TML Assignment - SARSA, Expected SARSA and Q Learning

Rehas Mehar Kaur Sachdeva - 201401102

Run on a 4 by 4 grid world with (0,0) as start state, (3,3) as goal state and (1,2) as a bad state. Notation - 0 : Left, 1 : Right, 2 : Up, 3 : Down.

SARSA

Epsilon = 0.05

Optimal Policy

```
[[ 2.  1.  2.  1.]
 [ 2.  1.  1.  1.]
 [ 2.  1.  1.  1.]
 [ 0.  2.  2.  0.]]
```

Optimal Value functions

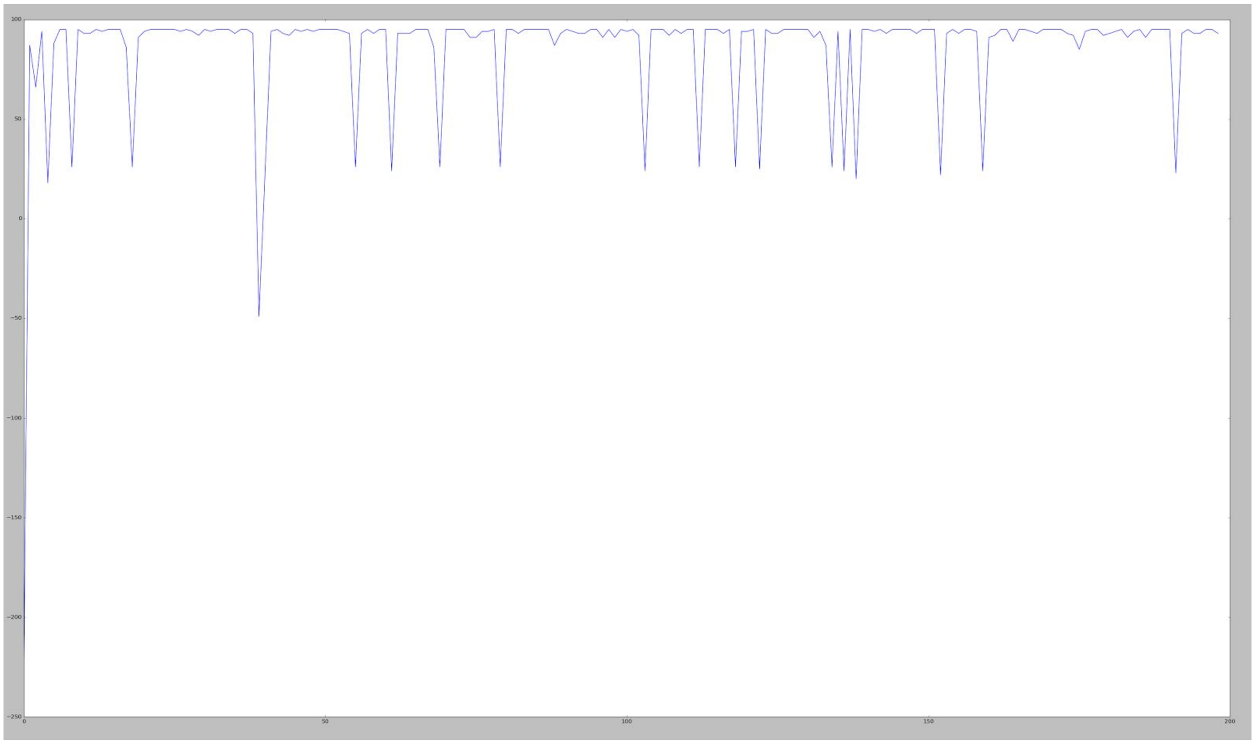
```
[[[  2.68517441e+00   3.67435563e+00   3.08867837e+01   1.28790810e+01]
  [  1.41845698e+01   4.13431219e+01  -5.07410469e-02   1.00140419e+01]
  [ -1.82335000e+00  -3.61131971e+01   1.25919132e+01  -1.99000000e+00]
  [ -1.82335000e+00   4.75906891e+01  -8.71507990e+00  -1.95575500e+01]]

 [[ -2.26671653e+00  -2.10203830e+00   3.45395743e+01  -2.24905330e+00]
  [  1.85542633e+01   5.73219806e+01  -7.00000000e+01   1.64597968e+01]
  [  7.73984791e-01   6.77842623e+01  -1.99000000e+00  -1.00000000e+00]
  [ -3.55000000e+01   7.80660069e+01  -1.00000000e+00  -7.00000000e+01]]

 [[  9.59794100e-01  -1.51826748e+00   5.97021704e+01   1.51970297e+01]
  [  2.65438447e+01   7.87243700e+01   3.85088000e+01   1.28439728e+01]
  [ -3.81871770e+01   8.87428588e+01   3.52402639e+01  -2.97010000e+00]
  [ -1.00000000e+00   9.61713390e+01   0.00000000e+00   0.00000000e+00]]

 [[  2.89915996e+01   4.64597000e+00  -1.00000000e+00  -1.49500000e+00]
  [  4.26441397e+01   4.22169819e+01   8.94773669e+01   2.33431170e+01]
  [  6.16804729e+01   6.81159648e+01   9.71796413e+01   7.67198847e+01]
  [  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00]]]
```

