of the 3 other cardinal directions.

Large MDP

I selected a gridworld maze for my large MDP. The hope is that an algorithm will be able to find the optimal policy to reaching the maze's end. An image is provided below. In the image, the gray circle represents the agent's start position, the black squares represent the walls of the maze, and the blue square represents the terminal/reward state.



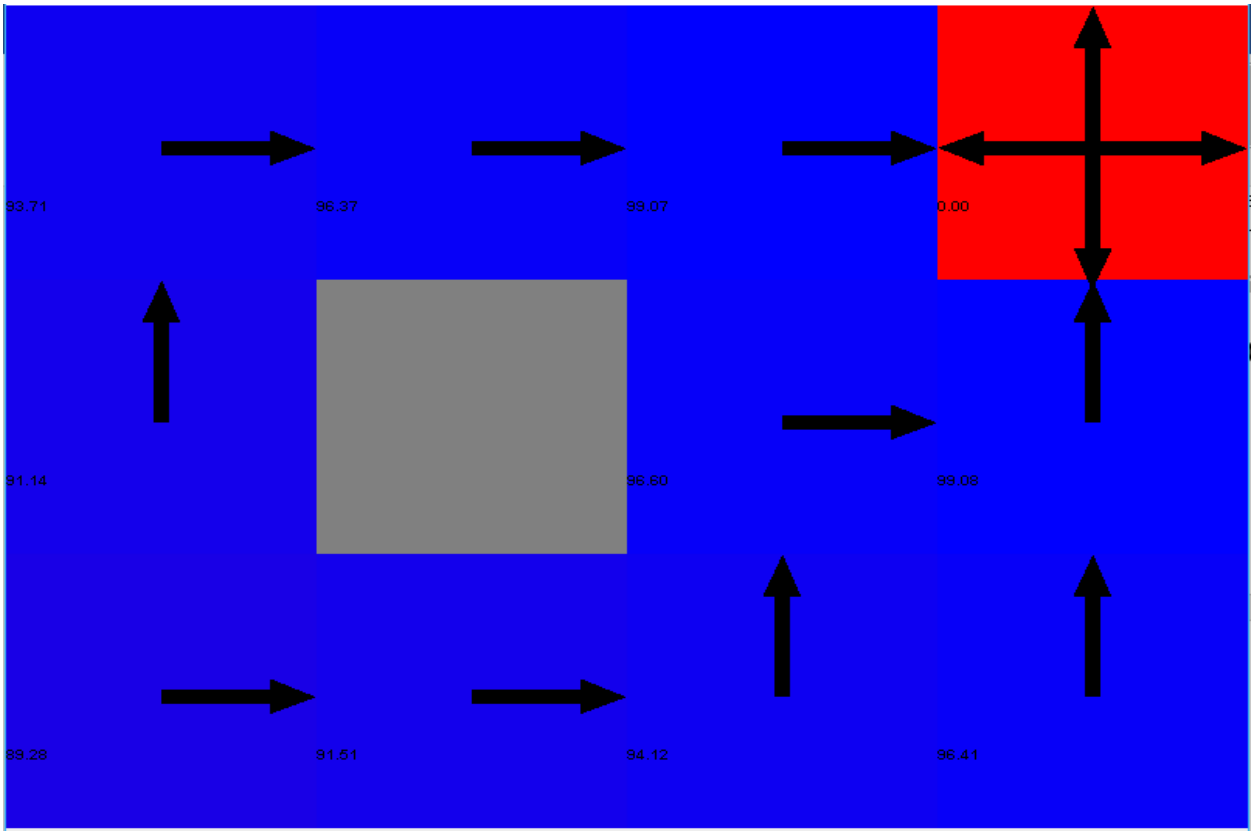
This gridworld is has 11 rows and 11 columns. With many spaces used as walls blocking the agent's path. I created this maze by hand.

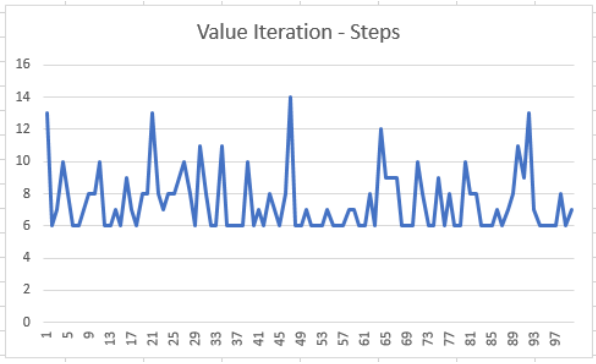
This MDP is interesting because it forces an agent to solve a problem that might not be trivial for a person. This is not a challenging maze, but it still takes a couple of seconds to find the answer by sight. A larger maze could be much more complicated. We could generalize the results in this example to see how different algorithms perform against non-trivial problems.

