CISC 856 - Reinforcement Learning

Assignment #2:

Windy Grid World

Problem Definition:

Windy Gridworld is a standard gridworld, with start and goal states, but with one difference: there is a crosswind running upward through the middle of the grid. The actions are the standard four: (up, down, right, and left), But in the middle region the resultant next states are shifted upward by a “wind,” the strength of which varies from column to column.

Actions:

The actions are defined as a list of standard four directions:

- [Up, Down, Left, Right]

And in the windy gridworld task with King’s moves, the actions will be a list of eight directions:

- [Up, Down, Left, Right, Up_Left, Up_Right, Down_Left, Down_Right]

Rewards:

The reward for reaching the goal state is defined with 0 and for any other state is defined with -1

Implementation:

The first step I made is creating the environment for the Windy Griswold problem that contains the initialization of the environment and actions (whether the standard four actions or the king’s moves) then creating the main function that: checks the terminal state, get the position of the cell, return the reward, and return the next state (wither the problem is stochastic or not)

Then I create functions that represent the agent such as: the function that choose the greedy action to be used in SARSA and Q-Learning Algorithms, and the functions that apply the SARSA and Q-Learning Algorithms for both cases (deterministic and stochastic)

Finally Experiment with both algorithms for (the deterministic case with the standard four actions) and for (the stochastic case with king’s moves) using three different combinations of for ∈ and α with gamma = 0.9, and for 1000 episodes

Observations:

By experimenting with both SARSA and Q-learning algorithms (in the deterministic and stochastic cases) for 1000 episodes with 3 different combinations for ∈ and α with gamma = 0.9.

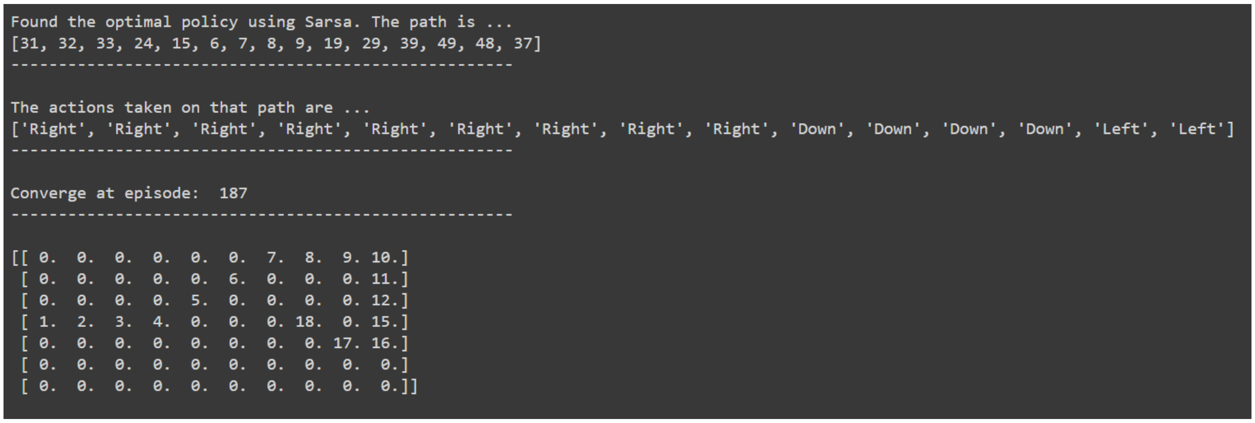
- Epsilon = 0.1, Alpha = 0.5
- Epsilon = 0.2, Alpha = 0.1
- Epsilon = 0.05, Alpha = 0.2

Comparisons (SARSA & Q-Learning)

Algorithm	**Epsilon**	**Alpha**	**Gamma**	**Number of Episode taken to reach convergence**	**Minimum time steps are taken to reach the goal**	**Total number of steps taken to reach 1000 episodes (approx.)**
SARSA	0.1	0.5	0.9	187	15	23500
Q-Learning	0.1	0.5	0.9	104	15	21000
SARSA	0.2	0.1	0.9	67	15	42000
Q-Learning	0.2	0.1	0.9	60	15	40000
SARSA	0.05	0.2	0.9	227	15	26000
Q-Learning	0.05	0.2	0.9	262	15	26000