Reward per episode



Epsilon = 0.2

Optimal Policy

[[1. 3. 2. 1.]
 [1. 1. 1. 1.]
 [2. 2. 1. 1.]
 [2. 2. 2. 0.]]

Optimal Value functions

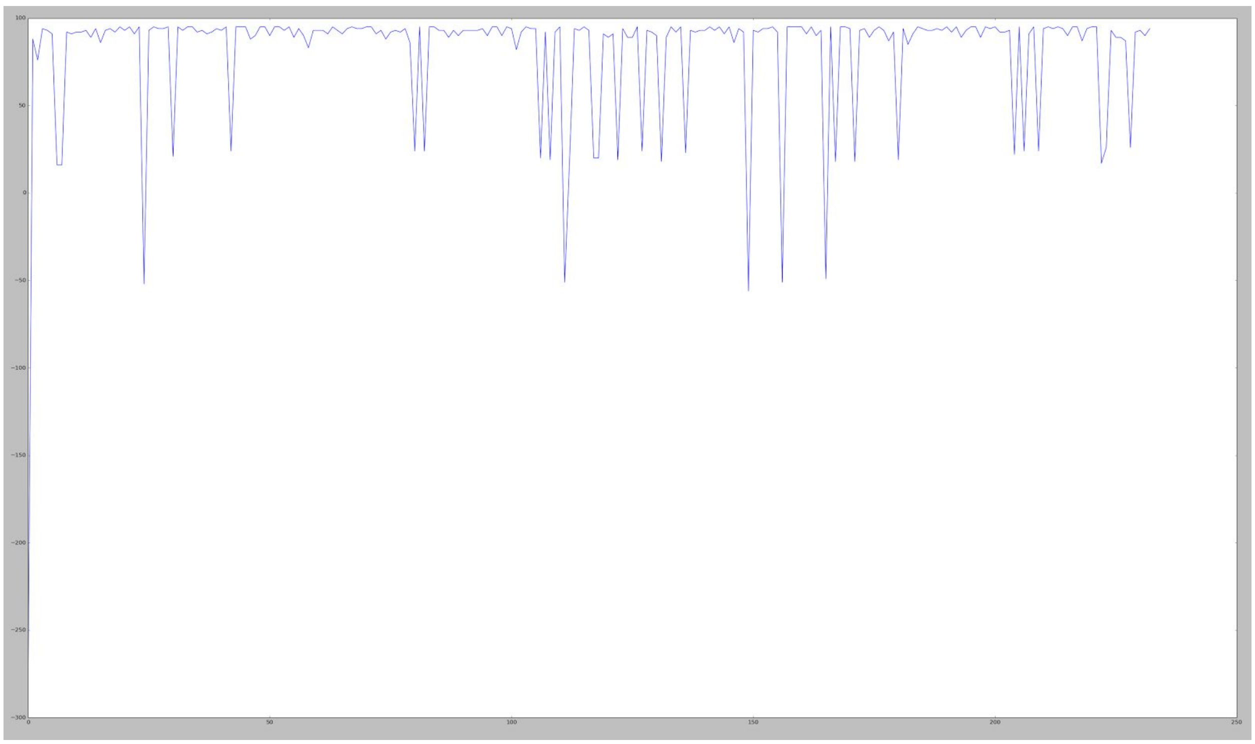
[[[1.05913734 12.72231838 -0.18762947   2.17831198]
  [-2.27628048 -5.17786607 -6.92610812   7.00936106]
  [-1.48071775 -36.48520349   4.1860226   -1.99   ]
  [-1.495   22.9211307   -1.92626875 -1.99   ]]

