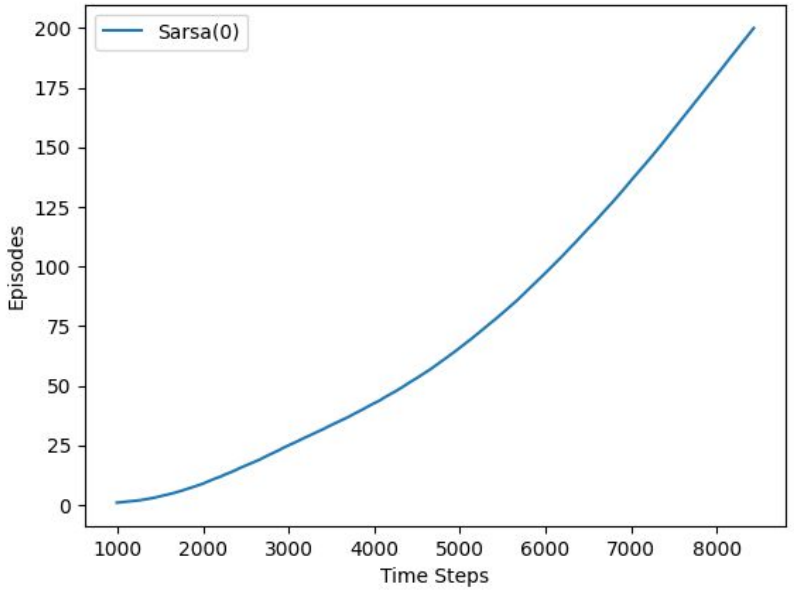
Programming Assignment 3

Hari Krishna Nama (170050077)

```
python solve.py
```

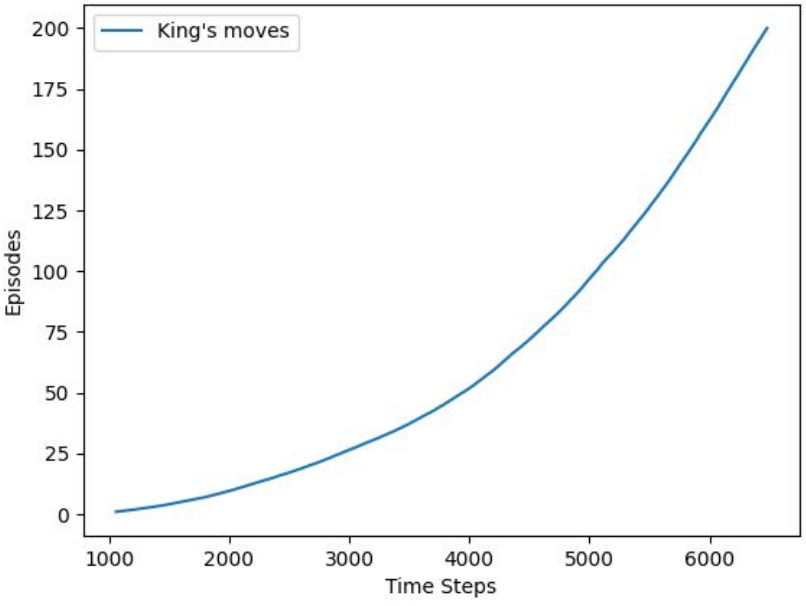
The above command generates all plots for the assignment over 200 episodes and 100 seeds. These generated plots are also attached in the report

Plot for Task-2:



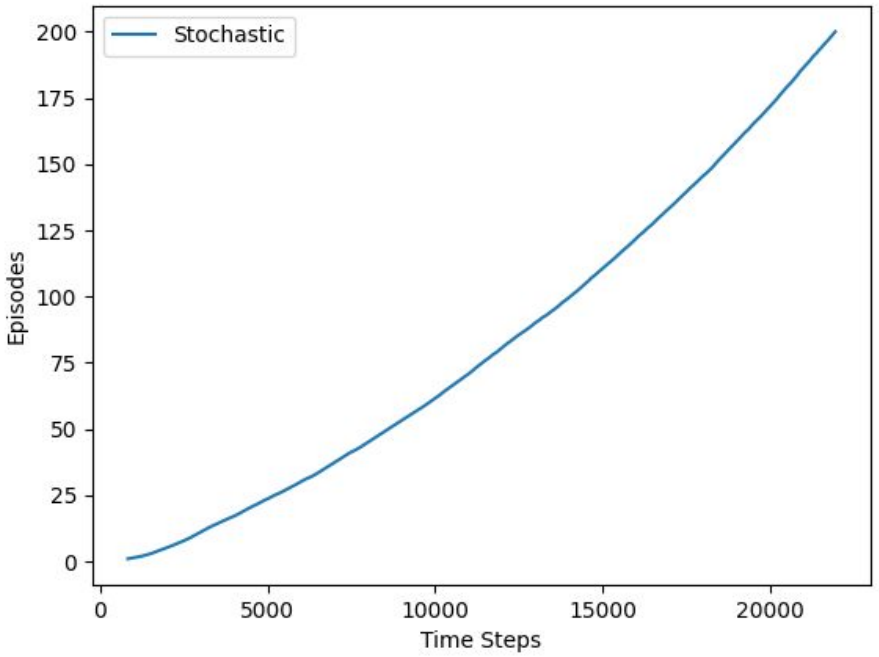
Here we can observe the graph has an increasing slope. So the goal will be reached quickly over time. And the graph obtained is very similar to the graph given in Example 6.5 of Sutton and Barto (2018)

Plot for Task-3:



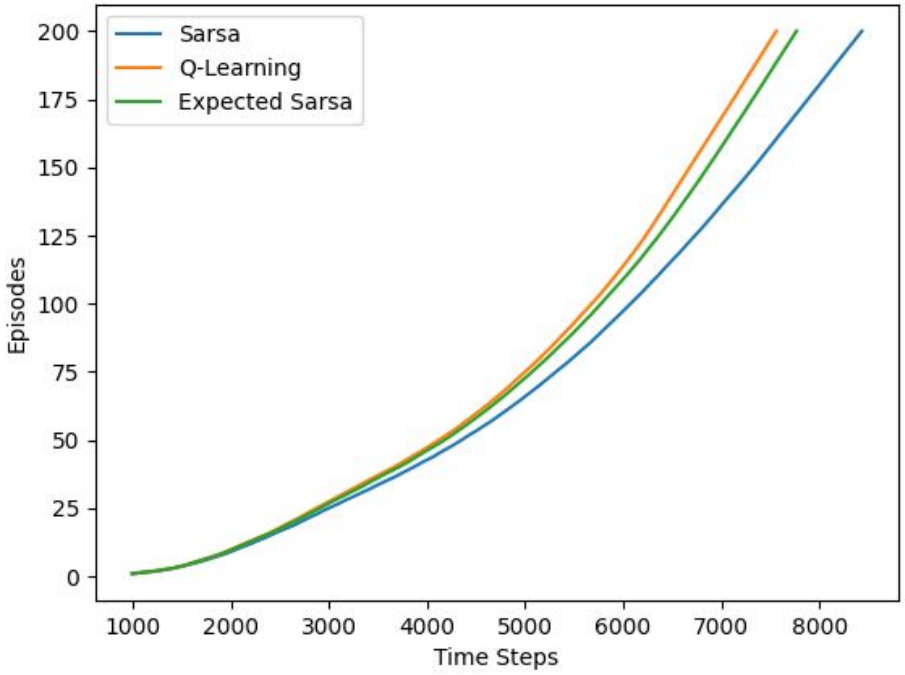
Here for the king's moves, the number of possible moves has increased by 2-fold. So this time the number of time steps took for 200 episodes on average is less compared to task-2 (no king's moves). Simply goal reached quickly(increasing slope) over less time steps

Plot for Task-4:



Here adding stochasticity increased the an uncertainty to the agent to reach the goal. So it took more steps than situation with no stochasticity. Here I assumed stochasticity to all columns even column with 0 wind strength. Another assumption I made around the edges is state change happens as per action and wind strength, if new state is out of bounds then it's state is adjusted(clipped) into the bounds

Plot for Task-5:



Expected sarsa did better than Sarsa because here we took weighted average over all possible actions. Q-learning took fewer time steps because Q-learning considers only max Q over all actions. But our epsilon-greedy policy has epsilon probability of random picking of action. So Q-learning took fewer time steps compared to other agents

Hyperparameters used:

- epsilon = 0.1 used for epsilon-greedy policy
- alpha = 0.5 used as learning rate for Action Value(Q)
- gamma = 1.0
- Number of episodes = 200 used as the default value for running episodes
- Number of seeds = 100 used as the default value for averaging time steps