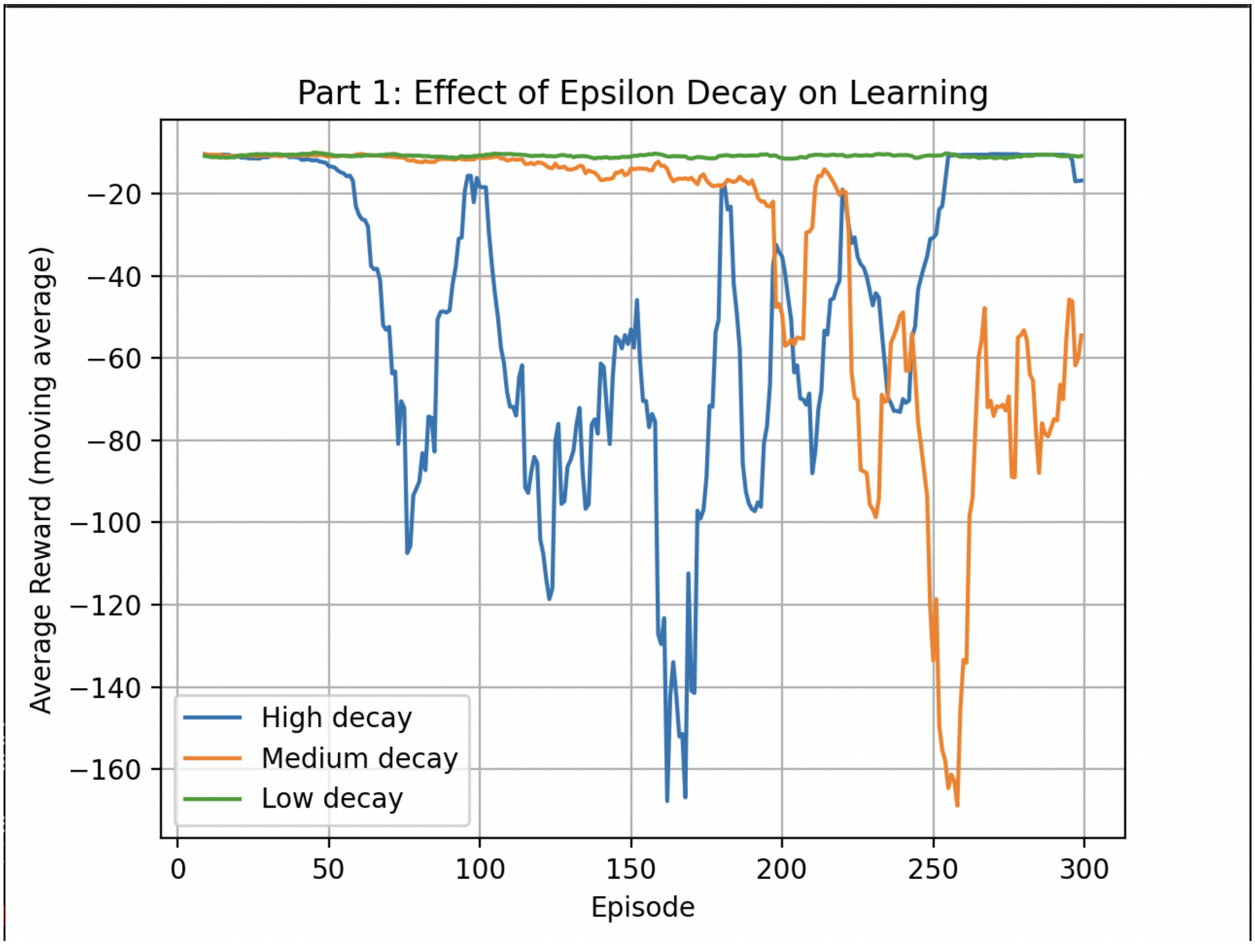
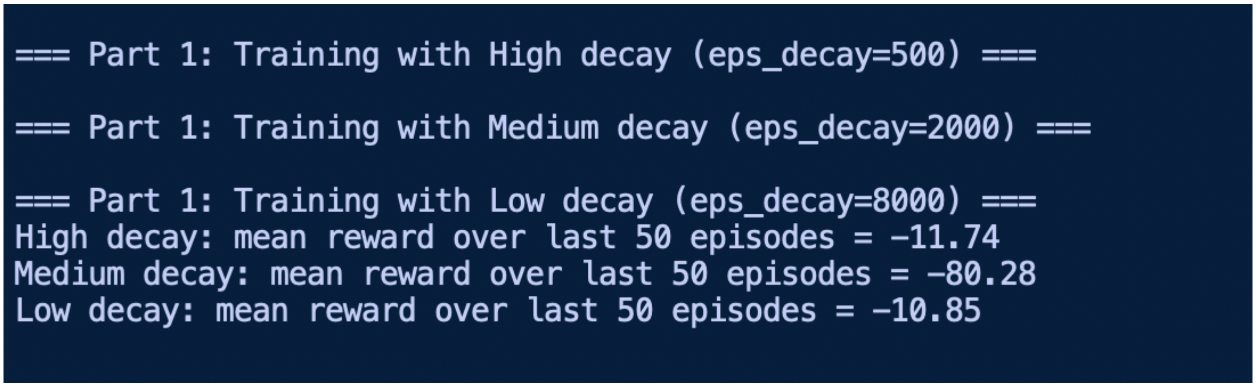
Simon Tice HW5
Part 1:





```
=== Part 1: Training with High decay (eps_decay=500) ===

=== Part 1: Training with Medium decay (eps_decay=2000) ===

=== Part 1: Training with Low decay (eps_decay=8000) ===
High decay: mean reward over last 50 episodes = -11.74
Medium decay: mean reward over last 50 episodes = -80.28
Low decay: mean reward over last 50 episodes = -10.85
```