 [[5.23429572 30.23934273 24.61642063 10.5466985]
  [-7.92555031 43.42754466 -42.99891544 -1.83978963]
  [-1.   64.88006128 28.10848544 35.20503452]
  [7.8693247   68.75175103 32.73128652 -60.13877812]]

```
[[ 12.83641056  14.88686271  50.12586225  13.39634576]
 [ 31.49178651  47.58000115  68.41118569  27.42709646]
 [-35.23093391  88.72288928  77.5492931   49.40030817]
 [ 21.11119901  95.99739027  54.83824483  87.66444648]]

[[ -2.22438827  -2.62493382  42.50285053   7.93723796]
 [ 36.49980811  32.20981207  73.82058494  -1.99      ]
 [ 41.86620214  67.9556181   96.48201792  59.4535089 ]
 [  0.           0.           0.           0.        ]]]
```

Reward per episode



Effect of Epsilon on Rate of convergence and optimal policy

Takes slightly longer (~40 more iterations) to converge for epsilon=0.2. Epsilon=0.05 seems to converge to better policy, but the final reward they converge to are almost same.

Expected SARSA

Epsilon = 0.05

Optimal Policy

```
[[ 1.  1.  2.  1.]
 [ 1.  1.  1.  1.]
 [ 1.  1.  3.  1.]
 [ 2.  2.  2.  0.]]
```