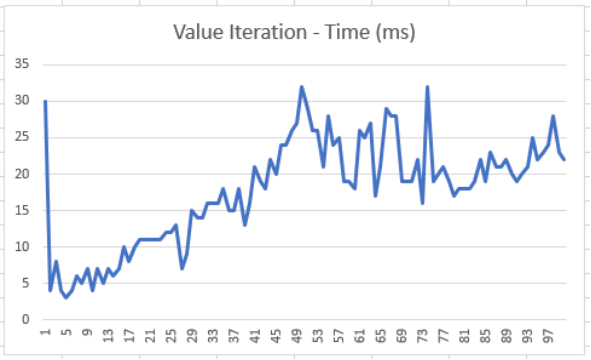
Like in the example above, transitions in the gridworld are defined probabilistically. If an agent intends to move in a given direction, there is an 80% chance that movement is made in that direction and a 20% chance that movement is made (or attempted, in the presence of a wall) in one of the 3 other cardinal directions.

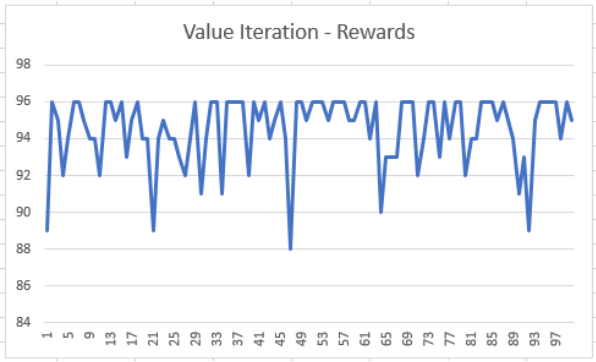
Value Iteration vs Policy Iteration

Small MDP - Value Iteration



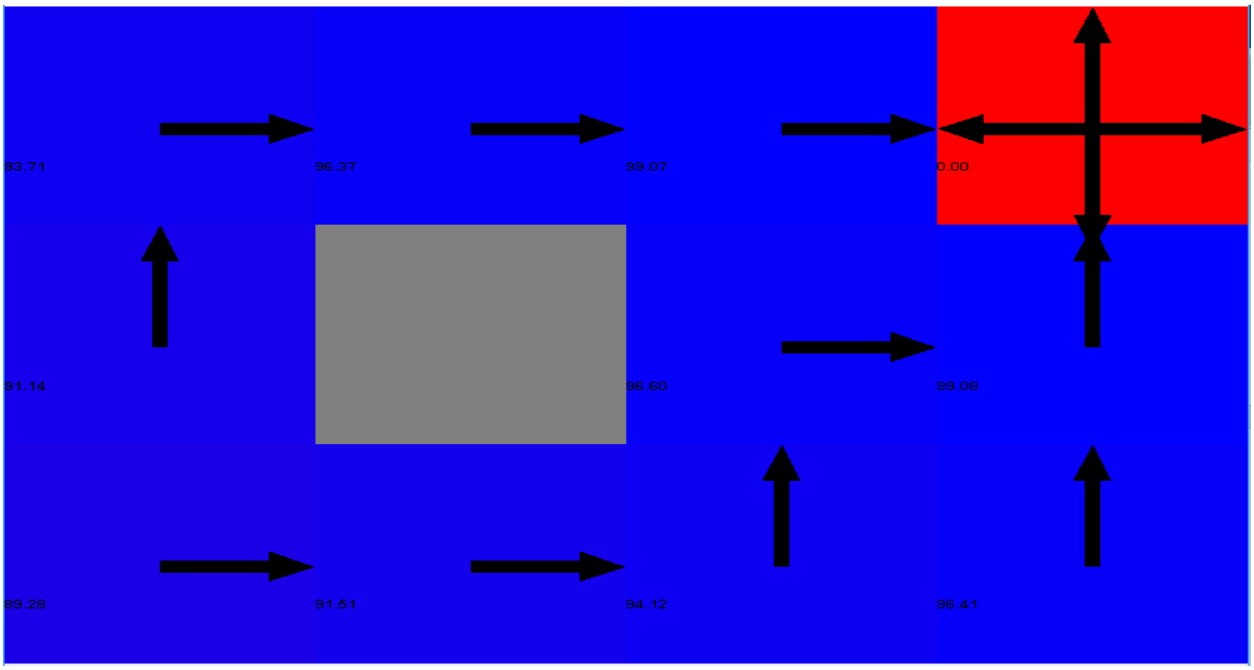


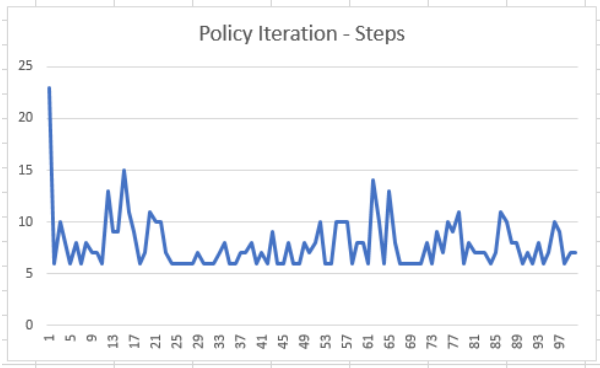


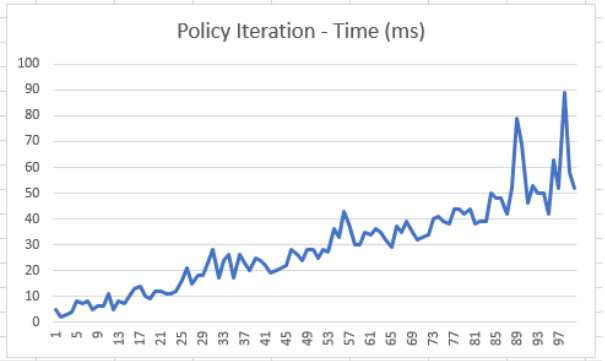


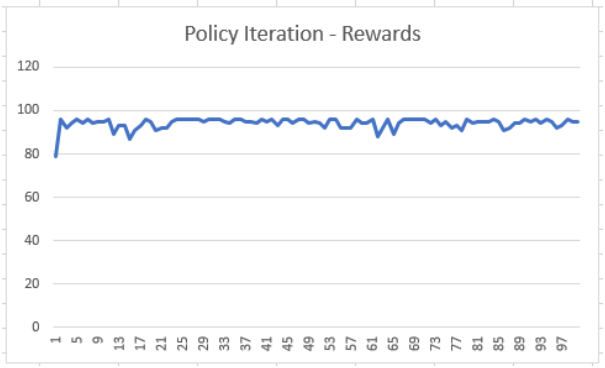
The small MDP takes 6 steps to converge with value iteration. The reward value at convergence is 96. The amount of time necessary to converge varies between 4 and 32 milliseconds.

