CS6700: Home Work 4

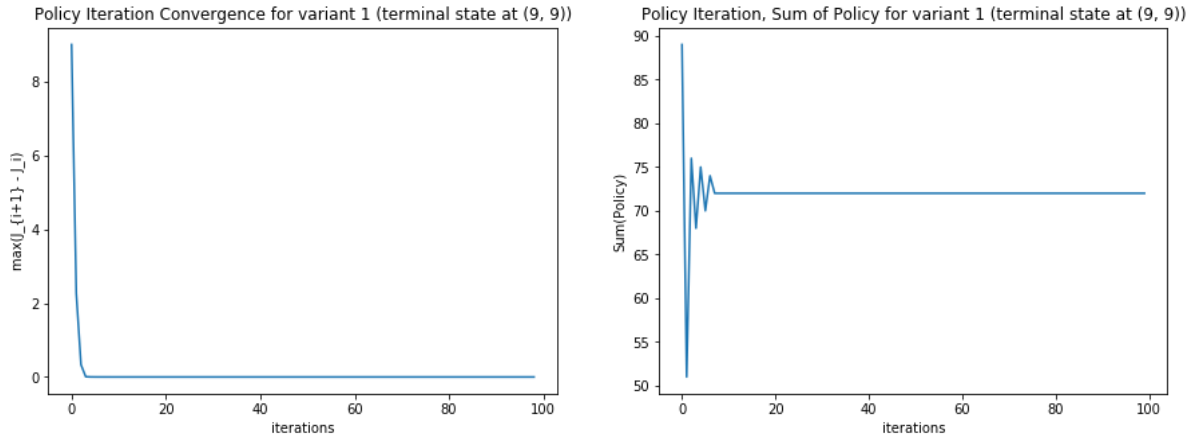
Avinash G. Kori | ED15B006

Question 1

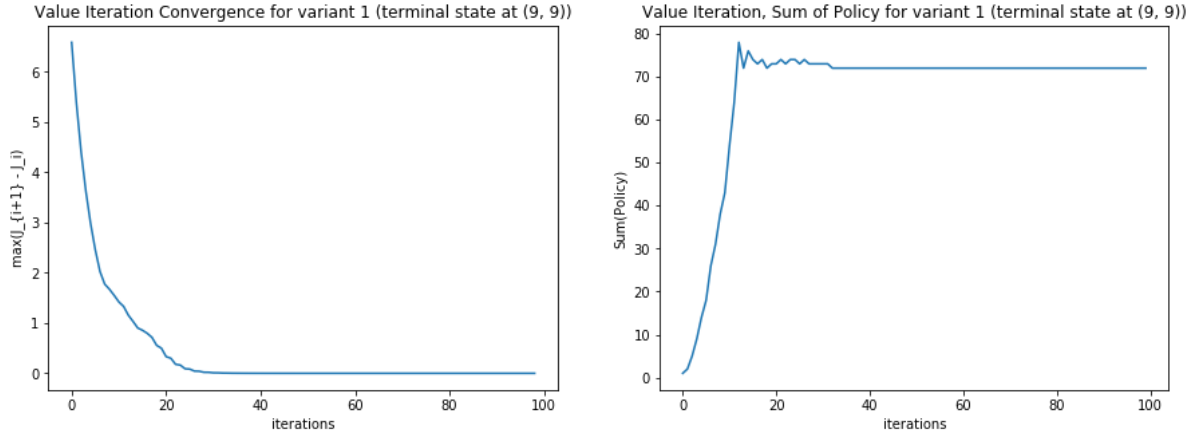
Q1. 1> Value and policy iteration Implementation, in ipython notebook

2.a) Plot graph of maxs |Ji+1(s) − Ji(s)| vs iterations and Ps πi+1(s) 6= πi(s) vs iterations for both value iteration and policy iteration.

