CISC 474

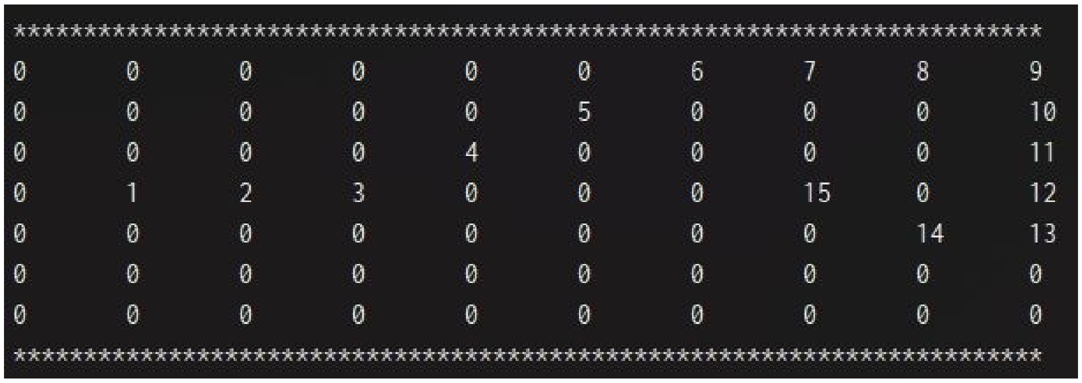
Assignment 3 - Windy Grid World

Duncan Clarke - 20056561

Daniel Goldstein - 20119615

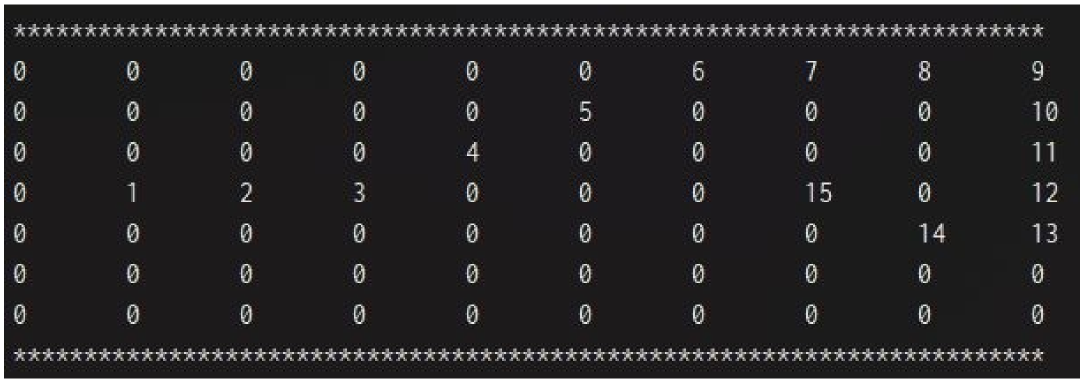
Comparison of Solutions - Normal Winds

SARSA:



```
***************************************************************************
0         0         0         0         0         0         6         7         8         9
0         0         0         0         0         5         0         0         0         10
0         0         0         0         4         0         0         0         0         11
0         1         2         3         0         0         0         15        0         12
0         0         0         0         0         0         0         0         14        13
0         0         0         0         0         0         0         0         0         0
0         0         0         0         0         0         0         0         0         0
***************************************************************************
```

Q-Learning:

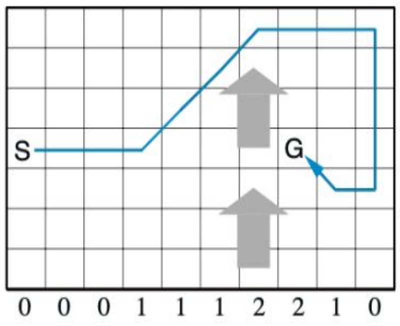


```
***************************************************************************
0         0         0         0         0         0         6         7         8         9
0         0         0         0         0         5         0         0         0         10
0         0         0         0         4         0         0         0         0         11
0         1         2         3         0         0         0         15        0         12
0         0         0         0         0         0         0         0         14        13
0         0         0         0         0         0         0         0         0         0
0         0         0         0         0         0         0         0         0         0
***************************************************************************
```

