Jazel A. Suguitan
HW 3 - CS 45

## About the code

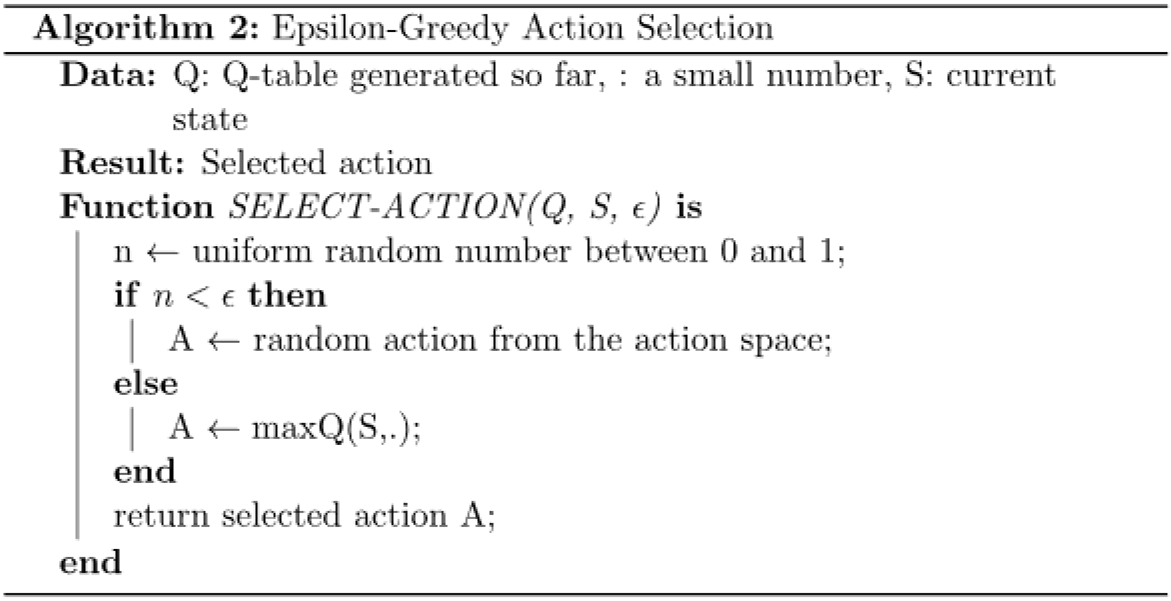
hw3.m - Made with Matlab R2019a

## About the algorithm

**Parameters** used:

- alpha = 0.95
- gamma = 0.05
- epsilon = 0.05

The **action selection policy** used was epsilon-greedy, as described in this pseudocode:



```
Algorithm 2: Epsilon-Greedy Action Selection
  Data: Q: Q-table generated so far, : a small number, S: current
        state
  Result: Selected action
  Function SELECT-ACTION(Q, S, $\epsilon$) is
      n ← uniform random number between 0 and 1;
      if $n < \epsilon$ then
          A ← random action from the action space;
      else
          A ← maxQ(S,.);
      end
      return selected action A;
  end
```

The **actions** a robot can take in any state is to go up, down, right, or left, encoded by the numbers 1-4.

The **states** a robot can be in corresponds with the cells of a grid, state 1 being the top, leftmost cell. Cells/states are numbered from left to right, top to bottom. For example, the numbering schema for a 5x5 grid is pictured below.

|  |  |  |  |  |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |

The **reward system** is slightly different than the system suggested in the assignment guidelines. Reward for actions were given as such:

- Action that makes the robot tend to go out of the grid will get a reward of -1 (when the robot is in the border cells)
- Action that makes the robot reach the goal will get a reward of 100
- Action that brings the robot to a cell previously explored in the current episode will get a reward of 0
- All other actions will get a reward of 0

The **start** cell of each grid is colored green, while the **goal** cell of each grid is colored red.

The **Q table** was organized so that columns are actions (1-4) and rows are states (1-25, assuming a 5x5 grid).

The **total rewards plot** has black vertical lines indicating the ends of learning episodes.

To clearly see the directions of the **arrows** in the grid, it is recommended to view the plots in full screen. Note that arrows may overlap.

## Run #1

- 5x5 grid
- start = 1 (top left cell)
- goal = 25 (bottom right cell)

- number of episodes = 6
- number of iterations in each episode = 180

Figure 2, the first learning episode:

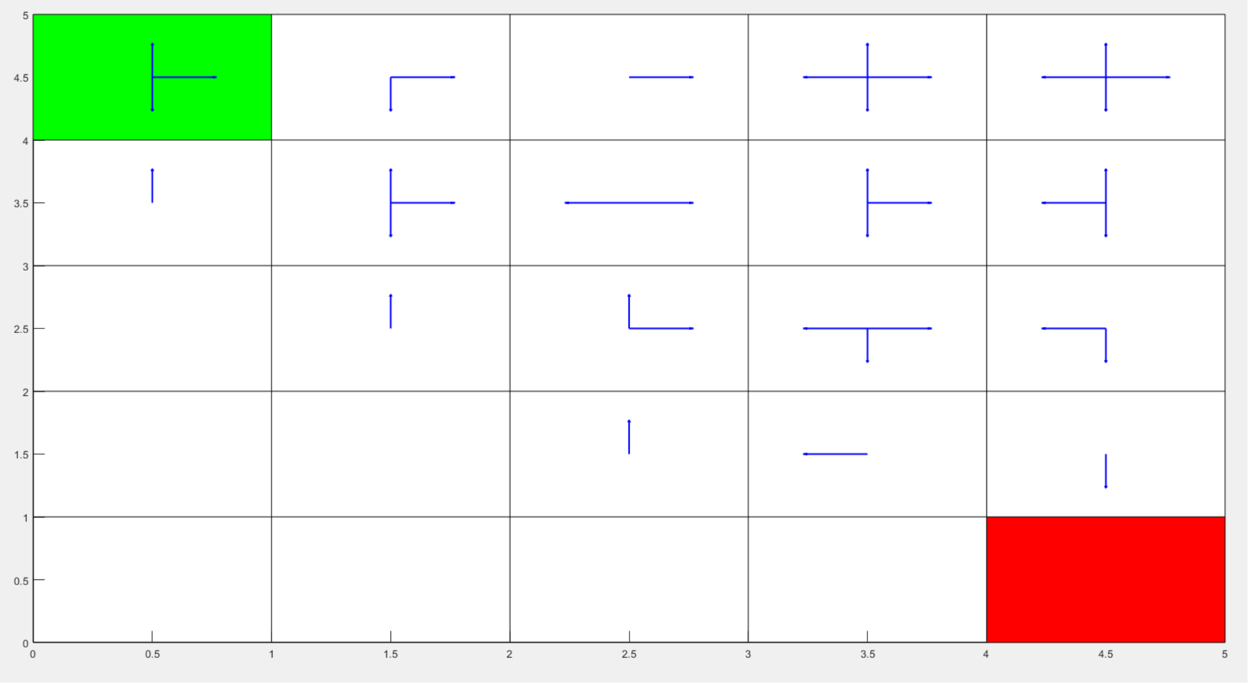
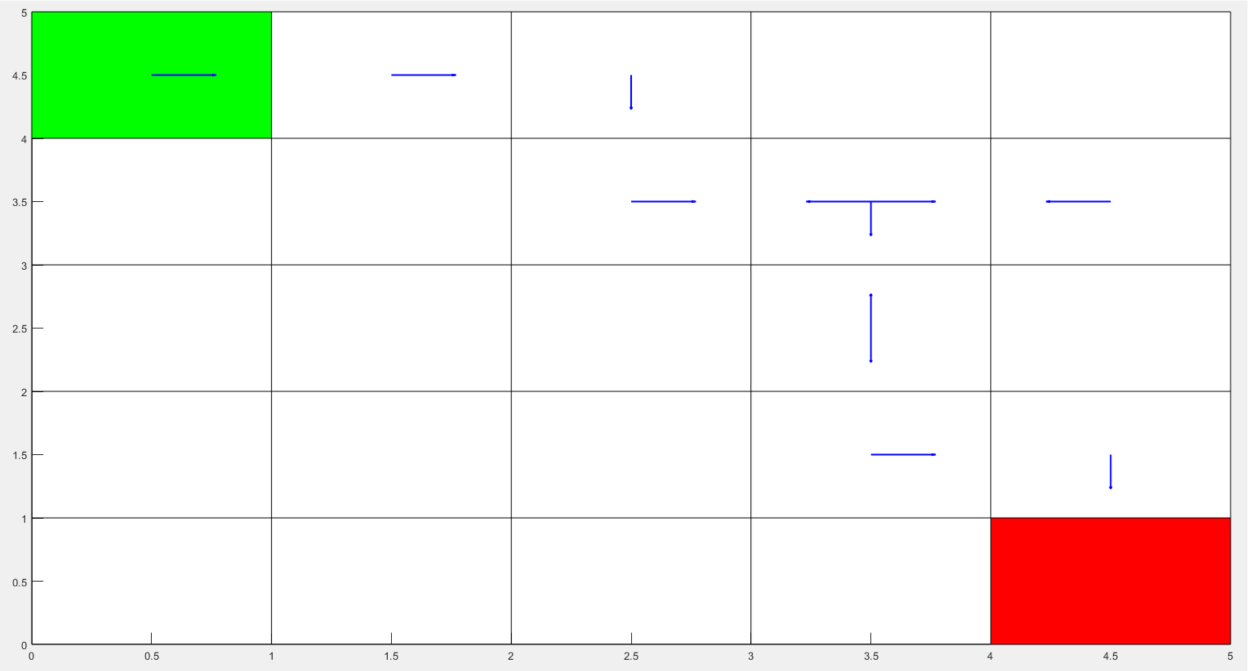
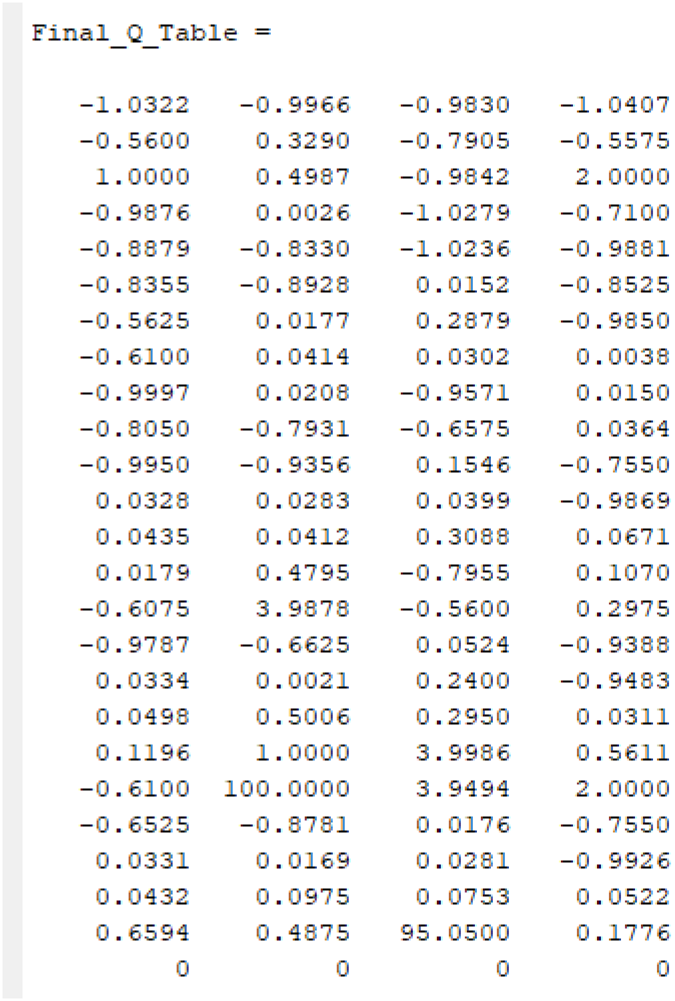