Optimal Value functions

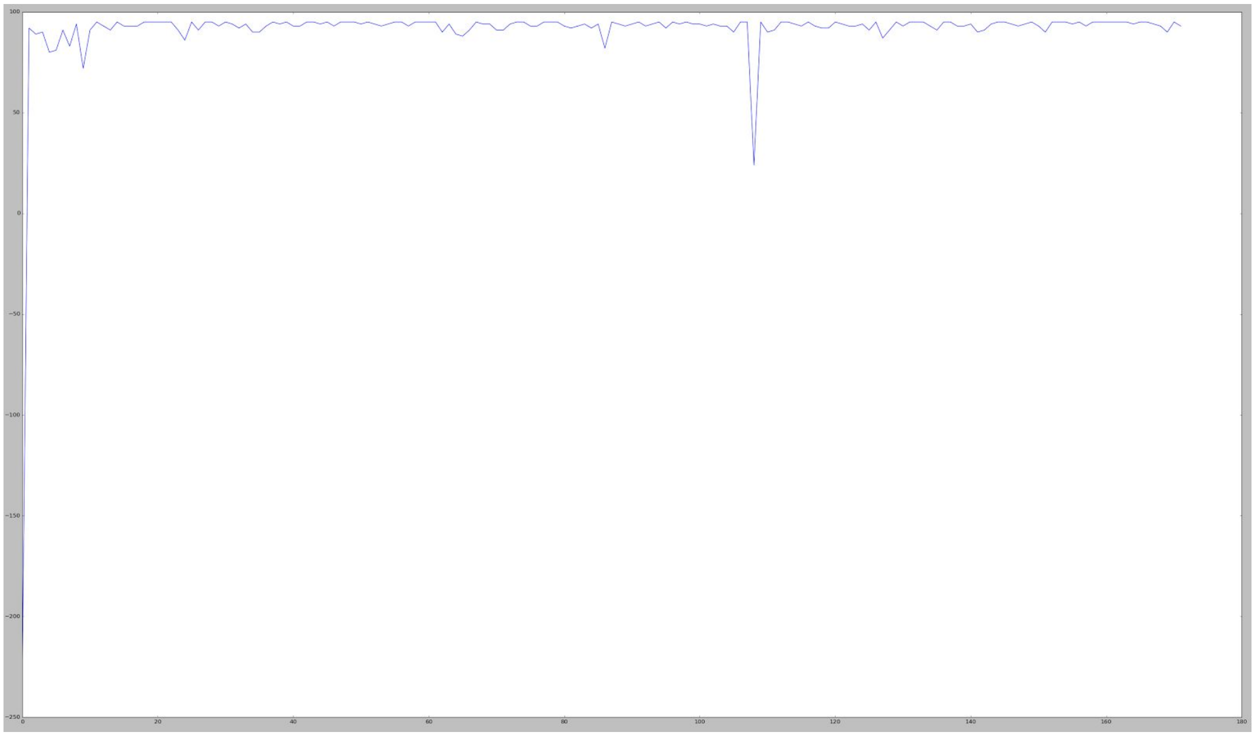
```
[[[   4.44956751   25.69825885   -2.13382104    2.35475723]
  [  -2.49587152   16.51568866   -2.27035961   -2.41292394]
  [  -2.55439801  -70.            4.19124546   -2.58330734]
  [  -2.01248152   32.32477917   -2.09420559   -2.70015564]]

 [[   2.06167022   45.97037987    3.39090972    2.80990192]
  [  -2.01934187   45.02987161  -70.51765665   -1.53820316]
  [  -1.87870157   26.47260263   -1.01268223   -1.89159493]
  [  -1.02482657   75.21985193   -1.03139977  -70.04815587]]

 [[  16.77443987   66.1993347    34.8825798    42.15782294]
  [  -1.02505723   76.79363319   -1.87935171   -2.86802905]
  [ -70.03562393   -1.0231003    -1.01268508   62.52473399]
  [  -1.02505723  100.            0.            0.        ]]

 [[  28.8117151    24.26006905   80.65656162   36.91371893]
  [  58.57520113   57.13330652   90.25652742   62.23570261]
  [  38.15310606   60.95438523   97.26729612   85.62567911]
  [   0.            0.            0.            0.        ]]]
```

Reward per episode



Epsilon = 0.2

Optimal Policy

```
[[ 1.  3.  0.  0.]
 [ 1.  3.  1.  1.]
 [ 1.  1.  1.  1.]
 [ 2.  2.  2.  0.]]
```