Comparison

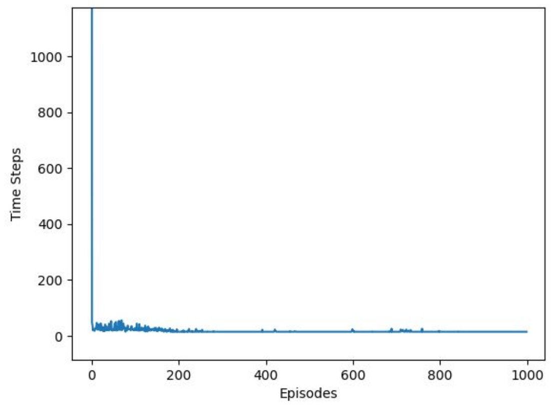
Based on the above solutions it is clear that both QLearning and Sarsa return the same path. They also both return the optimal path.

Image is of optimal solution.

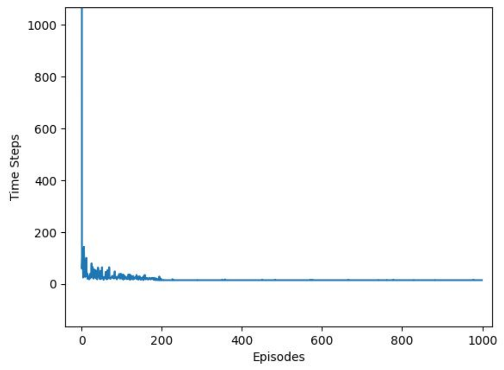


Convergence

SARSA - Converges after ~4 episodes



Q-Learning - Converges after ~4 episodes

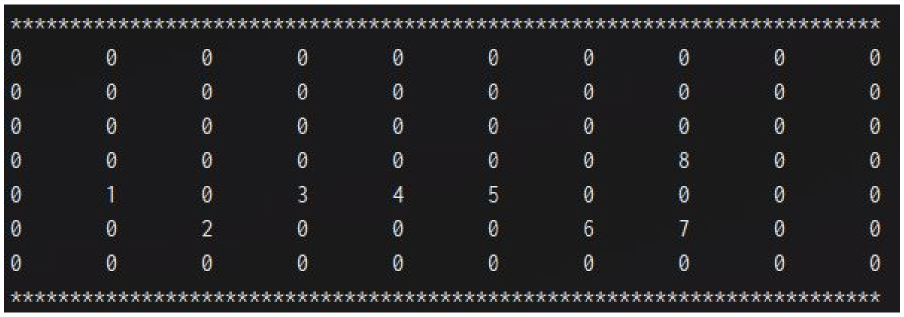


Comparison of Solutions - King's Moves with Stochastic Wind

Sample Solutions

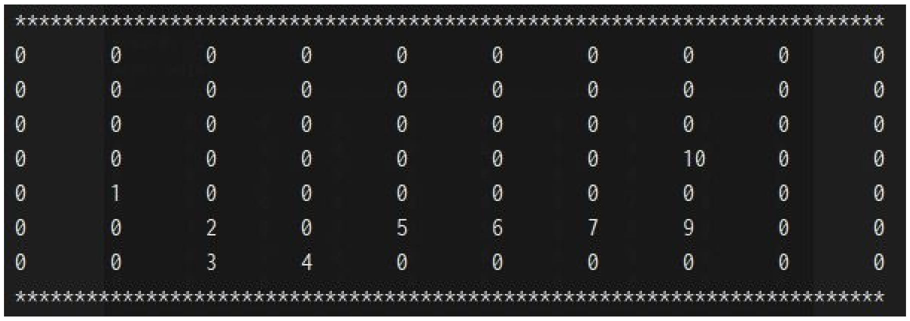
- Solutions vary depending on behavior of stochastic wind