Small MDP – Policy Iteration



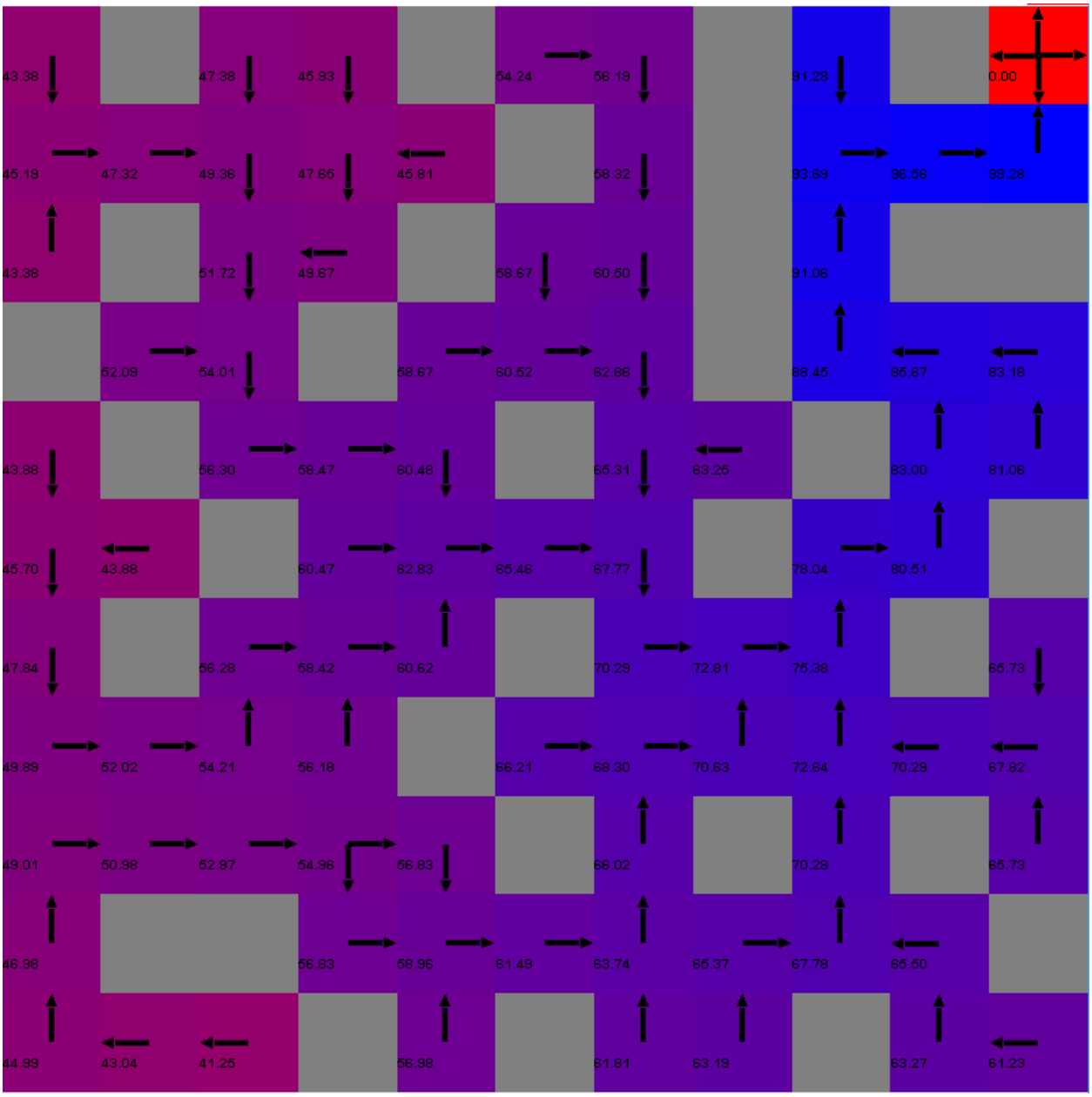


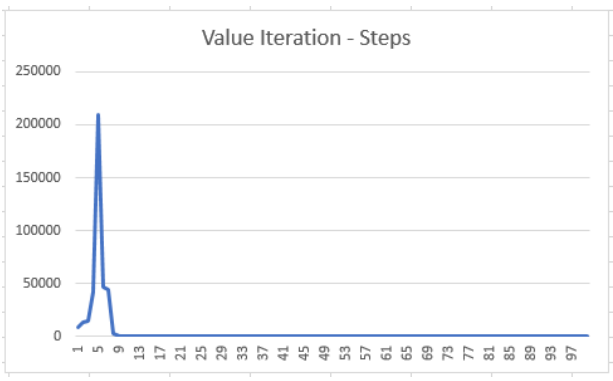


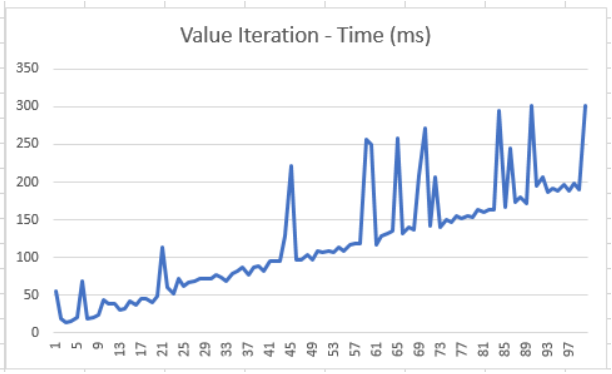


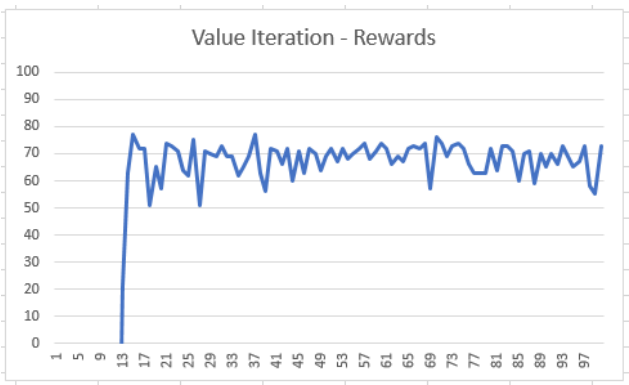
The small MDP takes 6 steps to converge with policy iteration. The reward value at convergence is 96. The amount of time necessary to converge varies between 2 and 89 milliseconds.