Optimal Value functions

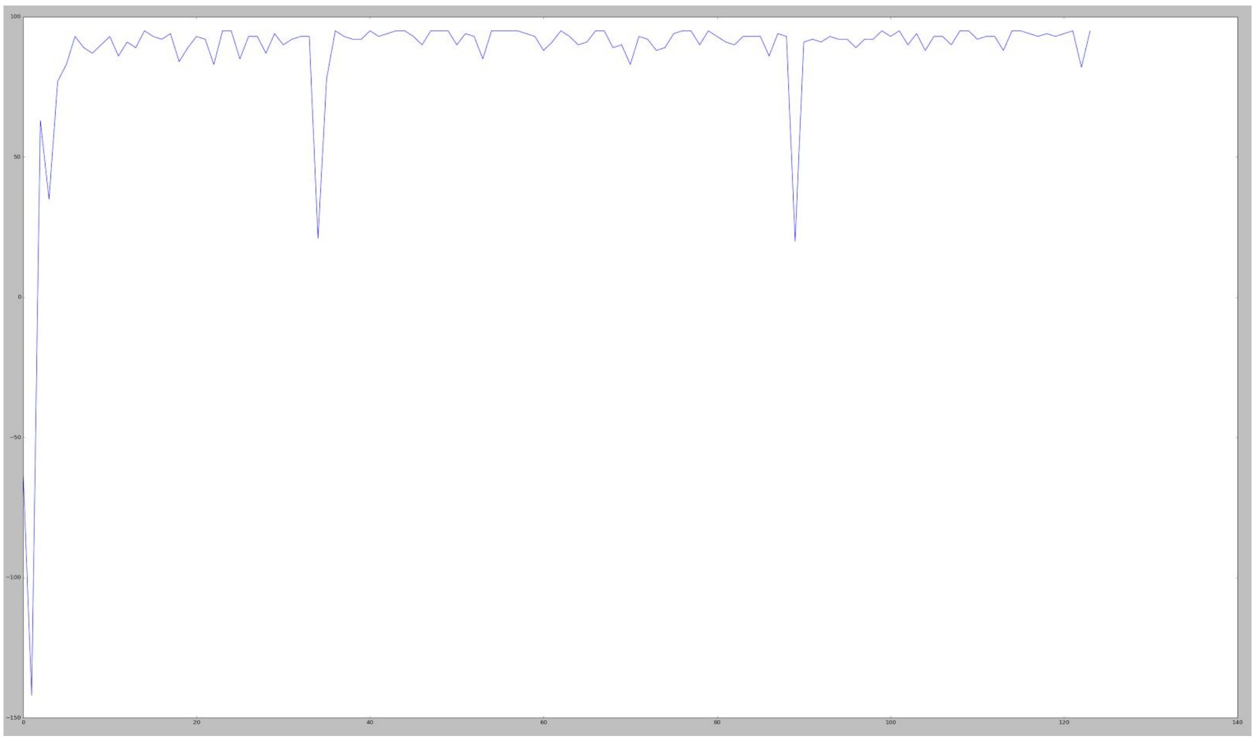
```
[[[  -1.34080785    6.7896073    -3.26989679   -1.40539162]
  [  -3.28717199   -4.39746642   -4.51817538   -1.06818724]
  [  -1.02475     -70.0495       -1.88716135   -1.22364968]
  [  -1.95020693   -3.23995545   -2.49041859   -3.30300988]]

 [[  -1.75848928   21.10152633   -4.56811208    7.09622304]
  [  -1.61216682  -36.17222375  -60.52194081    5.17984448]
  [  -2.78937302   23.34716558   -1.10145025   -4.21576581]
  [  -1.58803254   48.92438709   -1.12871114  -70.32503929]]
```

```
[[  13.87728692   39.92695972   34.75632287    9.95958937]
 [  -5.9085166    63.4464957    -4.99500515   11.28772105]
 [ -47.96168312   80.25907572   41.02427488   -1.41434934]
 [  14.44276596  100.           -1.0495        0.        ]]

[[  14.50711651   15.71168546   61.24020121   27.95117894]
 [  29.96042575   41.04120089   84.12986366   29.57971726]
 [  50.28974817   66.34947216   96.9731518    63.52371588]
 [   0.            0.            0.            0.        ]]]
```

Reward per episode



Effect of Epsilon on Rate of convergence and optimal policy

Takes about 35 more iterations for epsilon=0.05 to converge but it converges to a significantly better policy than for epsilon=0.2, with higher reward at convergence.

Q Learning

Epsilon = 0.05

Optimal Policy

```
[[ 1.  1.  3.  0.]
 [ 2.  1.  1.  1.]
 [ 2.  2.  2.  1.]
 [ 1.  2.  2.  0.]]
```