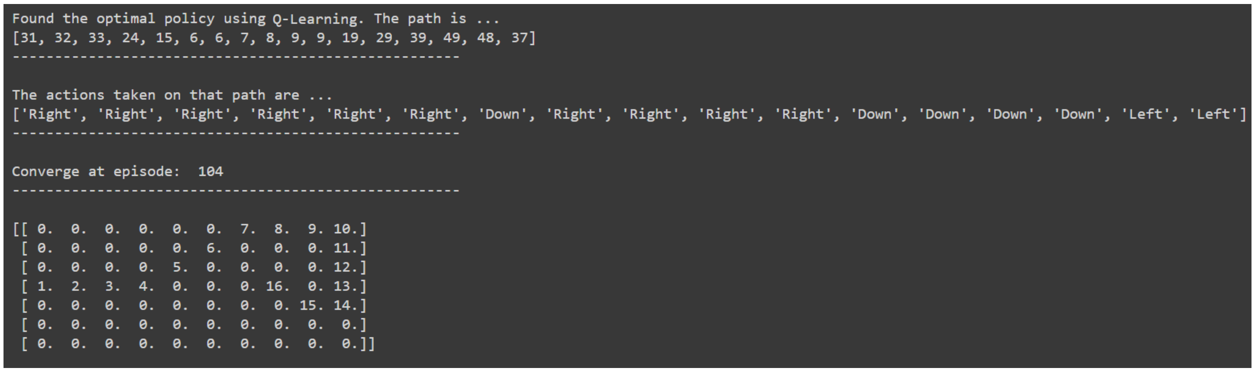
From the previous table, we can notice that:

- The minimum steps that are taken to reach the goal for all experiments are the same value which is 15 steps
- For (∈=0.1, α=0.5) and (∈=0.2, α=0.1): SARSA take more episodes than Q-Learning to converge
- For (∈=0.05, α=0.2): Q-Learning takes more episodes than SARSA to converge
- SARSA using (∈=0.1, α=0.5) converges after episode 187



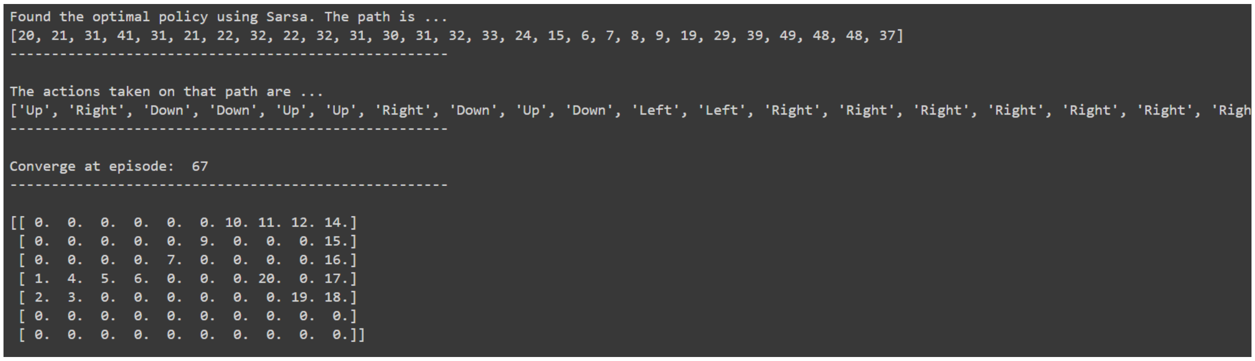
```
Found the optimal policy using Sarsa. The path is ...
[31, 32, 33, 24, 15, 6, 7, 8, 9, 19, 29, 39, 49, 48, 37]
----------------------------------------------------

The actions taken on that path are ...
['Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Down', 'Down', 'Down', 'Down', 'Left', 'Left']
----------------------------------------------------

Converge at episode:  187
----------------------------------------------------

[[ 0.  0.  0.  0.  0.  0.  7.  8.  9. 10.]
 [ 0.  0.  0.  0.  0.  6.  0.  0.  0. 11.]
 [ 0.  0.  0.  0.  5.  0.  0.  0.  0. 12.]
 [ 1.  2.  3.  4.  0.  0.  0. 18.  0. 15.]
 [ 0.  0.  0.  0.  0.  0.  0.  0. 17. 16.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
```