Large MDP – Value Iteration



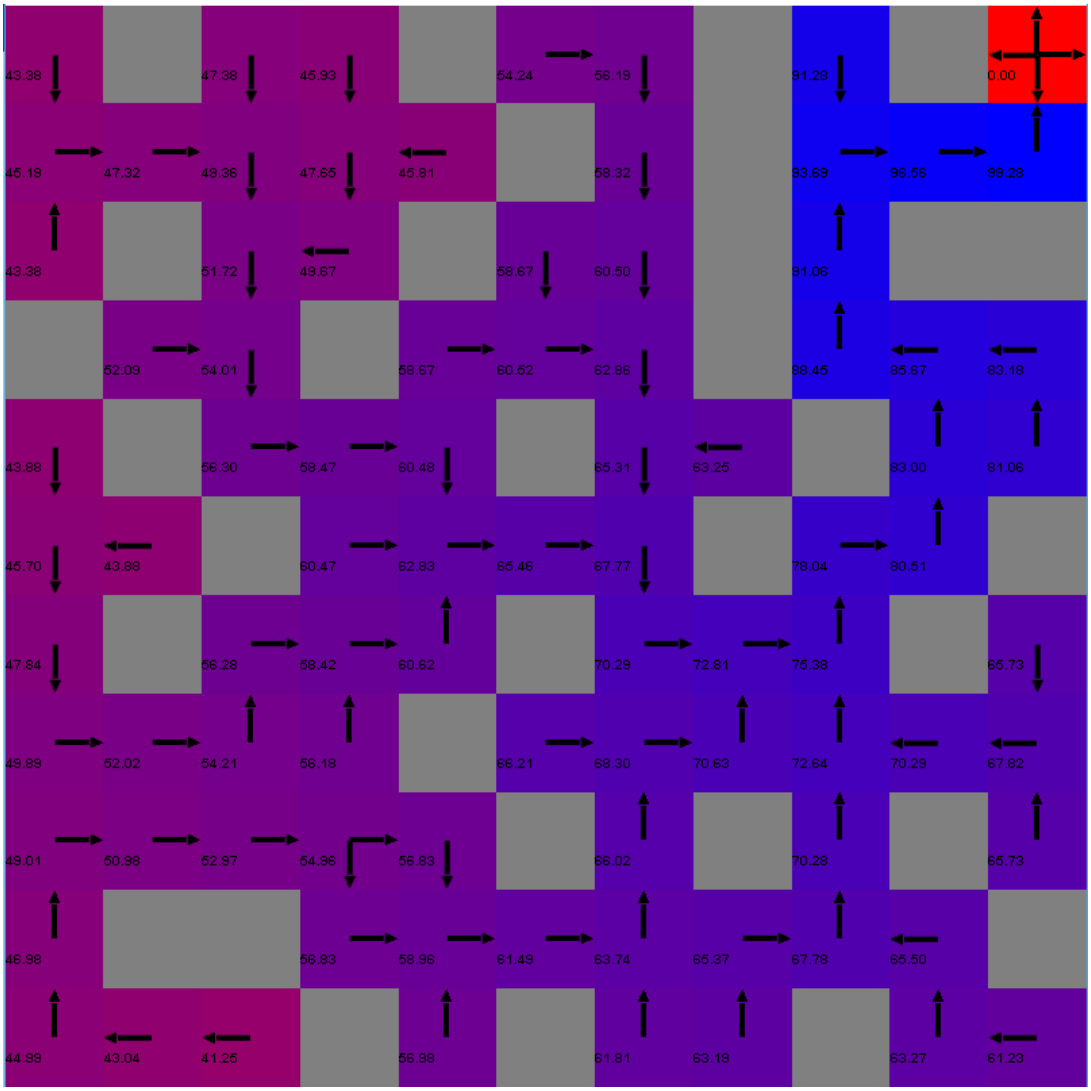


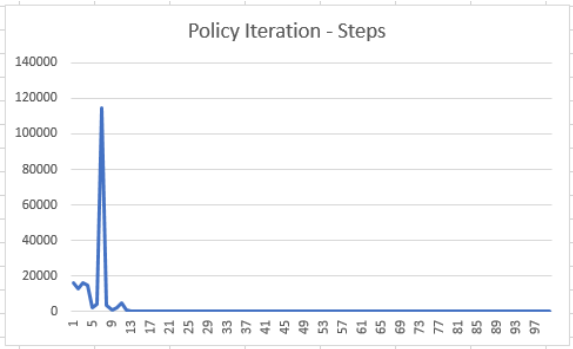


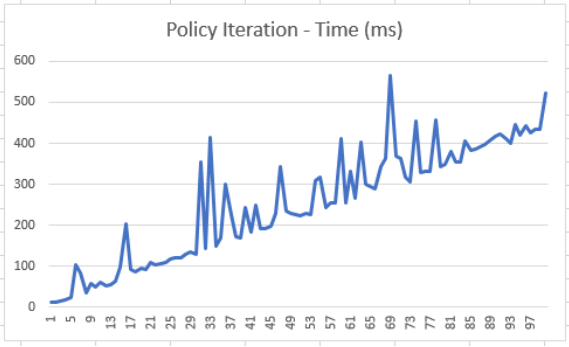


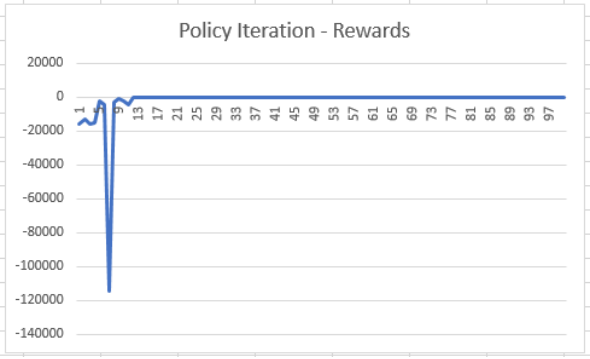
The large MDP takes 25 steps to converge with value iteration. The reward value at convergence is 77. The amount of time necessary to converge varies between 96 and 225 milliseconds.