Figure 3, the final learning episode:



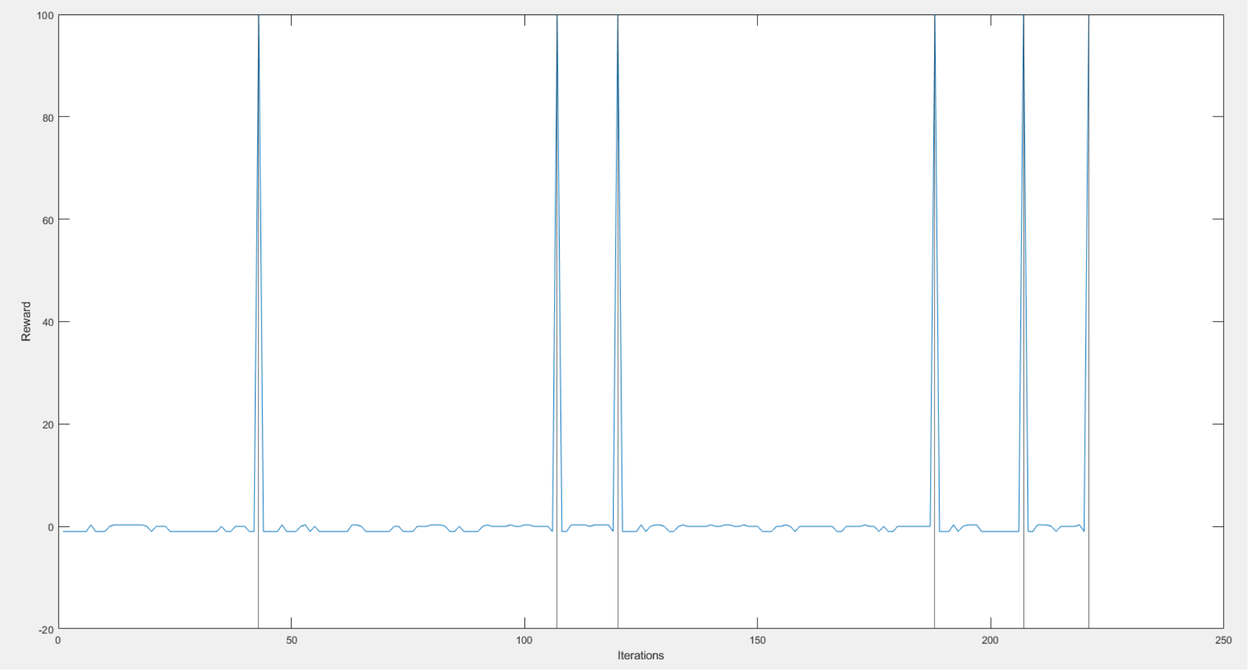
Q table for final episode:



```
Final_Q_Table =

   -1.0322   -0.9966   -0.9830   -1.0407
   -0.5600    0.3290   -0.7905   -0.5575
    1.0000    0.4987   -0.9842    2.0000
   -0.9876    0.0026   -1.0279   -0.7100
   -0.8879   -0.8330   -1.0236   -0.9881
   -0.8355   -0.8928    0.0152   -0.8525
   -0.5625    0.0177    0.2879   -0.9850
   -0.6100    0.0414    0.0302    0.0038
   -0.9997    0.0208   -0.9571    0.0150
   -0.8050   -0.7931   -0.6575    0.0364
   -0.9950   -0.9356    0.1546   -0.7550
    0.0328    0.0283    0.0399   -0.9869
    0.0435    0.0412    0.3088    0.0671
    0.0179    0.4795   -0.7955    0.1070
   -0.6075    3.9878   -0.5600    0.2975
   -0.9787   -0.6625    0.0524   -0.9388
    0.0334    0.0021    0.2400   -0.9483
    0.0498    0.5006    0.2950    0.0311
    0.1196    1.0000    3.9986    0.5611
   -0.6100  100.0000    3.9494    2.0000
   -0.6525   -0.8781    0.0176   -0.7550
    0.0331    0.0169    0.0281   -0.9926
    0.0432    0.0975    0.0753    0.0522
    0.6594    0.4875   95.0500    0.1776
         0         0         0         0
```

Figure 1, reward for all episodes:



# Run #2

- 5x5 grid
- start = 7 (second row from top, second column)
- goal = 19 (fourth row from top, fourth column)
- number of episodes = 6
- number of iterations in each episode = 180

Figure 2, the first learning episode:

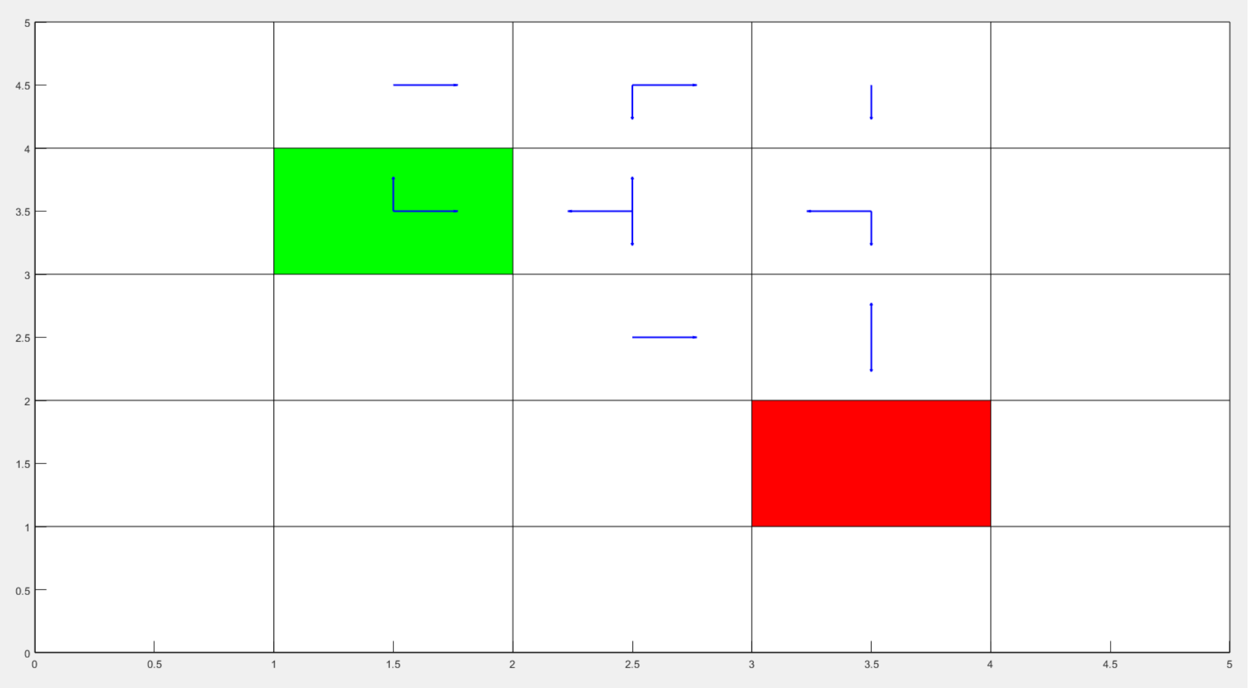
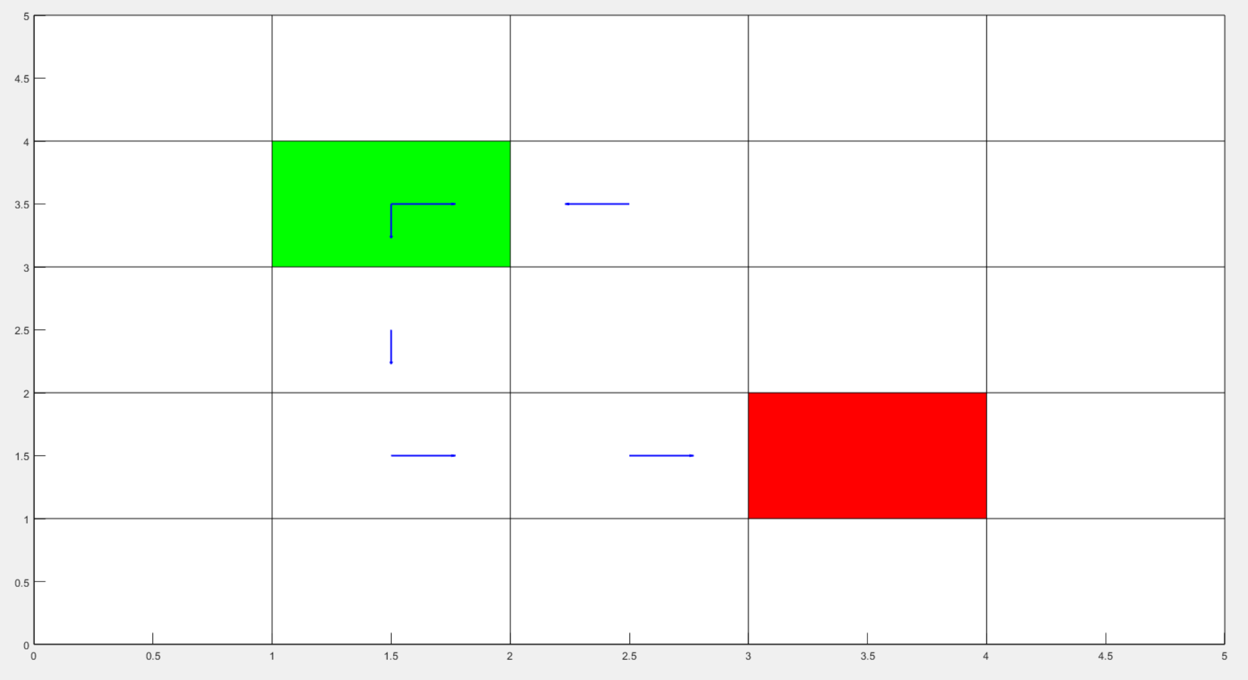
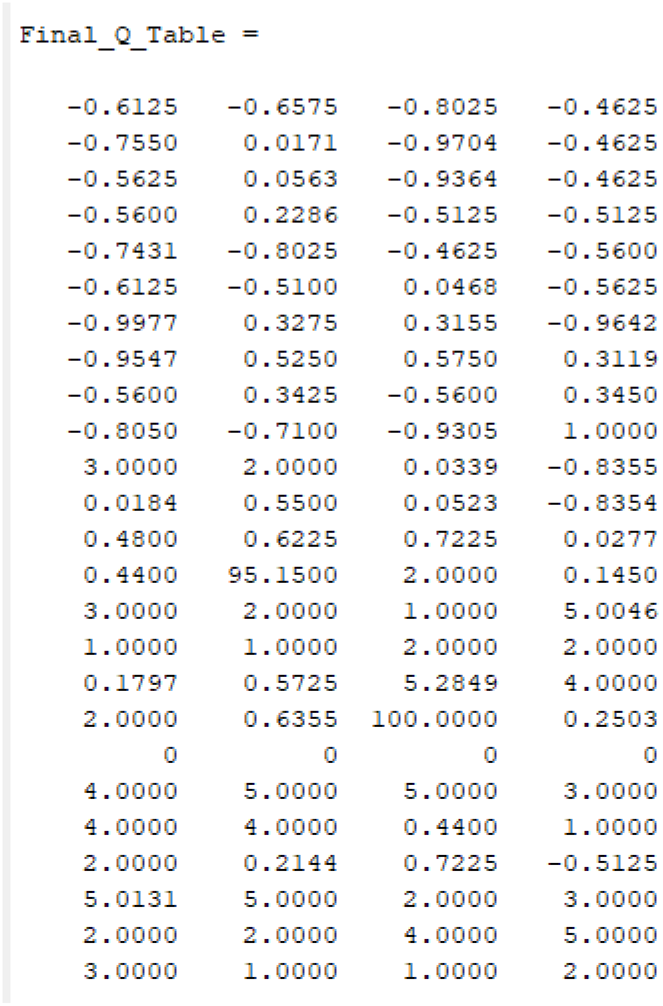


Figure 3, the final learning episode:



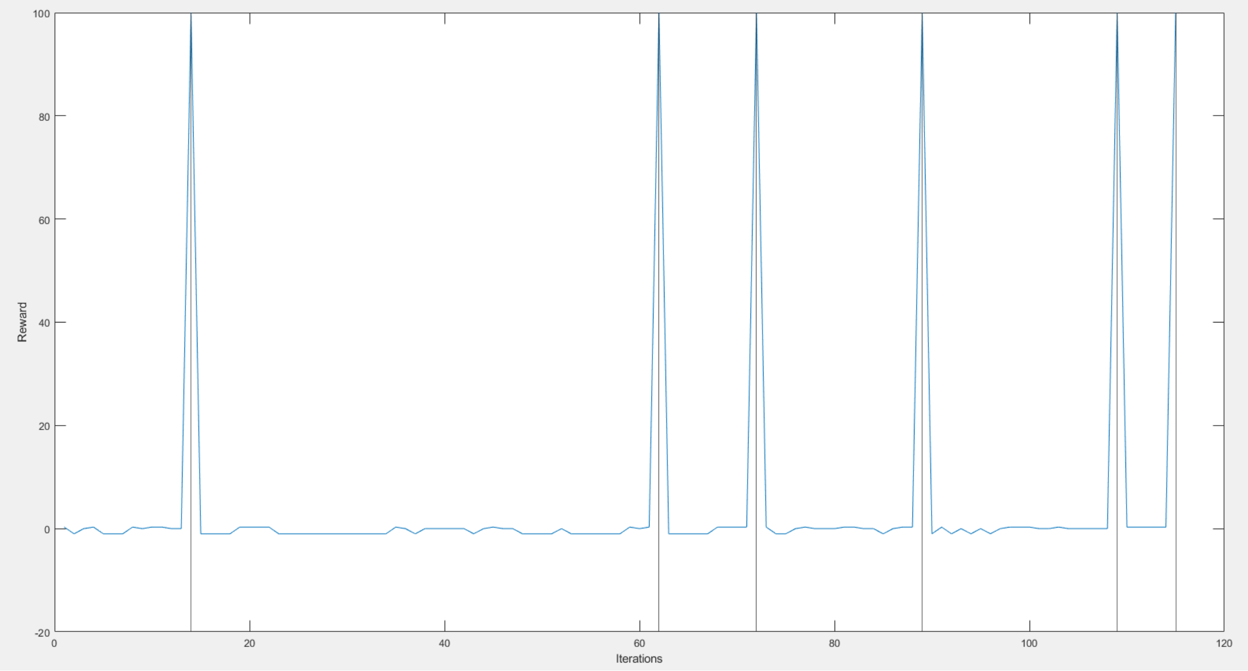
Q table for final episode:



```
Final_Q_Table =

   -0.6125   -0.6575   -0.8025   -0.4625
   -0.7550    0.0171   -0.9704   -0.4625
   -0.5625    0.0563   -0.9364   -0.4625
   -0.5600    0.2286   -0.5125   -0.5125
   -0.7431   -0.8025   -0.4625   -0.5600
   -0.6125   -0.5100    0.0468   -0.5625
   -0.9977    0.3275    0.3155   -0.9642
   -0.9547    0.5250    0.5750    0.3119
   -0.5600    0.3425   -0.5600    0.3450
   -0.8050   -0.7100   -0.9305    1.0000
    3.0000    2.0000    0.0339   -0.8355
    0.0184    0.5500    0.0523   -0.8354
    0.4800    0.6225    0.7225    0.0277
    0.4400   95.1500    2.0000    0.1450
    3.0000    2.0000    1.0000    5.0046
    1.0000    1.0000    2.0000    2.0000
    0.1797    0.5725    5.2849    4.0000
    2.0000    0.6355  100.0000    0.2503
         0         0         0         0
    4.0000    5.0000    5.0000    3.0000
    4.0000    4.0000    0.4400    1.0000
    2.0000    0.2144    0.7225   -0.5125
    5.0131    5.0000    2.0000    3.0000
    2.0000    2.0000    4.0000    5.0000
    3.0000    1.0000    1.0000    2.0000
```

Figure 1, reward for all episodes:



# Run #3

- 5x5 grid
- start = 24 (bottom row, fourth column)
- goal = 2 (top row, second column)
- number of episodes = 6
- number of iterations in each episode = 180

Figure 2, the first learning episode:

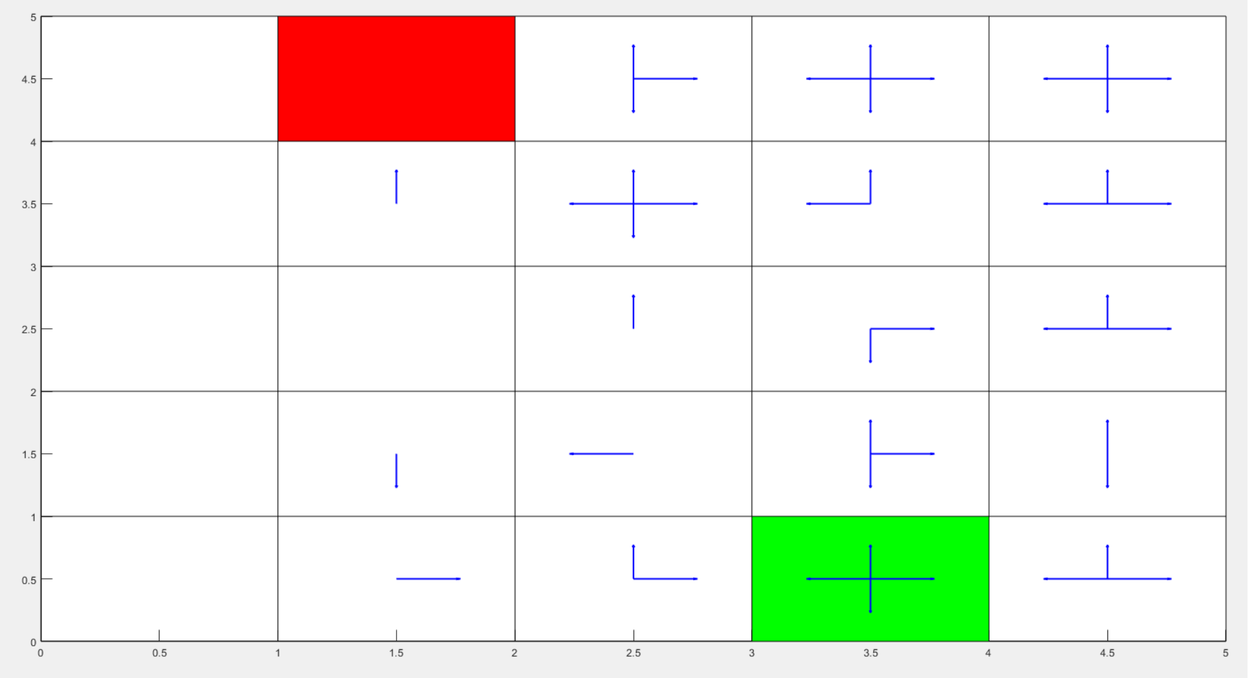
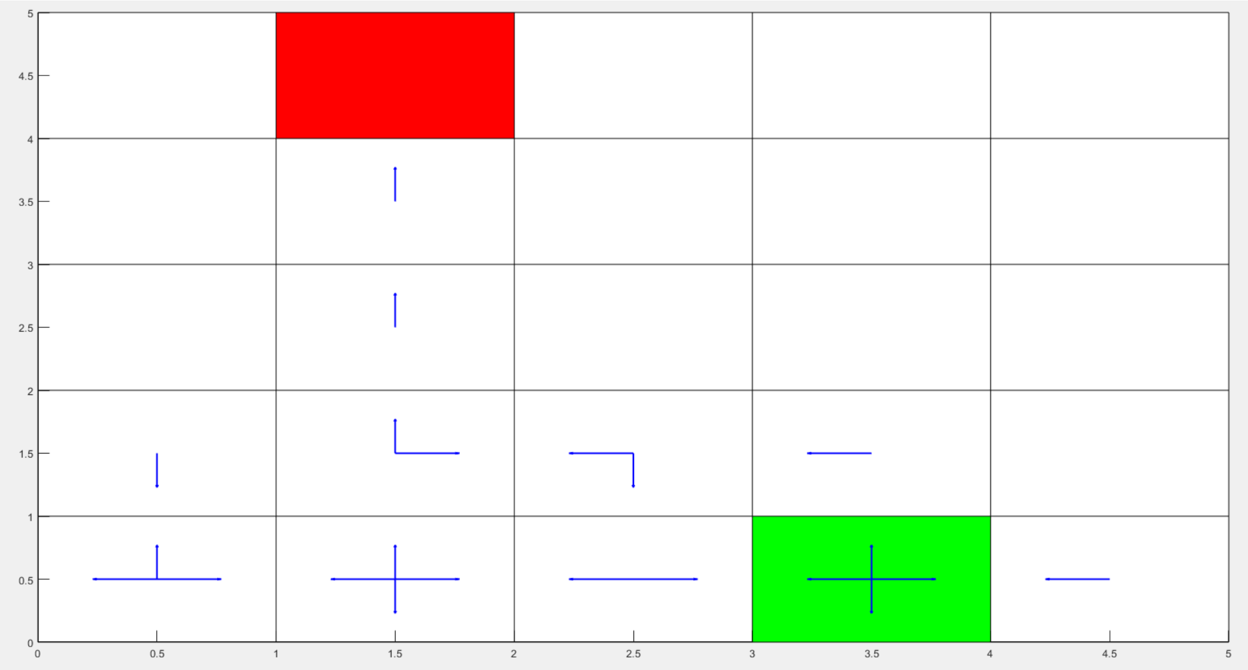
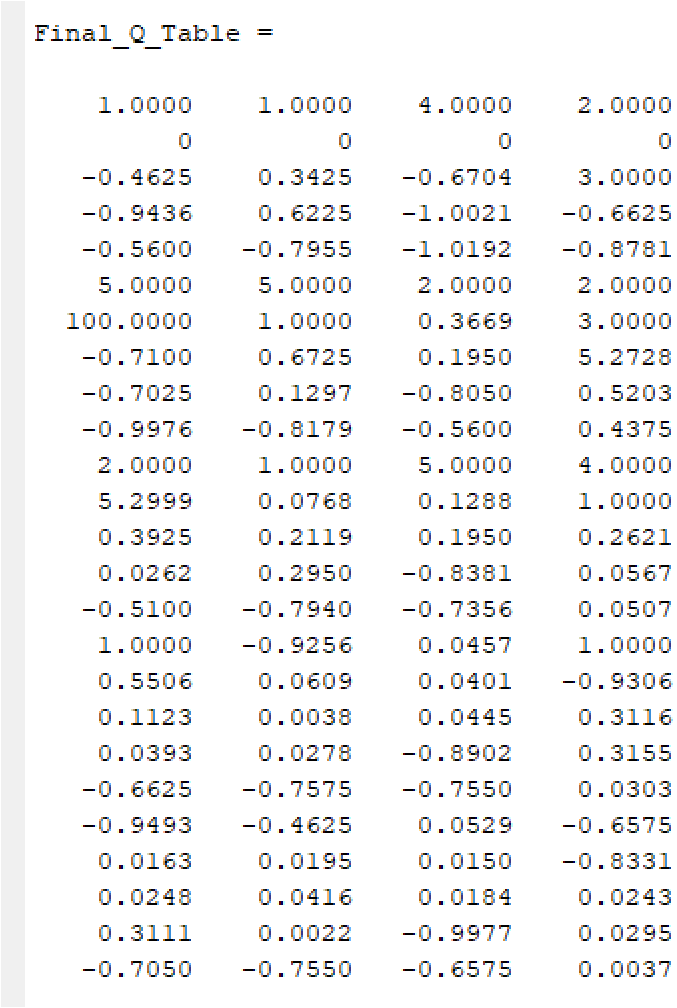