- Q-Learning using (∈=01, α=0.5) converges after episode 104



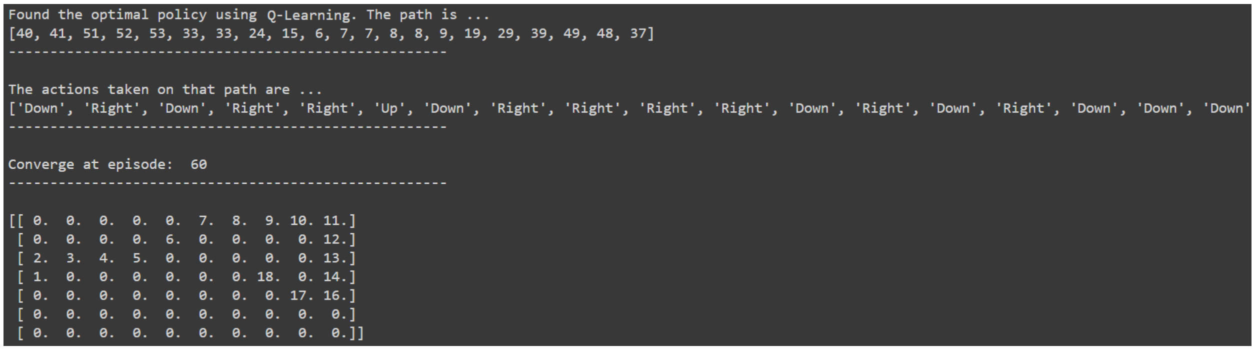
```
Found the optimal policy using Q-Learning. The path is ...
[31, 32, 33, 24, 15, 6, 6, 7, 8, 9, 9, 19, 29, 39, 49, 48, 37]
-----------------------------------------------------

The actions taken on that path are ...
['Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Down', 'Right', 'Right', 'Right', 'Right', 'Down', 'Down', 'Down', 'Down', 'Left', 'Left']
-----------------------------------------------------

Converge at episode:  104
-----------------------------------------------------

[[ 0.  0.  0.  0.  0.  0.  7.  8.  9. 10.]
 [ 0.  0.  0.  0.  0.  6.  0.  0.  0. 11.]
 [ 0.  0.  0.  0.  5.  0.  0.  0.  0. 12.]
 [ 1.  2.  3.  4.  0.  0.  0. 16.  0. 13.]
 [ 0.  0.  0.  0.  0.  0.  0.  0. 15. 14.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
```

