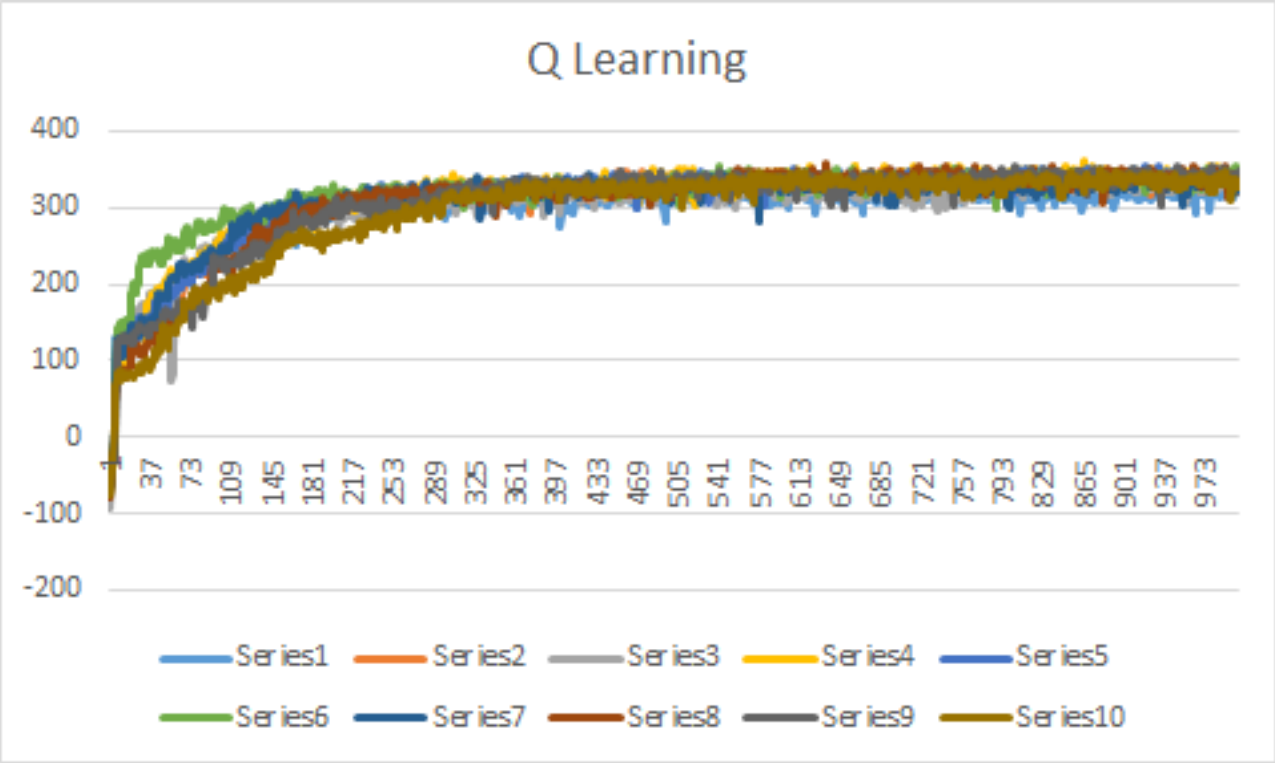
Sunny Duan

MP1
Part 2

1.
   a) All the episode values are 0 because since there is no randomness, the simulator will never explore new regions. Because of this, it will, in most cases take the safe route and do nothing resulting in action 0 since it is not doing any exploration and therefore does not do any learning.
   b) The agent starts out with negative reward values, which is worse than the previous case. This is probably because of the randomness, so the agent will sometimes take a worse action, but because of this the agent is able to do more exploration and is able to find ultimately the more rewarding states. And then because of this, it can update its utility function and slowly as I did more and more tries, the episode values increase up to around 160. So at the end of the learning cycle, this agent has a better policy than the previous.
   c) This time, the episode value doesn't seem to get better. Since epsilon is so high, even though the agent is able to explore a lot of the map faster, it takes the random route so often that the agent does not converge. Rather, it keeps taking random steps. This causes most of the values to be negative, sometimes the agent will get lucky and yield a positive result.

2.
   a) The episode value seem to be pretty erratic at the beginning while it is still learning the world, but then seems to stabilize a bit around -10. Also the bot tends to stay at the company and not move out as this is the best way to not have negative reward. This is most likely due to the fact that we don't know where the thief is, and the thief moves around, we will, from our scope of view, recieve random negative rewards periodically. Due to this fact, it is pretty hard for the program to achieve a good reward value.
   b) In this case it ends up stabilizing around a bit less than 300. This is probably because the agent can see the thief and adjust its movement accordingly. This way it can avoid the negative reward associated with meeting the thief. Furthermore, it will learn to avoid the thief and eventually after many episodes, it has generated a good policy to maximize the reward.
   c) I ended up with the optimal value of epsilon=.0045 and a learning rate of .15. I started with setting the boundaries of 0<epsilon<.4 and 0<learning rate<.5 And I sampled 9 values in a grid-like fashion. Then I found that values towards the middle of the learning rate region and values of small but non-zero epsilon provided the best episode values. So I shrunk the box to around 0<epsilon<.1 and .1<learning rate<.2. And then after applying this similar strategy, I found that changing the learning rate within this range had very little effect on the episode value since most of the variation was due to randomness. But epsilon seemed to perform the best at a value of .0045. So I picked the value .15 for the learning rate and a value of .0045 for the epsilon.

3.




I notice on the single graph that the values toward the beginning are pretty low, but we see after around 180 episodes, we're about pretty close to the best possible episode value. I also notice after running all 10 runs that at the beginning, the 10 different runs vary pretty significantly, but after the agents are about to converge, they are all pretty similar in value. This is probably because they have all reached that upper limit of what they can learn using this algorithm. Towards the end of the simulation, there is still some variation just due to the randomness, but they all appear to have converged at this point.