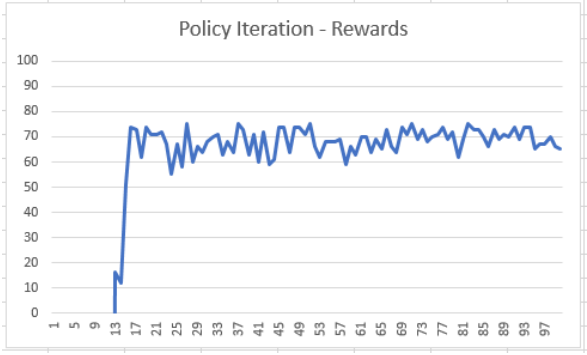
Large MDP – Policy Iteration











The large MDP takes 27 steps to converge with policy iteration. The reward value at convergence is 75. The amount of time necessary to converge varies between 68 and 163 milliseconds.

Analysis

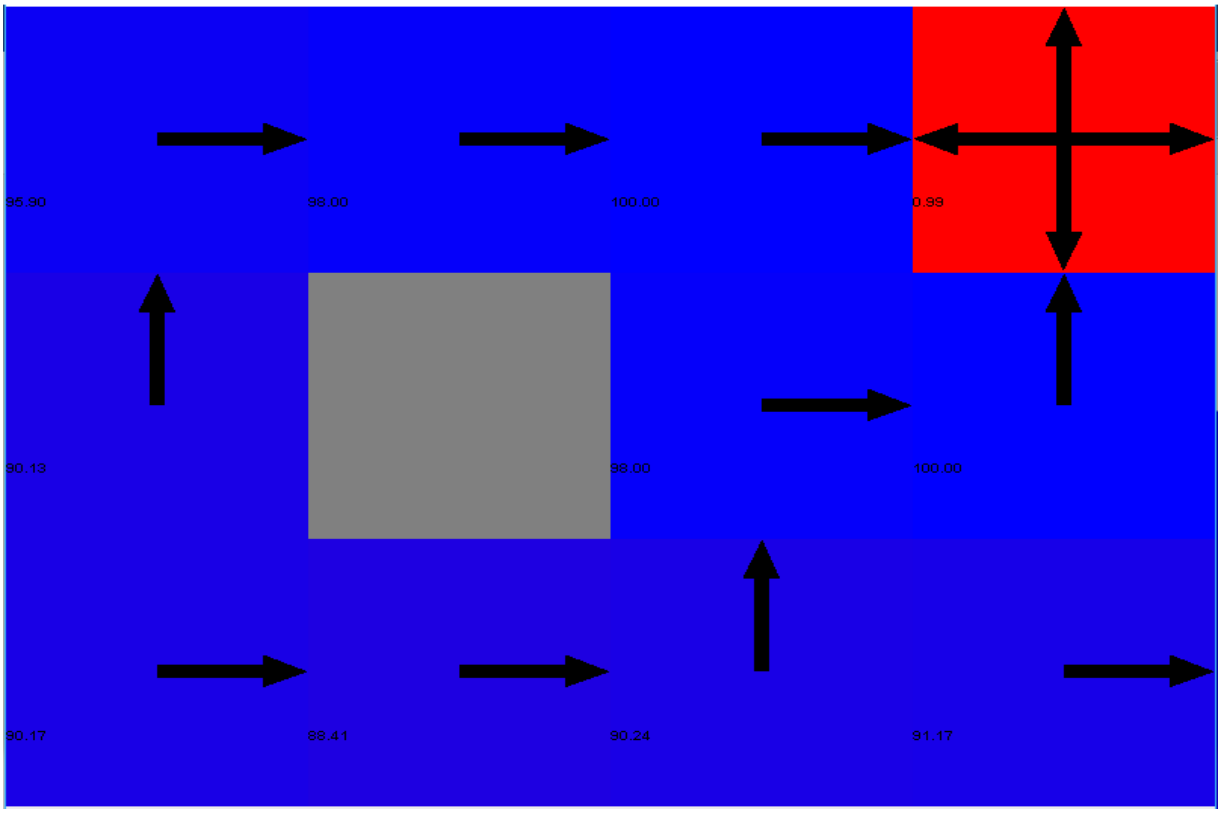
In the small MDP, value and policy iteration each take similar amounts of time and converge to the same result in the same number of steps. The MDP is small enough that both algorithms find the optimum policy/value function relatively easily.