- SARSA using (∈=0.2, α=0.1) converges after episode 67



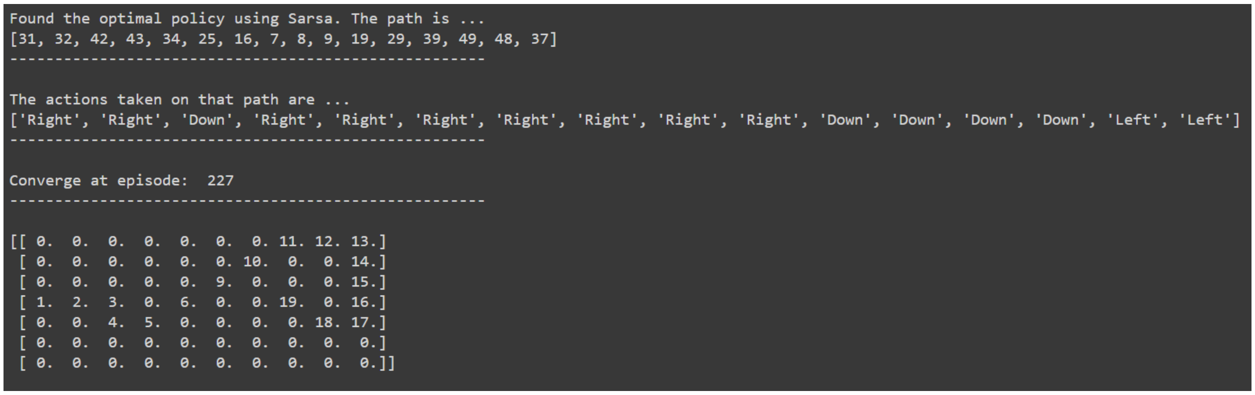
```
Found the optimal policy using Sarsa. The path is ...
[20, 21, 31, 41, 31, 21, 22, 32, 22, 32, 31, 30, 31, 32, 33, 24, 15, 6, 7, 8, 9, 19, 29, 39, 49, 48, 48, 37]
-----------------------------------------------------

The actions taken on that path are ...
['Up', 'Right', 'Down', 'Down', 'Up', 'Up', 'Right', 'Down', 'Up', 'Down', 'Left', 'Left', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Righ
-----------------------------------------------------

Converge at episode:  67
-----------------------------------------------------

[[ 0.  0.  0.  0.  0.  0. 10. 11. 12. 14.]
 [ 0.  0.  0.  0.  0.  9.  0.  0.  0. 15.]
 [ 0.  0.  0.  0.  7.  0.  0.  0.  0. 16.]
 [ 1.  4.  5.  6.  0.  0.  0. 20.  0. 17.]
 [ 2.  3.  0.  0.  0.  0.  0.  0. 19. 18.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
```

- Q-Learning using (∈=0.2, α=0.1) converges after episode 60



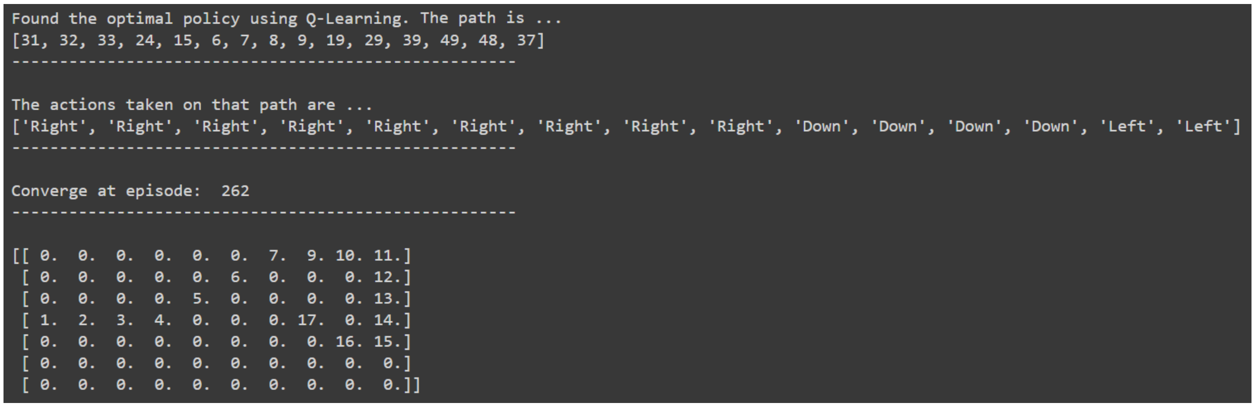
```
Found the optimal policy using Q-Learning. The path is ...
[40, 41, 51, 52, 53, 33, 33, 24, 15, 6, 7, 7, 8, 8, 9, 19, 29, 39, 49, 48, 37]
-----------------------------------------------------

The actions taken on that path are ...
['Down', 'Right', 'Down', 'Right', 'Right', 'Up', 'Down', 'Right', 'Right', 'Right', 'Right', 'Down', 'Right', 'Down', 'Right', 'Down', 'Down', 'Down'
-----------------------------------------------------

Converge at episode:  60
-----------------------------------------------------

[[ 0.  0.  0.  0.  0.  7.  8.  9. 10. 11.]
 [ 0.  0.  0.  0.  6.  0.  0.  0.  0. 12.]
 [ 2.  3.  4.  5.  0.  0.  0.  0.  0. 13.]
 [ 1.  0.  0.  0.  0.  0.  0. 18.  0. 14.]
 [ 0.  0.  0.  0.  0.  0.  0.  0. 17. 16.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
```

