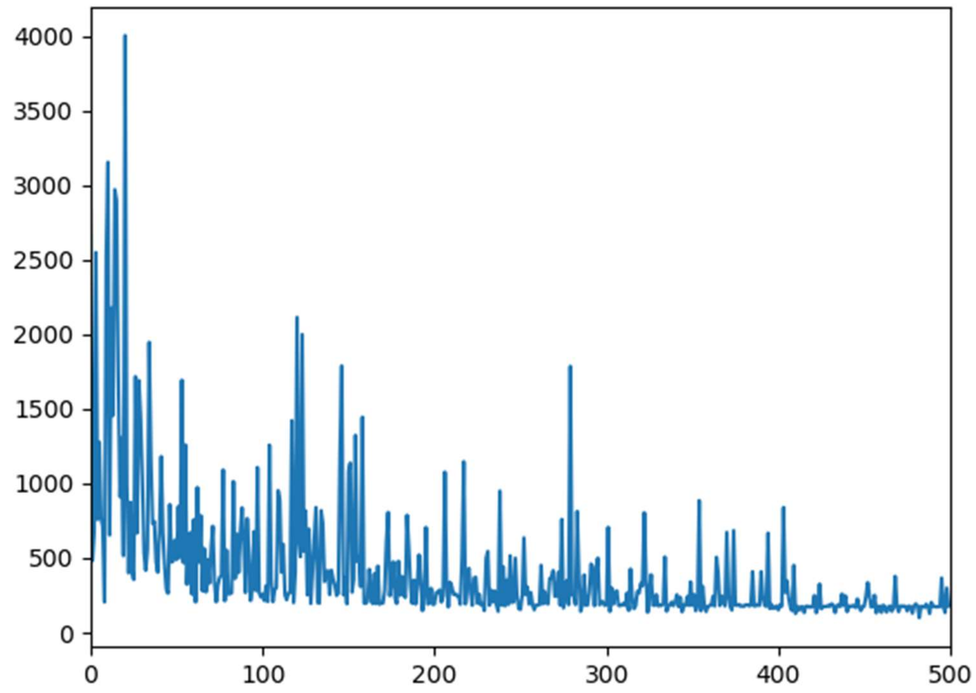
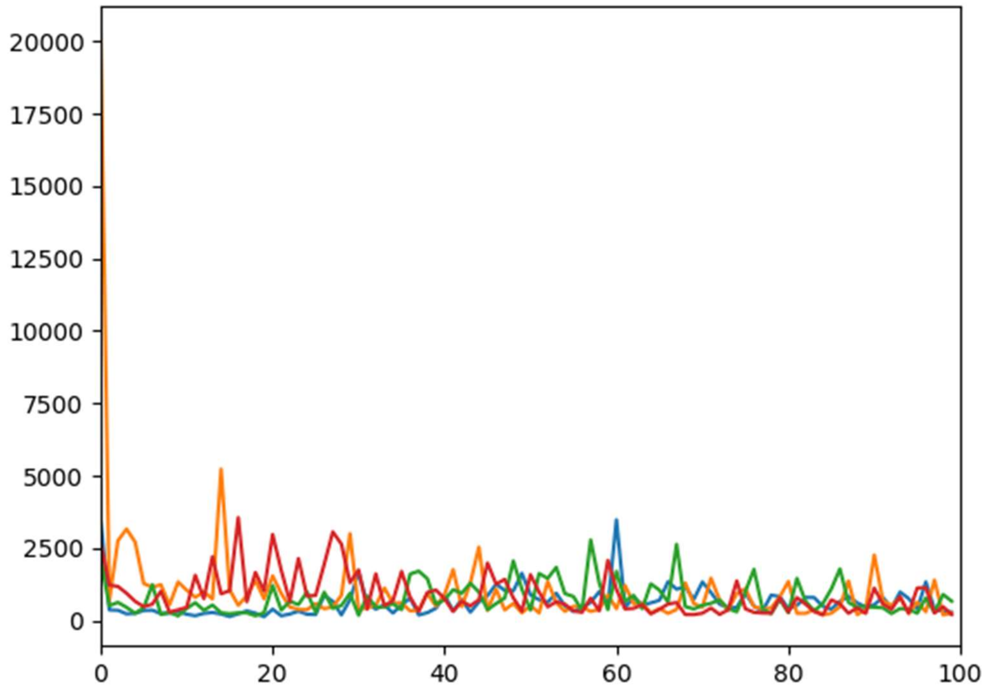
First, We normalize the observation and unwrapped the environment. Most important, we only have two actions : left and right. Because we choose DQN , so we have a memory pool which size is 100000. We set the target model will update while each episode's step reach 10000, 20000, etc. And we select RMSprop optimizer with learning rate = 0.0001



This Figure is the timestep while running 500 episode



This figure show the performance of different batch size

Q1. What kind of RL algorithms did you use? value-based, policy-based, model-based? why?

A1. I use value-based's DQN algorithm. Because DQN can overcomes: (1)Experience replay (2) Target Network (3) Clipping rewards (4) Skipping Frames

Q2. This algorithms is off-policy or on-policy? why?

A2. Off-policy. Because DQN has a replay memory which record many history samples. When we update the target of Q function, we use these history samples. Not have to use the current strategy.

Q3. How does your algorithm solve the correlation problem in the same MDP?

A3. Experience Replay stores experiences including state transitions, rewards and actions, which are necessary data to perform Q learning, and makes mini-batches to update neural networks. This technique expects to reduces correlation between experiences in updating DNN