Optimal Value functions

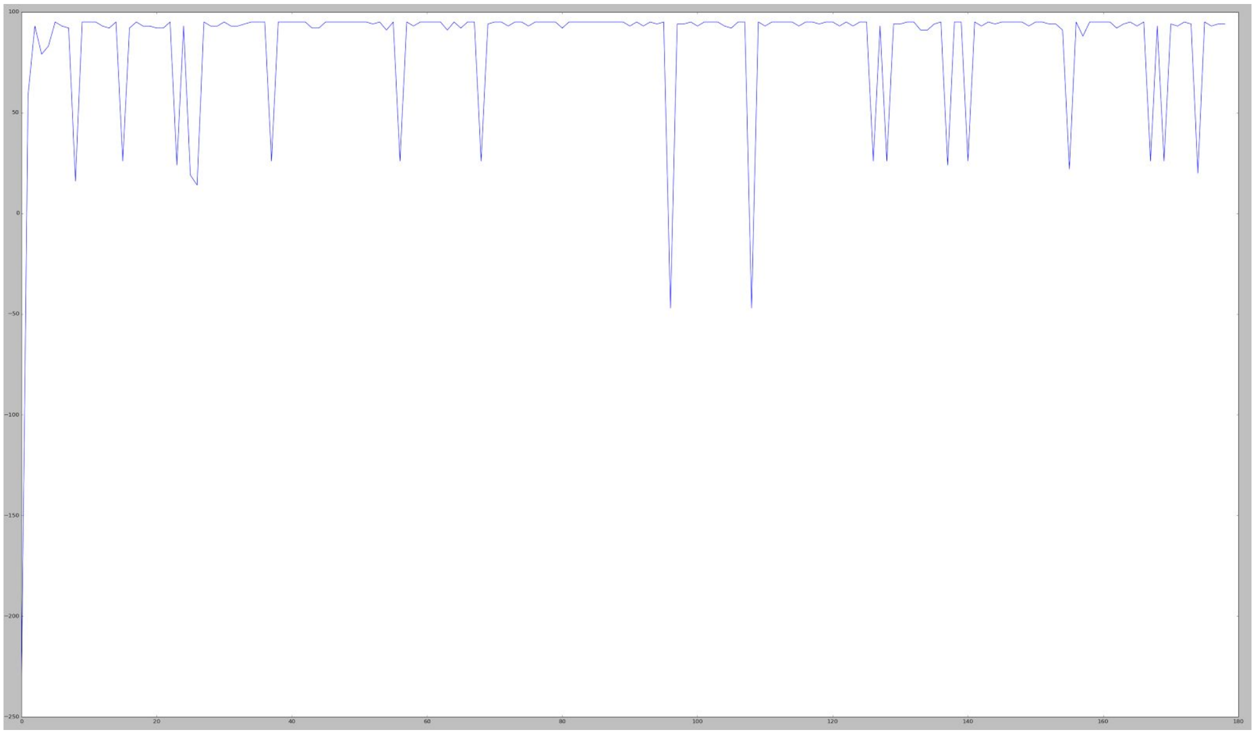
```
[[[  3.79105358  29.68351057   6.92033283   0.76459648]
  [ -2.06490849  38.11166321  -1.99        12.59404944]
  [ -1.82335    -70.          -1.66        -0.51597695]
  [ -1.33        -1.740025    -1.495       -1.76589147]]

 [[ -1.33383856   3.70834858  49.75447866  13.97230549]
  [ 15.06349965  68.40910654 -17.33234627  16.58639499]
  [ -1.39041117  78.94074807  -1.          -1.        ]
  [ -1.495       -1.          -1.         -70.        ]]

 [[ -1.79068      1.53607982  53.51924235  -2.01153745]
  [ 22.90671556  14.35489668  85.31616926   6.46395398]
  [-30.49165888  50.70966156  94.88765344  40.57804717]
  [ -1.          99.4724454    0.          90.46687647]]

 [[ -1.99        -1.82335     -1.95535     -2.06879163]
  [ -1.6175125   -1.82335     70.17828518  -1.8217    ]
  [ -1.495       -1.495       94.33850923  -1.99      ]
  [  0.           0.           0.           0.        ]]]
```

Reward per episode



Epsilon = 0.2

Optimal Policy

```
[[ 1.  1.  3.  1.]
 [ 2.  1.  1.  1.]
 [ 2.  1.  2.  1.]
 [ 2.  2.  2.  0.]]
```