Figure 3, the final learning episode:



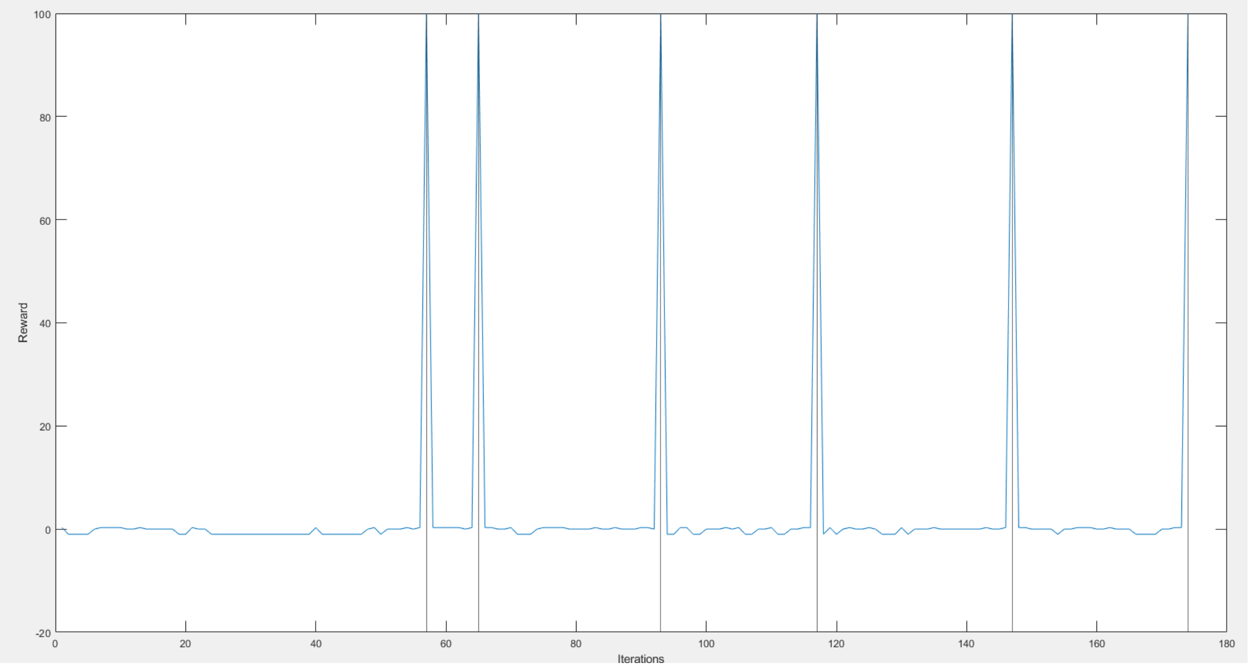
Q table for final episode:



```
Final_Q_Table =

    1.0000    1.0000    4.0000    2.0000
         0         0         0         0
   -0.4625    0.3425   -0.6704    3.0000
   -0.9436    0.6225   -1.0021   -0.6625
   -0.5600   -0.7955   -1.0192   -0.8781
    5.0000    5.0000    2.0000    2.0000
  100.0000    1.0000    0.3669    3.0000
   -0.7100    0.6725    0.1950    5.2728
   -0.7025    0.1297   -0.8050    0.5203
   -0.9976   -0.8179   -0.5600    0.4375
    2.0000    1.0000    5.0000    4.0000
    5.2999    0.0768    0.1288    1.0000
    0.3925    0.2119    0.1950    0.2621
    0.0262    0.2950   -0.8381    0.0567
   -0.5100   -0.7940   -0.7356    0.0507
    1.0000   -0.9256    0.0457    1.0000
    0.5506    0.0609    0.0401   -0.9306
    0.1123    0.0038    0.0445    0.3116
    0.0393    0.0278   -0.8902    0.3155
   -0.6625   -0.7575   -0.7550    0.0303
   -0.9493   -0.4625    0.0529   -0.6575
    0.0163    0.0195    0.0150   -0.8331
    0.0248    0.0416    0.0184    0.0243
    0.3111    0.0022   -0.9977    0.0295
   -0.7050   -0.7550   -0.6575    0.0037
```

Figure 1, reward for all episodes:



# Run #4

- 10x10 grid
- start = 1 (top row, first column)
- goal = 100 (bottom row, last column)
- number of episodes = 6
- number of iterations in each episode = 180

Figure 2, the first learning episode:

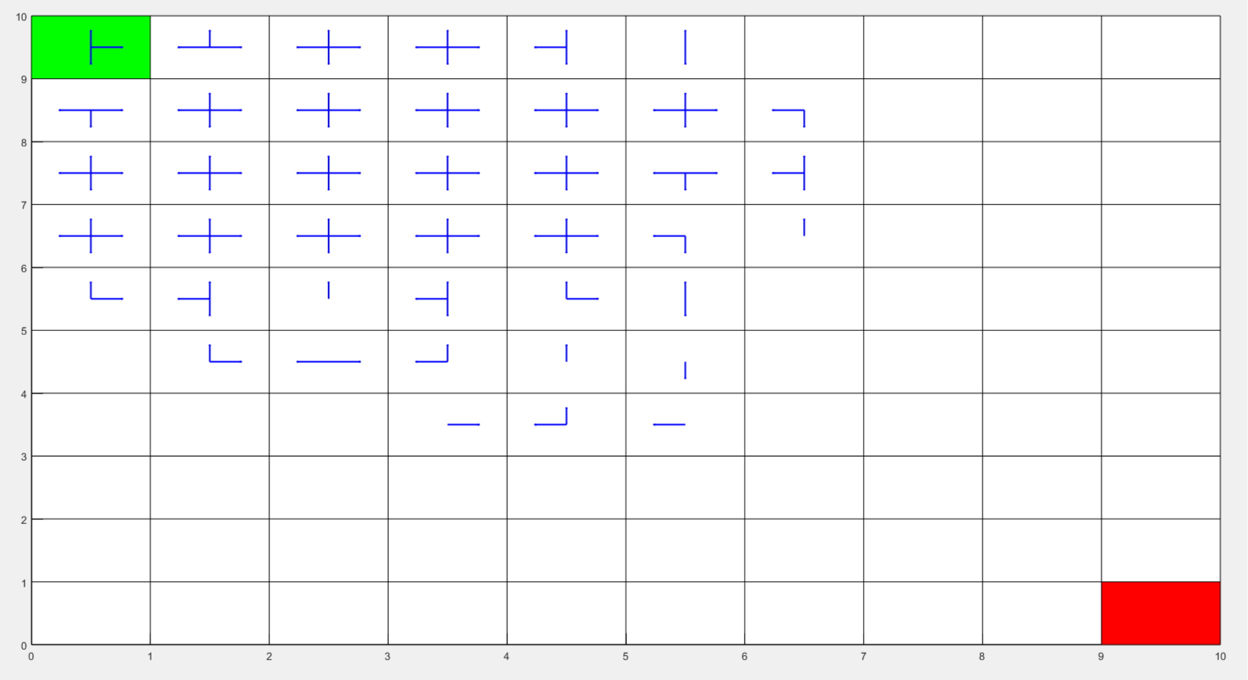
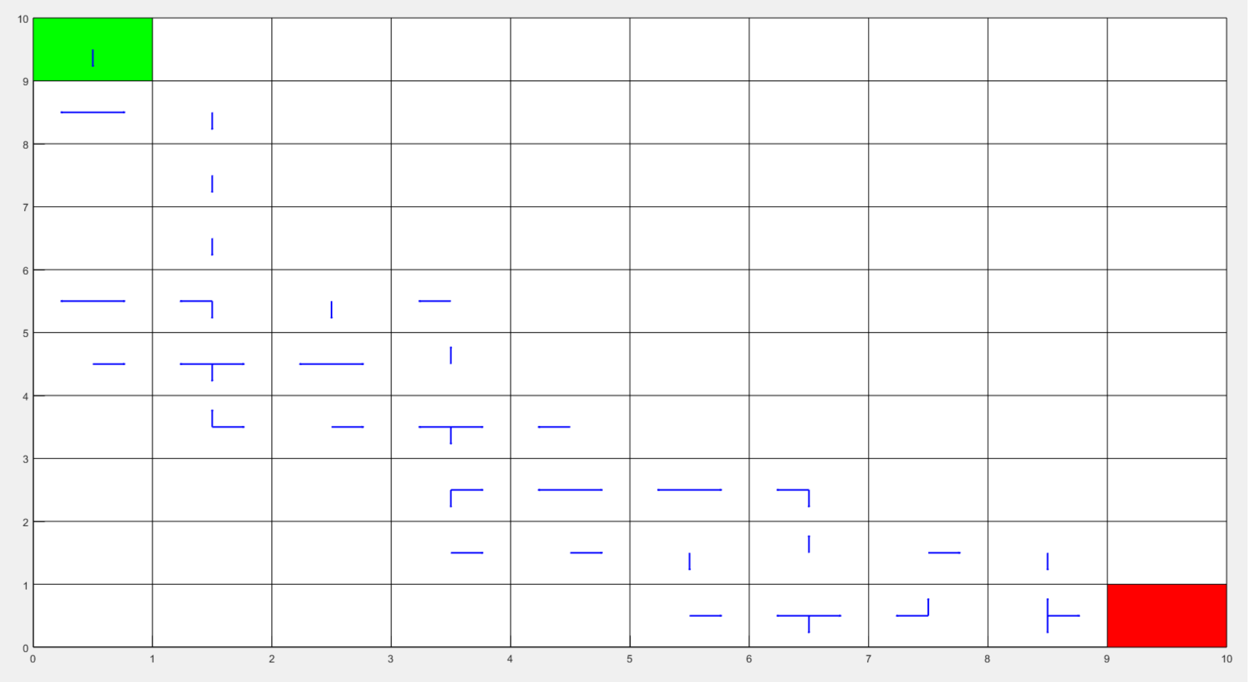
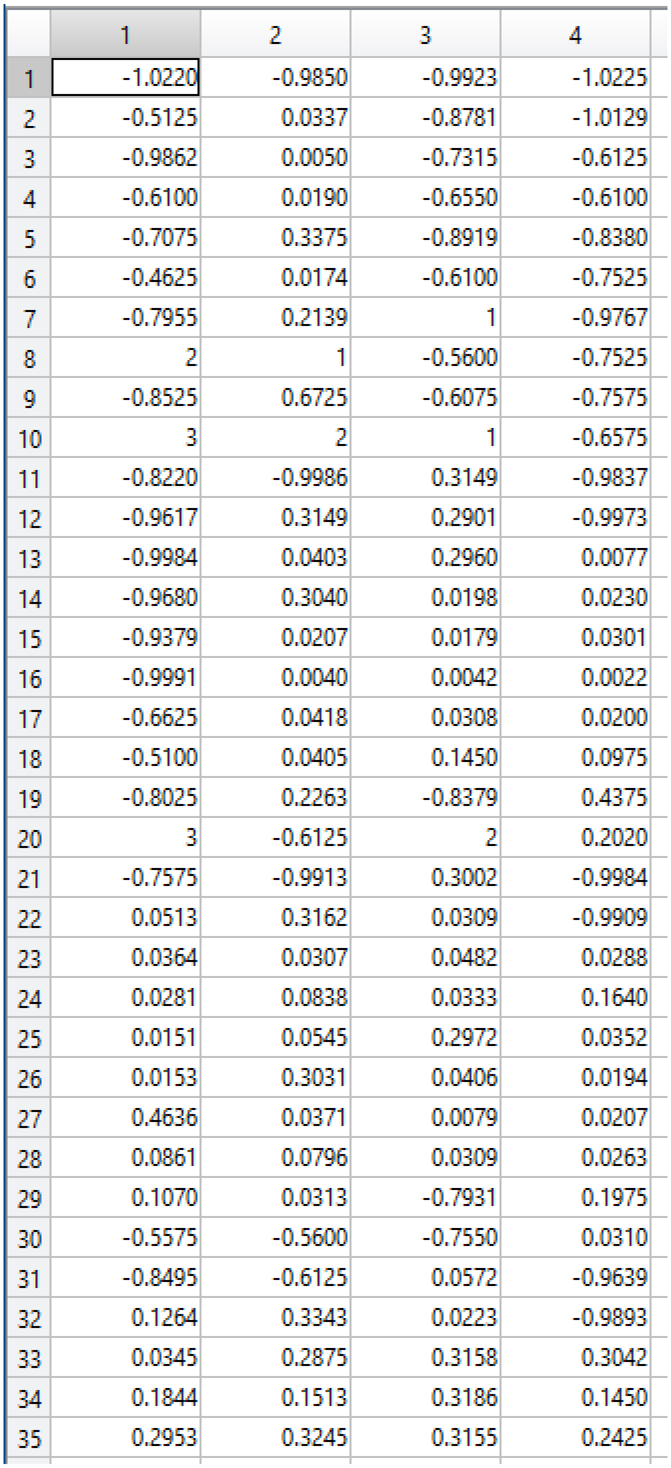