- SARSA using (∈=0.05, α=0.2) converges after episode 227



```
Found the optimal policy using Sarsa. The path is ...
[31, 32, 42, 43, 34, 25, 16, 7, 8, 9, 19, 29, 39, 49, 48, 37]
-----------------------------------------------------

The actions taken on that path are ...
['Right', 'Right', 'Down', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Down', 'Down', 'Down', 'Down', 'Left', 'Left']
-----------------------------------------------------

Converge at episode:  227
-----------------------------------------------------

[[ 0.  0.  0.  0.  0.  0.  0. 11. 12. 13.]
 [ 0.  0.  0.  0.  0.  0. 10.  0.  0. 14.]
 [ 0.  0.  0.  0.  0.  9.  0.  0.  0. 15.]
 [ 1.  2.  3.  0.  6.  0.  0. 19.  0. 16.]
 [ 0.  0.  4.  5.  0.  0.  0.  0. 18. 17.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
```

- Q-Learning using (∈=0.05, α=0.2) converges after episode 262



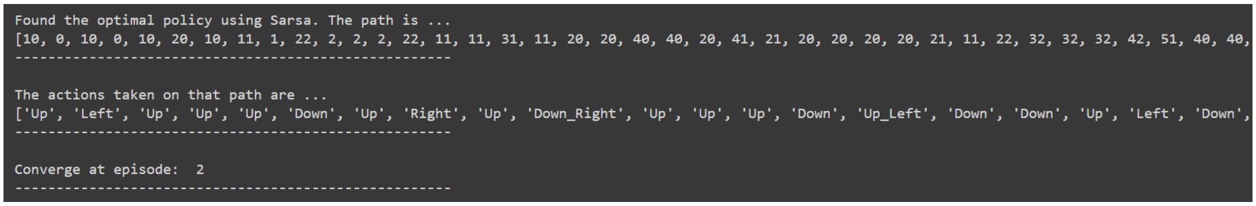
```
Found the optimal policy using Q-Learning. The path is ...
[31, 32, 33, 24, 15, 6, 7, 8, 9, 19, 29, 39, 49, 48, 37]
----------------------------------------------------

The actions taken on that path are ...
['Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Down', 'Down', 'Down', 'Down', 'Left', 'Left']
----------------------------------------------------

Converge at episode:  262
----------------------------------------------------

[[ 0.  0.  0.  0.  0.  0.  7.  9. 10. 11.]
 [ 0.  0.  0.  0.  0.  6.  0.  0.  0. 12.]
 [ 0.  0.  0.  0.  5.  0.  0.  0.  0. 13.]
 [ 1.  2.  3.  4.  0.  0.  0. 17.  0. 14.]
 [ 0.  0.  0.  0.  0.  0.  0.  0. 16. 15.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
```

Comparisons (SARSA & Q-Learning with ‘Stochastic wind’)

Algorithm	**Epsilon**	**Alpha**	**Gamma**	**Number of Episode taken to reach convergence**	**Minimum time steps are taken to reach the goal**	**Total number of steps taken to reach 1000 episodes (approx.)**
SARSA	0.1	0.5	0.9	----	7	41000
Q-Learning	0.1	0.5	0.9	----	7	40000
SARSA	0.2	0.1	0.9	2	7	60000
Q-Learning	0.2	0.1	0.9	1	7	62000
SARSA	0.05	0.2	0.9	----	7	46000
Q-Learning	0.05	0.2	0.9	----	7	44000