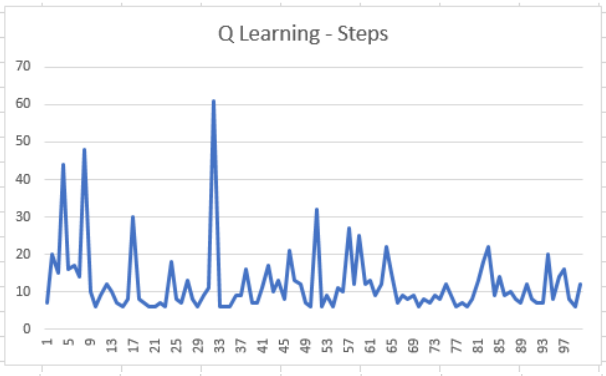
In the large MDP, value iteration finds a policy that allows it to converge in 25 steps with a reward of 77. Policy iteration finds a policy that allows it to converge in 27 steps with a reward of 75. The policy iteration algorithm works more quickly.

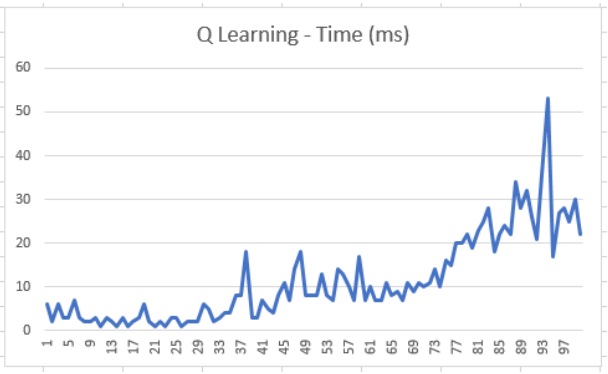
The larger number of states caused the algorithms to converge to different results, take longer to complete, and have more variability each run.