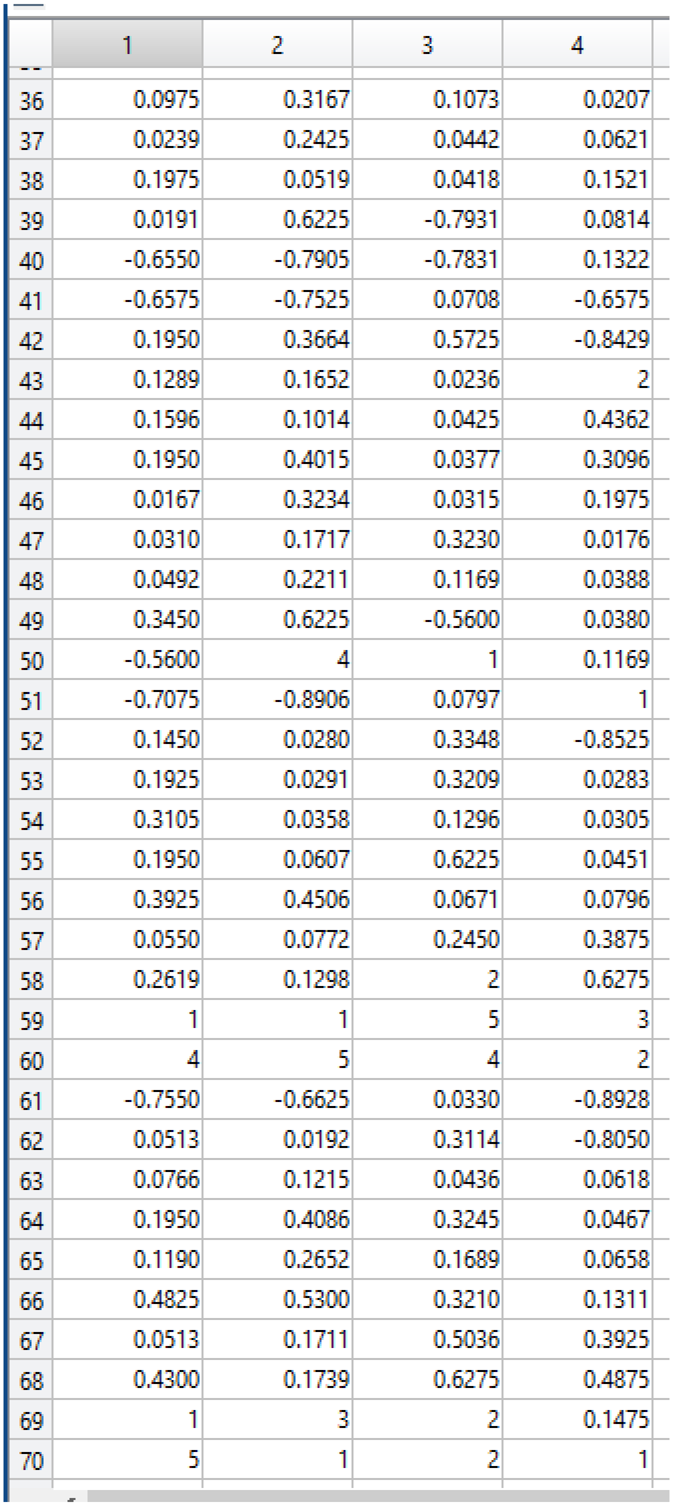
Figure 3, the final learning episode:



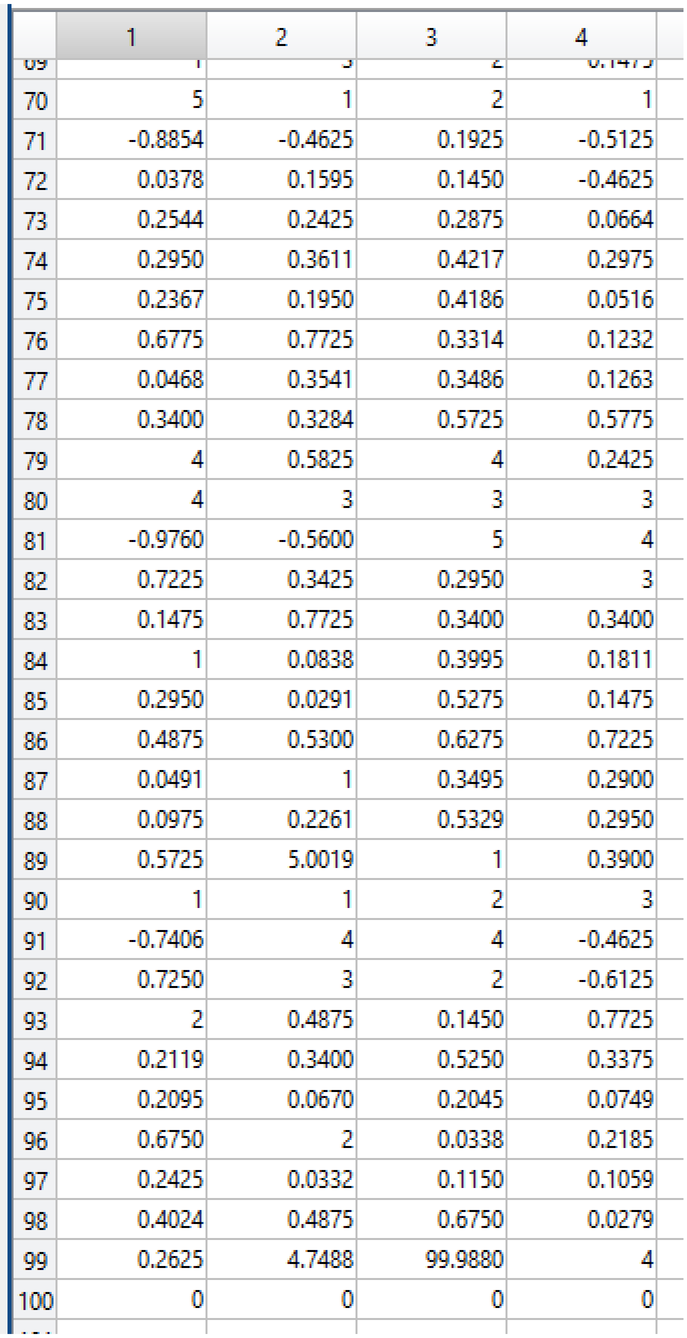
Q table for final episode:



| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | -1.0220 | -0.9850 | -0.9923 | -1.0225 |
| 2 | -0.5125 | 0.0337 | -0.8781 | -1.0129 |
| 3 | -0.9862 | 0.0050 | -0.7315 | -0.6125 |
| 4 | -0.6100 | 0.0190 | -0.6550 | -0.6100 |
| 5 | -0.7075 | 0.3375 | -0.8919 | -0.8380 |
| 6 | -0.4625 | 0.0174 | -0.6100 | -0.7525 |
| 7 | -0.7955 | 0.2139 | 1 | -0.9767 |
| 8 | 2 | 1 | -0.5600 | -0.7525 |
| 9 | -0.8525 | 0.6725 | -0.6075 | -0.7575 |
| 10 | 3 | 2 | 1 | -0.6575 |
| 11 | -0.8220 | -0.9986 | 0.3149 | -0.9837 |
| 12 | -0.9617 | 0.3149 | 0.2901 | -0.9973 |
| 13 | -0.9984 | 0.0403 | 0.2960 | 0.0077 |
| 14 | -0.9680 | 0.3040 | 0.0198 | 0.0230 |
| 15 | -0.9379 | 0.0207 | 0.0179 | 0.0301 |
| 16 | -0.9991 | 0.0040 | 0.0042 | 0.0022 |
| 17 | -0.6625 | 0.0418 | 0.0308 | 0.0200 |
| 18 | -0.5100 | 0.0405 | 0.1450 | 0.0975 |
| 19 | -0.8025 | 0.2263 | -0.8379 | 0.4375 |
| 20 | 3 | -0.6125 | 2 | 0.2020 |
| 21 | -0.7575 | -0.9913 | 0.3002 | -0.9984 |
| 22 | 0.0513 | 0.3162 | 0.0309 | -0.9909 |
| 23 | 0.0364 | 0.0307 | 0.0482 | 0.0288 |
| 24 | 0.0281 | 0.0838 | 0.0333 | 0.1640 |
| 25 | 0.0151 | 0.0545 | 0.2972 | 0.0352 |
| 26 | 0.0153 | 0.3031 | 0.0406 | 0.0194 |
| 27 | 0.4636 | 0.0371 | 0.0079 | 0.0207 |
| 28 | 0.0861 | 0.0796 | 0.0309 | 0.0263 |
| 29 | 0.1070 | 0.0313 | -0.7931 | 0.1975 |
| 30 | -0.5575 | -0.5600 | -0.7550 | 0.0310 |
| 31 | -0.8495 | -0.6125 | 0.0572 | -0.9639 |
| 32 | 0.1264 | 0.3343 | 0.0223 | -0.9893 |
| 33 | 0.0345 | 0.2875 | 0.3158 | 0.3042 |
| 34 | 0.1844 | 0.1513 | 0.3186 | 0.1450 |
| 35 | 0.2953 | 0.3245 | 0.3155 | 0.2425 |