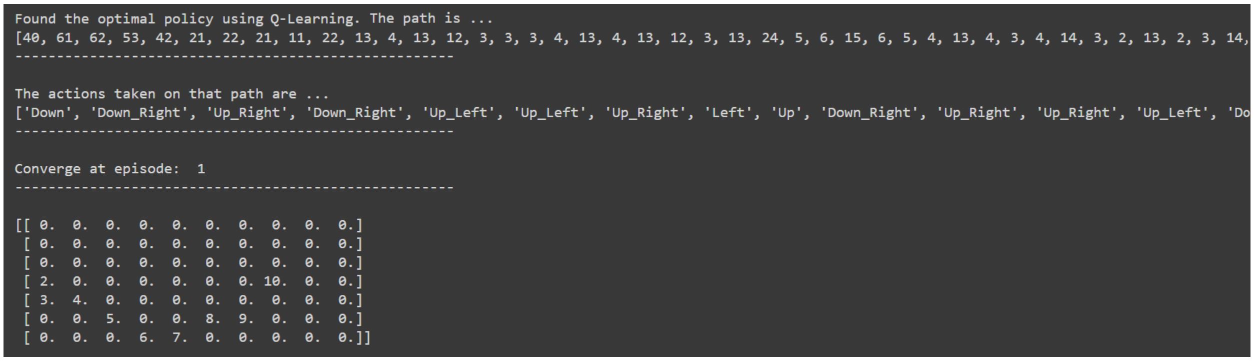
From the previous table, we can notice that:

- SARSA & Q-Learning with ‘Stochastic wind’ using (∈=0.1, α=0.5) and (∈=0.05, α=0.2) don’t converge by trying them with 1000 episodes
- The minimum steps that are taken to reach the goal for all experiments are the same value which is 7 steps
- Q-Learning with ‘Stochastic wind’ converges faster than SARSA with ‘Stochastic wind’ using (∈=0.2, α=0.1)
- SARSA with ‘Stochastic wind’ using (∈=0.2, α=0.1) converges after the second episode



```
Found the optimal policy using Sarsa. The path is ...
[10, 0, 10, 0, 10, 20, 10, 11, 1, 22, 2, 2, 2, 22, 11, 11, 31, 11, 20, 20, 40, 40, 20, 41, 21, 20, 20, 20, 20, 21, 11, 22, 32, 32, 32, 42, 51, 40, 40,
----------------------------------------------------

The actions taken on that path are ...
['Up', 'Left', 'Up', 'Up', 'Up', 'Down', 'Up', 'Right', 'Up', 'Down_Right', 'Up', 'Up', 'Up', 'Down', 'Up_Left', 'Down', 'Down', 'Up', 'Left', 'Down',
----------------------------------------------------

Converge at episode:  2
----------------------------------------------------
```

- Q-Learning with ‘Stochastic wind’ using (∈=0.2, α=0.1) converges after the first episode



```
Found the optimal policy using Q-Learning. The path is ...
[40, 61, 62, 53, 42, 21, 22, 21, 11, 22, 13, 4, 13, 12, 3, 3, 3, 4, 13, 4, 13, 12, 3, 13, 24, 5, 6, 15, 6, 5, 4, 13, 4, 3, 4, 14, 3, 2, 13, 2, 3, 14,
----------------------------------------------------

The actions taken on that path are ...
['Down', 'Down_Right', 'Up_Right', 'Down_Right', 'Up_Left', 'Up_Left', 'Up_Right', 'Left', 'Up', 'Down_Right', 'Up_Right', 'Up_Right', 'Up_Left', 'Do
----------------------------------------------------

Converge at episode:  1
----------------------------------------------------

[[ 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.]
 [ 2.  0.  0.  0.  0.  0.  0. 10.  0.  0.]
 [ 3.  4.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  5.  0.  0.  8.  9.  0.  0.  0.]
 [ 0.  0.  0.  6.  7.  0.  0.  0.  0.  0.]]
```