I trained the DQN three times with different epsilon-decay rates (each 4x bigger than the last):

- High decay: eps_decay = 500

- Medium decay: eps_decay = 2000

- Low decay: eps_decay = 8000

I kept all other settings the same and plotted the moving-average reward per episode. Because each step has a negative cost, the best means the least negative reward.
Final averages over the last 50 episodes:

- High decay: –11.74
  Medium decay: –80.28
- Low decay: –10.85

High Decay:
The blue line (high decay) starts off near the top but then swings up and down a lot, with some very bad stretches (rewards around –100 or worse). This means:

- Epsilon drops quickly because the agent stops exploring early.
- It also latches onto some paths that are okay at first, but because it doesn't keep exploring, it falls into bad habits and struggles to recover.

This fits the expected explanation that high decay pushes the agent into exploitation too soon, which can lead to unstable behavior.

Medium Decay:
The orange line (medium decay) behaves the strangest:

For the first 200 or so episodes, it sits between high and low decay, roughly as expected. Then it collapses into very low rewards (–100 to –160 range) and never really recovers. Its final average (–80.28) is by far the worst.

In theory, the medium decay case should be a compromise between high and low: not too greedy, not too random. In this case, it seems to be a bad compromise. This means:

- Epsilon decays fast enough that the agent starts exploiting. But not slow enough to thoroughly explore, so it can drift into a poor policy later, and it doesn't explore enough to fix it.

So the shape is not exactly what was predicted.

Low Decay:

The green line (low decay) is almost flat near the top of the plot: it stays around –10 for basically the whole training run, and its final average (–10.85) is the best of the three.

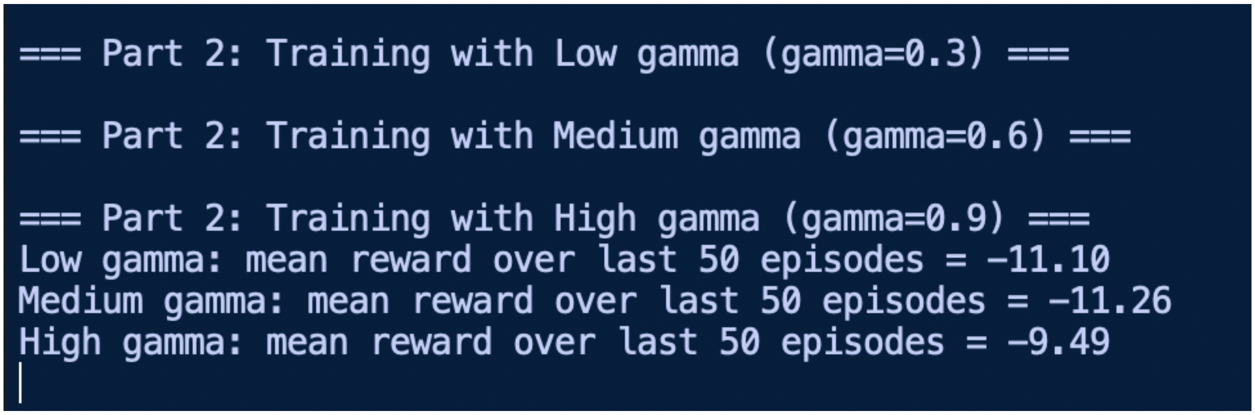
According to the theory, low decay should keep epsilon high longer, leading to more random exploration. It should also take longer to stabilize but eventually find the best path.

In my test run, the “slow to stabilize” part is not very visible. The line is good from early on, but the main expected effect does show up:

- The low-decay version ends up with the best (least negative) long-term reward, meaning it learns the strongest overall policy.

So overall, my experiments in part 1 follow the general rules you would expect for the weird behavior of the medium decay halfway through the test.

Part 2:



```
=== Part 2: Training with Low gamma (gamma=0.3) ===

=== Part 2: Training with Medium gamma (gamma=0.6) ===

=== Part 2: Training with High gamma (gamma=0.9) ===
Low gamma: mean reward over last 50 episodes = -11.10
Medium gamma: mean reward over last 50 episodes = -11.26
High gamma: mean reward over last 50 episodes = -9.49
```

What I measured:
I kept the best epsilon-decay setting from Part 1 (low decay) and only changed gamma:

- Low gamma (0.3): mean reward ≈ –11.10
- Medium gamma (0.6): mean reward ≈ –11.26

- High gamma (0.9): mean reward ≈ –9.49

  Because each step has a negative cost, less negative = better. So the high-gamma run clearly did best.

Expected:

- **Low gamma:** the agent is short-sighted. The future goal reward is heavily discounted, so it doesn't care very much about eventually getting +10. It may wander more, take longer paths, or get stuck in "okay" states that have slightly less penalty.

- **Medium gamma:** Is somewhere in between high and low gamma.

- **High gamma:** the agent is far-sighted. The future goal reward is discounted very little, so it is strongly motivated to push toward the goal and should find the shortest or near-shortest path, giving the best overall return.

Observed in my results:

- The **low-gamma** and **medium-gamma** runs have very similar final rewards (around –11), which suggests both are behaving somewhat short-sighted: they reach the goal but not in the most efficient way.

- The **high-gamma** run has a noticeably better (less negative) final reward, –9.49, meaning the agent is losing fewer points on the way to the goal. This is exactly what we'd expect if it is taking shorter, more direct routes because it values the future goal reward more.

So overall, my experiments for part 2 do match the expected outcomes.