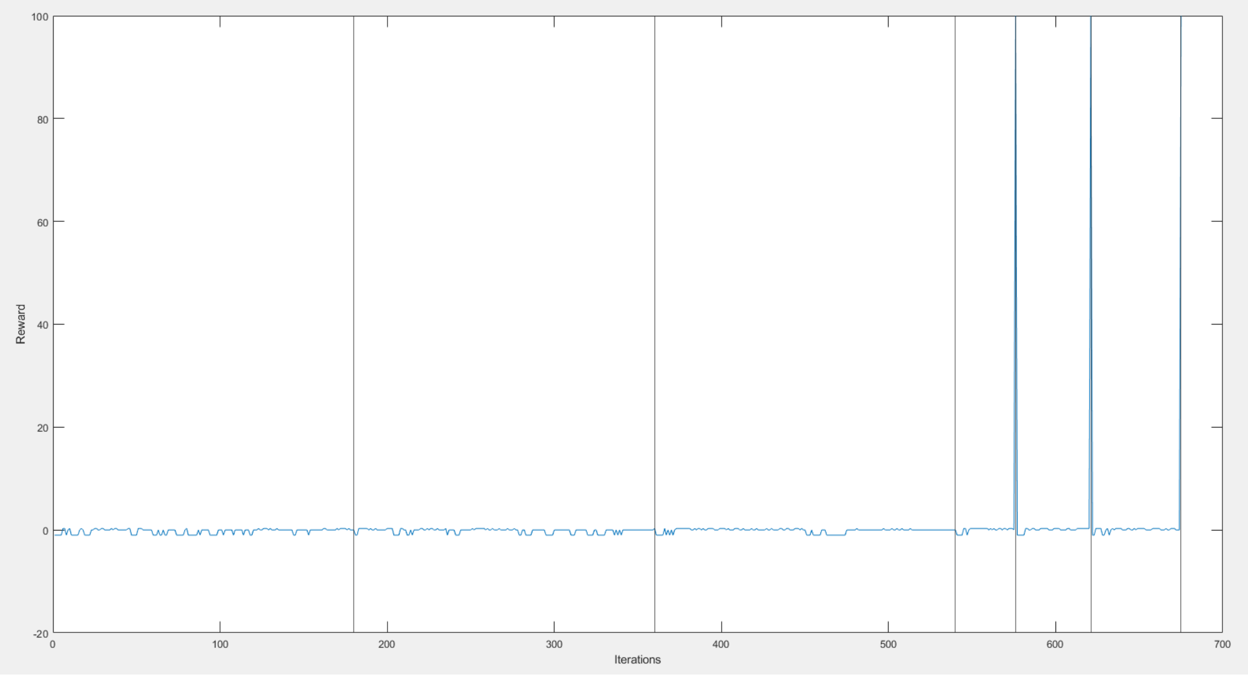


| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 36 | 0.0975 | 0.3167 | 0.1073 | 0.0207 |
| 37 | 0.0239 | 0.2425 | 0.0442 | 0.0621 |
| 38 | 0.1975 | 0.0519 | 0.0418 | 0.1521 |
| 39 | 0.0191 | 0.6225 | -0.7931 | 0.0814 |
| 40 | -0.6550 | -0.7905 | -0.7831 | 0.1322 |
| 41 | -0.6575 | -0.7525 | 0.0708 | -0.6575 |
| 42 | 0.1950 | 0.3664 | 0.5725 | -0.8429 |
| 43 | 0.1289 | 0.1652 | 0.0236 | 2 |
| 44 | 0.1596 | 0.1014 | 0.0425 | 0.4362 |
| 45 | 0.1950 | 0.4015 | 0.0377 | 0.3096 |
| 46 | 0.0167 | 0.3234 | 0.0315 | 0.1975 |
| 47 | 0.0310 | 0.1717 | 0.3230 | 0.0176 |
| 48 | 0.0492 | 0.2211 | 0.1169 | 0.0388 |
| 49 | 0.3450 | 0.6225 | -0.5600 | 0.0380 |
| 50 | -0.5600 | 4 | 1 | 0.1169 |
| 51 | -0.7075 | -0.8906 | 0.0797 | 1 |
| 52 | 0.1450 | 0.0280 | 0.3348 | -0.8525 |
| 53 | 0.1925 | 0.0291 | 0.3209 | 0.0283 |
| 54 | 0.3105 | 0.0358 | 0.1296 | 0.0305 |
| 55 | 0.1950 | 0.0607 | 0.6225 | 0.0451 |
| 56 | 0.3925 | 0.4506 | 0.0671 | 0.0796 |
| 57 | 0.0550 | 0.0772 | 0.2450 | 0.3875 |
| 58 | 0.2619 | 0.1298 | 2 | 0.6275 |
| 59 | 1 | 1 | 5 | 3 |
| 60 | 4 | 5 | 4 | 2 |
| 61 | -0.7550 | -0.6625 | 0.0330 | -0.8928 |
| 62 | 0.0513 | 0.0192 | 0.3114 | -0.8050 |
| 63 | 0.0766 | 0.1215 | 0.0436 | 0.0618 |
| 64 | 0.1950 | 0.4086 | 0.3245 | 0.0467 |
| 65 | 0.1190 | 0.2652 | 0.1689 | 0.0658 |
| 66 | 0.4825 | 0.5300 | 0.3210 | 0.1311 |
| 67 | 0.0513 | 0.1711 | 0.5036 | 0.3925 |
| 68 | 0.4300 | 0.1739 | 0.6275 | 0.4875 |
| 69 | 1 | 3 | 2 | 0.1475 |
| 70 | 5 | 1 | 2 | 1 |



|  | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 69 | 1 | 3 | 2 | 0.1475 |
| 70 | 5 | 1 | 2 | 1 |
| 71 | -0.8854 | -0.4625 | 0.1925 | -0.5125 |
| 72 | 0.0378 | 0.1595 | 0.1450 | -0.4625 |
| 73 | 0.2544 | 0.2425 | 0.2875 | 0.0664 |
| 74 | 0.2950 | 0.3611 | 0.4217 | 0.2975 |
| 75 | 0.2367 | 0.1950 | 0.4186 | 0.0516 |
| 76 | 0.6775 | 0.7725 | 0.3314 | 0.1232 |
| 77 | 0.0468 | 0.3541 | 0.3486 | 0.1263 |
| 78 | 0.3400 | 0.3284 | 0.5725 | 0.5775 |
| 79 | 4 | 0.5825 | 4 | 0.2425 |
| 80 | 4 | 3 | 3 | 3 |
| 81 | -0.9760 | -0.5600 | 5 | 4 |
| 82 | 0.7225 | 0.3425 | 0.2950 | 3 |
| 83 | 0.1475 | 0.7725 | 0.3400 | 0.3400 |
| 84 | 1 | 0.0838 | 0.3995 | 0.1811 |
| 85 | 0.2950 | 0.0291 | 0.5275 | 0.1475 |
| 86 | 0.4875 | 0.5300 | 0.6275 | 0.7225 |
| 87 | 0.0491 | 1 | 0.3495 | 0.2900 |
| 88 | 0.0975 | 0.2261 | 0.5329 | 0.2950 |
| 89 | 0.5725 | 5.0019 | 1 | 0.3900 |
| 90 | 1 | 1 | 2 | 3 |
| 91 | -0.7406 | 4 | 4 | -0.4625 |
| 92 | 0.7250 | 3 | 2 | -0.6125 |
| 93 | 2 | 0.4875 | 0.1450 | 0.7725 |
| 94 | 0.2119 | 0.3400 | 0.5250 | 0.3375 |
| 95 | 0.2095 | 0.0670 | 0.2045 | 0.0749 |
| 96 | 0.6750 | 2 | 0.0338 | 0.2185 |
| 97 | 0.2425 | 0.0332 | 0.1150 | 0.1059 |
| 98 | 0.4024 | 0.4875 | 0.6750 | 0.0279 |
| 99 | 0.2625 | 4.7488 | 99.9880 | 4 |
| 100 | 0 | 0 | 0 | 0 |

Figure 1, reward for all episodes:



# Conclusion

In all runs, the number of actions taken by the robot to reach the goal is reduced from the first learning episode to the final learning episode. You can see this in the number of arrows in the first learning episode, compared to the lower number of arrows in the final episode for each run.