Optimal Value functions

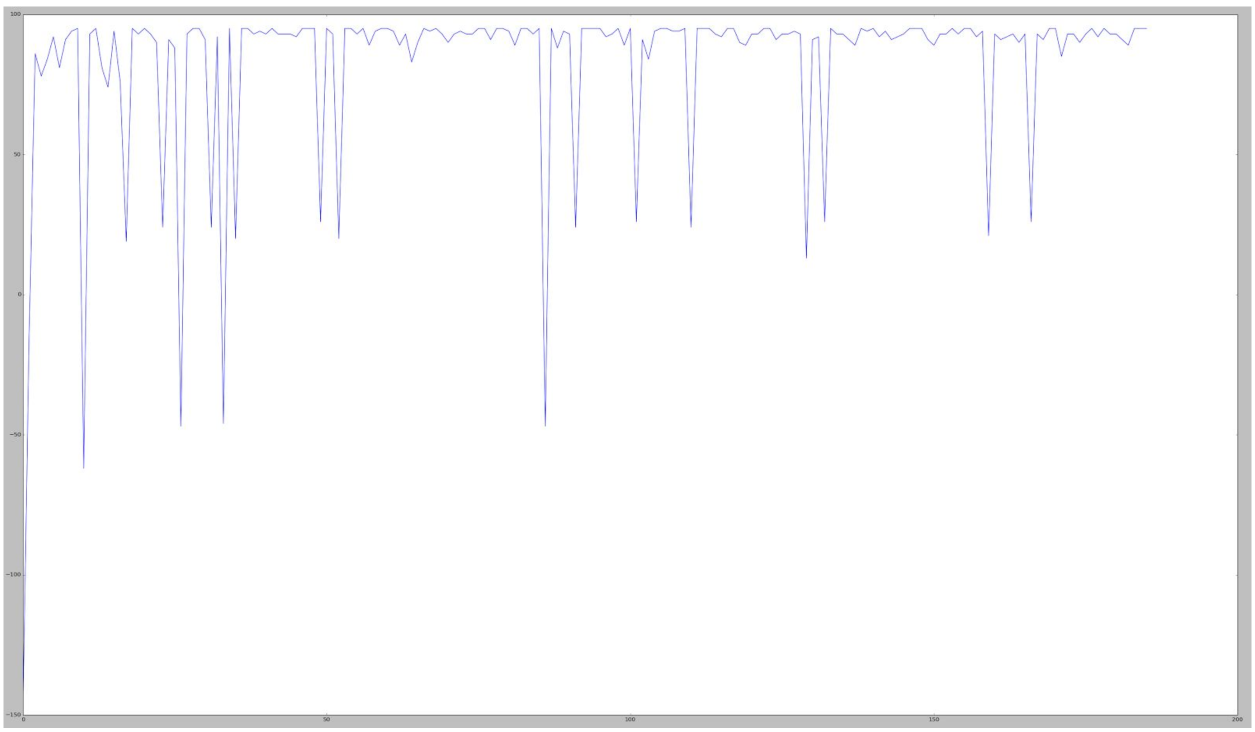
```
[[[  5.61071599e+00   3.01930433e+01   7.87716373e+00   6.38436679e+00]
  [ -2.22338008e+00   3.34282753e+01  -2.23502500e+00   7.63119049e+00]
  [ -2.53176071e+00  -3.55000000e+01  -7.00000000e+01   5.20698362e+00]
  [ -1.49500000e+00  -1.00000000e+00  -1.00000000e+00  -2.16177138e+00]]

 [[  1.74484924e+01   3.28710250e+01   4.68344693e+01   2.49311646e+01]
  [  9.42524509e+00   6.49351582e+01  -2.09308815e+01   2.97049233e+01]
  [ -1.55711675e+00   6.10135830e+01   2.07562674e+01   2.33080930e+01]
  [ -1.49500000e+00   4.73242402e+01  -1.00000000e+00  -4.53717261e+01]]
```

```
[[  1.21292460e+01   3.59715993e+00   6.85472037e+01   4.64727416e-02]
 [  4.64600849e+01   8.51033801e+01   5.60727565e+01   4.42514207e+01]
 [ -4.25246829e+01   4.84248517e+01   8.36574880e+01   2.46425181e+01]
 [ -1.00000000e+00   9.90485066e+01   7.22401453e+01   6.32551431e+01]]

[[  8.52266185e+00   1.42327500e+01   8.67142599e+01   9.07908541e+00]
 [  5.23213686e+01   7.93040687e+01   9.55689187e+01   5.69686765e+01]
 [  6.61470410e+01   7.60122809e+01   9.91638477e+01   9.30536751e+01]
 [  0.00000000e+00   0.00000000e+00   0.00000000e+00   0.00000000e+00]]]
```

Reward per episode



Effect of Epsilon on Rate of convergence and optimal policy

Both have very similar rate of convergence and very similar optimal policy.