Shravan Byra
CS 440
MP1.2

1) In WorldWithoutThief:
    a) The robot never moves. It just stays in the bottom left corner.
    b) With epsilon set to .1 the robot is able to find the ideal path through the puzzle. There are some simulations in which the robot does not find an optimal path and gets stuck. This result makes sense since epsilon is a low value, so the robot prioritizes. The episode rewards for this test are all positive, mostly above 100.
    c) With epsilon set to .5 the robot starts to find the ideal path but tends to get stuck at some point of the puzzle. Since the epsilon value is a high number it prioritizes a random direction equally with the highest reward path and isn't able to converge to a single solution. The episodes for the epsilon set to .5 all have negative reward values which shows that it isn't able to find an optimal path.
2) In WorldWithThief:
    a) After testing multiple simulations with epsilon set to .05 the robot does not find an optimal path. For every simulation the robot goes up to the top left corner and then gets stuck. The reward values for each episode were between -5.0 and -20.0. Since the robot doesn't detect the thief, it probably assigns negative values to paths moving to the right (towards the thief).
    b) Now that the robot can detect the thief/knows it is there, it is able to find an ideal path through the puzzle. It successfully delivers the packages while avoiding the thief.
    c) epsilon: .01
    learning rate: .2
    I first started by keeping learning rate at a constant value of .1 and discount at .9. I then went through and ran tests to find reward values when epsilon is 0.1, 0.2, 0.3, 0.4, and 0.5. I noticed that the reward values decreased as I increased epsilon with .1 having the highest value. I then decided to lower the value of epsilon from 0.10 to .01 with .01 increments, and the best value for epsilon is .01, the lowest value tested. Then I went on to test learning rate with values 0.1 - 0.9 with .1 increments and the highest reward value was achieved when learning rate was 0.2.
3) The reward always starts at a negative value and quickly converges to a positive value as more episodes occur. It reaches its expected value around 100 episodes but gets more precise as the simulation continues. The average graph is much smoother than the first trial. The graphs are on the following page.

Shravan Byra
CS 440
MP1.2