1. TD methods update your new estimated driving time every time you reach a certain destination interval, and Monte Carlo methods update your estimates only when you reach your new home. Consider a case that you are making a promise to someone requires your immediate answer (using the phone while driving is bad, okay) on will you be on time, with TD methods you can answer accordingly to your new estimate, while Monte Carlo methods have a larger chance of breaking the promise.

   Another example:

   Consider you are working at some company. You want to calculate your estimated wage annually so you can plan ahead on how to spend your money. You expected to earn a fixed amount (e.g. $5k/month) according to your contract.

   Suppose you were assessed by your boss for a bonus: for Monte Carlo methods, you will have to wait until the year ends to update your estimate (and change your spending plan); while TD methods updates your estimate every time you receive a paycheck, thus you can update your spending plan right after you get your money.

   P.S. This example is only applicable to people who enjoys life and does not save money by habit

2. In the first episode, it is likely that the agent terminated left from state A.

   Due to the algorithm of TD updating, under the condition of all state values are equal, the value of next state subtract the current state and adding a reward of 0 results in a 0, thus no updating of states other than state A is made.

   The updating is as follows:
   V(B) = V(C) = V(D) = V(E) = 0.5 + 0.1[0 + 1(0.5) – 0.5] = 0.5
   V(A) = 0.5 + 0.1[0 + 1(0) – 0.5] = 0.45

3. From the graph, we can see that TD using a higher step size learns faster but sacrifices with a higher long-term error (0.15), and lower step size learns slower but with lower long-term error (0.05). Thus we **cannot** conclude which algorithm is better since the selection of different step sizes is a double-edged sword, with both pros and cons.

   Since using any step size will eventually converge to a specific value, and also previously we cannot conclude which algorithm is better by trying different values of step size, therefore there does not exist such a fixed step size will have either algorithm performs better.

4. Dyna-Q+ finds optimal path faster than Dyna-Q at start, so there is a visible difference between. However, Dyna-Q+ keeps exploring even after finding the optimal path, reducing the learning efficiency unless a shorter path is created, thus the difference between Q and Q+ is narrowed.