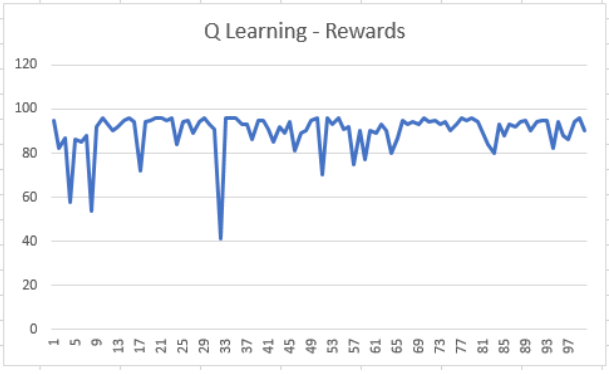
Q Learning

Small MDP



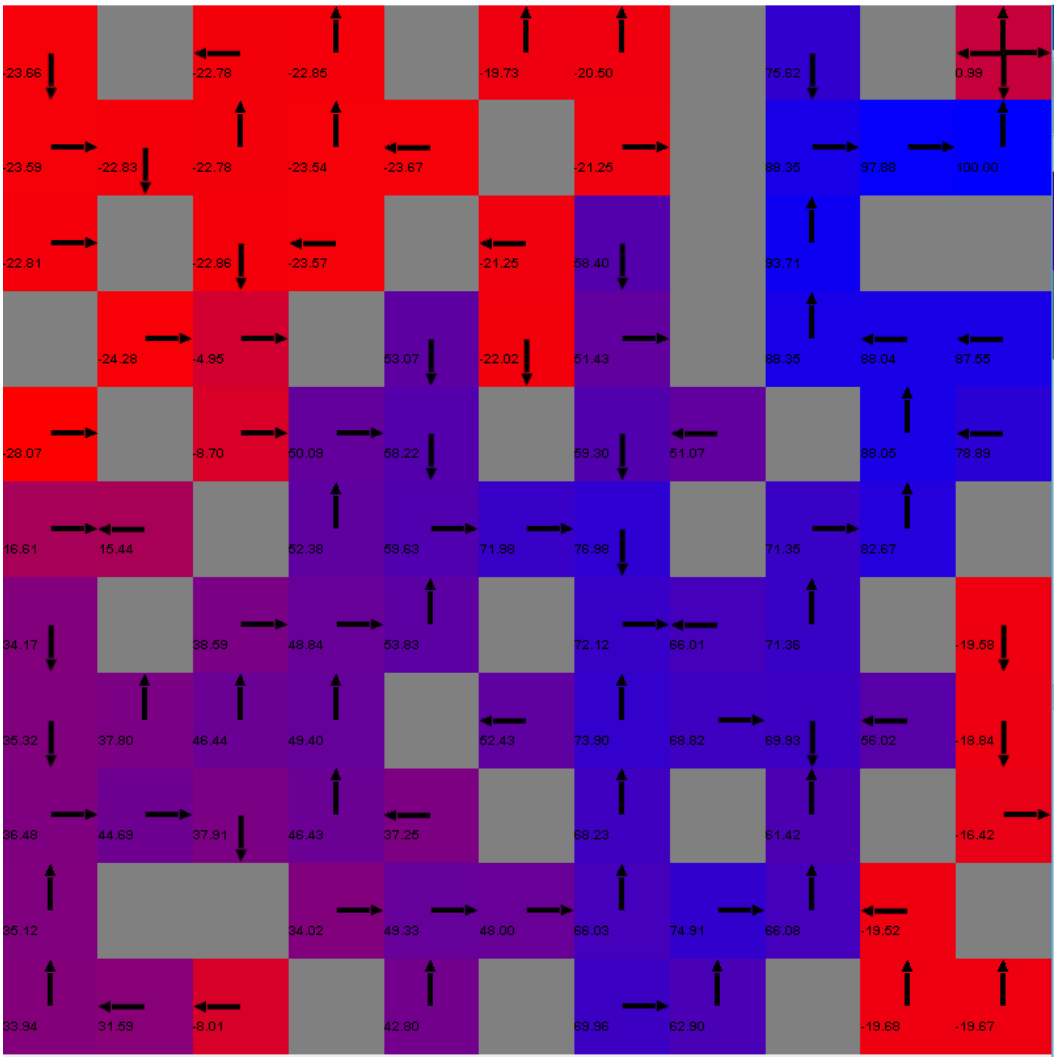


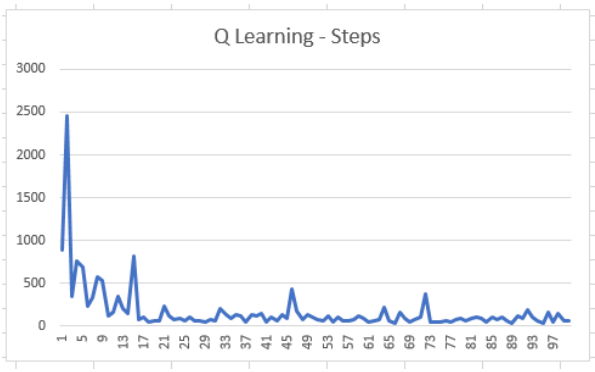


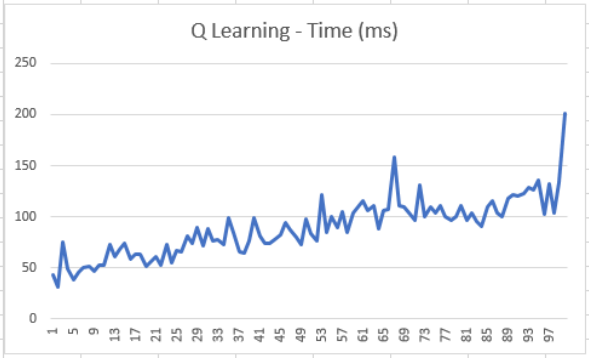


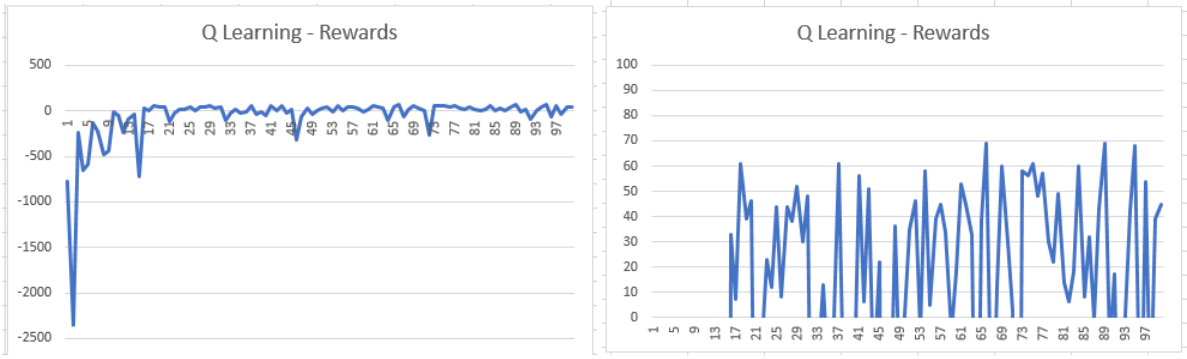
The small MDP takes 6 steps to converge with Q learning. The reward value at convergence is 96. The amount of time necessary to converge varies between 3 and 30 milliseconds.

Large MDP









The large MDP takes 33 steps to converge with Q learning. The reward value at convergence is 69. The amount of time necessary to converge varies between 107 and 118 milliseconds.

Analysis

Q learning performed similarly to the value iteration and policy iteration planning algorithms on the small MDP, converging to the same result in the same number of steps in similar time.

On the large MDP Q learning performed worse than either the value iteration or policy iteration planning algorithms. It took a similar amount of time, but converged to a policy that takes 33 steps to converge with a reward value of 69.

Both MDPs are small enough to map out the optimal policy by hand. We can see based on the policies presented in the images, that all 3 algorithms provided the clear best policy for the small MDP. We can see that the value iteration and policy iteration planning algorithms provide the same path, which appears to be the shortest path through the maze. The Q learning algorithm fails to provide a reasonable path that an agent could follow in the image above.