SARSA:



```
**************************************************************************
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       8       0       0
0       1       0       3       4       5       0       0       0       0
0       0       2       0       0       0       6       7       0       0
0       0       0       0       0       0       0       0       0       0
**************************************************************************
```

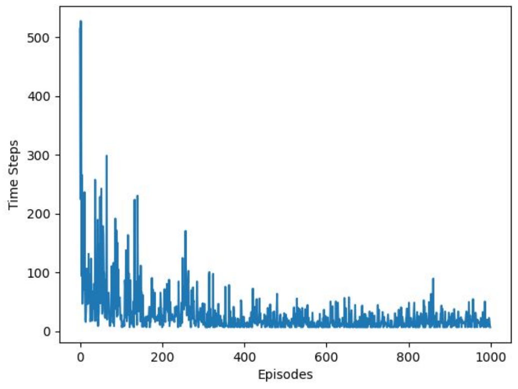
Q-Learning:



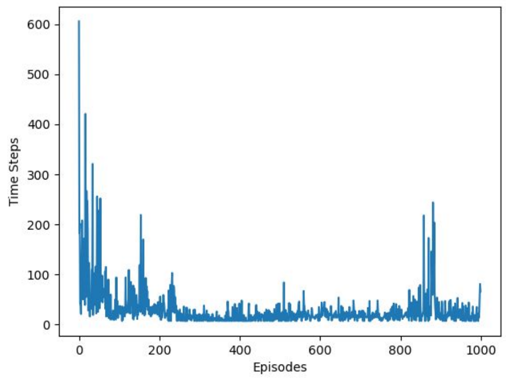
```
**************************************************************************
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       10      0       0
0       1       0       0       0       0       0       0       0       0
0       0       2       0       5       6       7       9       0       0
0       0       3       4       0       0       0       0       0       0
**************************************************************************
```

Convergence

SARSA - converges after ~300 episodes



Q-Learning - converges after ~300 episodes



Comparison of stochastic grid worlds

Both SARSA and Q-Learning algorithms behave similarly for the stochastic windy gridworld problem. Depending on the wind values, the agent must occasionally loop back around in the same way it does with normal wind. However, often the environment is such that the agent is able to move directly towards the goal without overshooting upwards, which significantly decreases the accumulated negative reward. Additionally, the pattern of convergence for both SARSA and Q-Learning also seems to be quite similar, and they each converge to a stable optimal policy for the particular wind array after around 300 episodes.

Alpha, Epsilon values discussion

The alpha and epsilon values were the same for both SARSA and QLearning. There was no need for them to be different as we achieved the optimal solution using the same values for both SARSA and QLearning.

Code Description and Analysis

Env: Class

action: (action) -> state, reward

Description: Updates the players state according to the action parameter. Method also handles kings moves as well as stochastic winds.

possible_actions: (state) -> list_actions

Description: Method returns a list of all possible actions the agent can take based on the current game state. Prevents the agent from choosing invalid moves.

Agent: Class

initialize_Q: (states) -> Q

Description: Initializes the state action value function by traversing the environment provided through the env class, based on the states parameter.

max_Q: (Q, state) -> (action, value)

Description: Returns the optimal action and value based on the given state

epsilon_greedy: (epsilon, position) -> action

Description: Returns an action based on the epsilon greedy approach

sarsa: (iteration, gamme, alpha, epsilon) -> total_reward, nSteps

Description: Applies the sarsa algorithm and returns the total reward and number of steps taken.

Q_learning: (iteration, gamme, alpha, epsilon) -> total_reward, nSteps

Description: Applies the Q_learning algorithm and returns the total reward and number of steps taken

optimal_policy: () -> reward, visits

Description: Applies the optimal policy generated through either Q_learning or SARSA, agent takes the path following that policy. Returns the given reward and visited grid locations.

All other code runs the methods defined in the above classes and outputs the solution to the Windy Grid World problem.