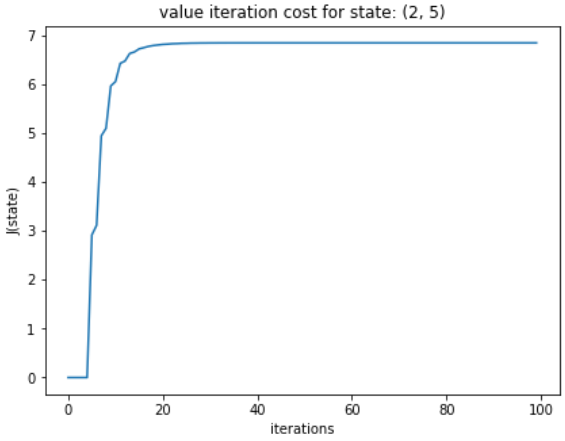
```
================== Value Iteration Plots ===============
```

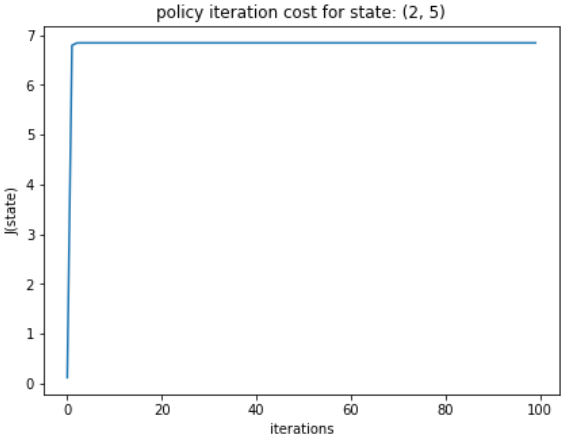


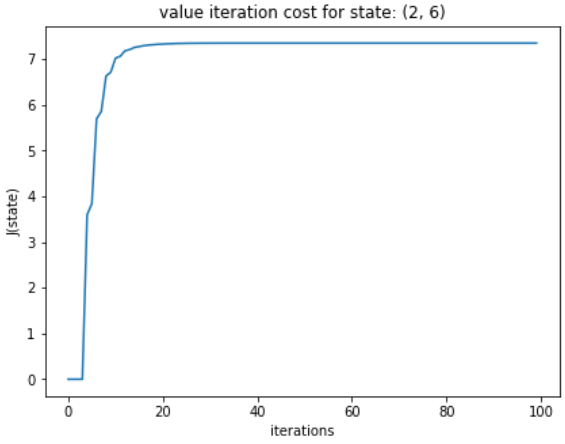

```
================== Policy Iteration Plots ===============
```

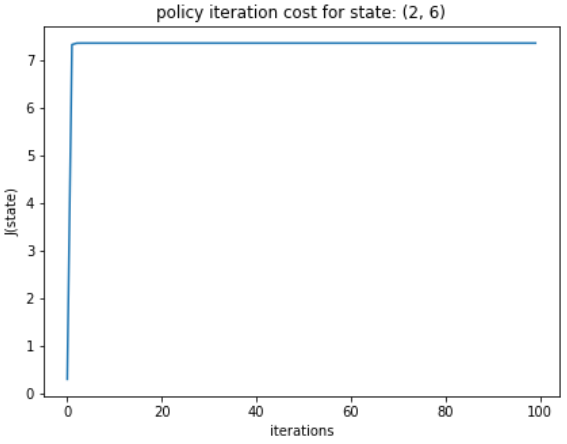


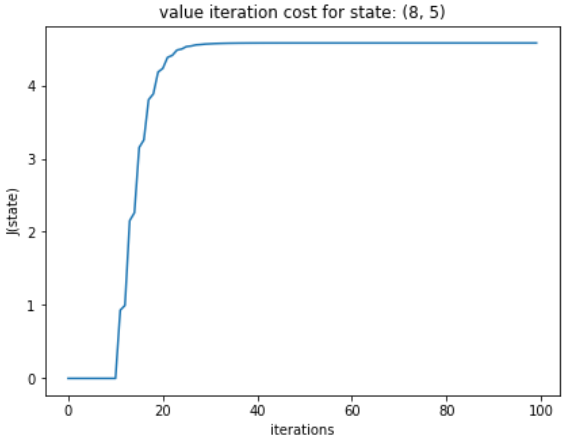
2.b) Compare value iteration and policy iteration by plotting J(s) vs iterations for three random states. Which converges faster? Why?

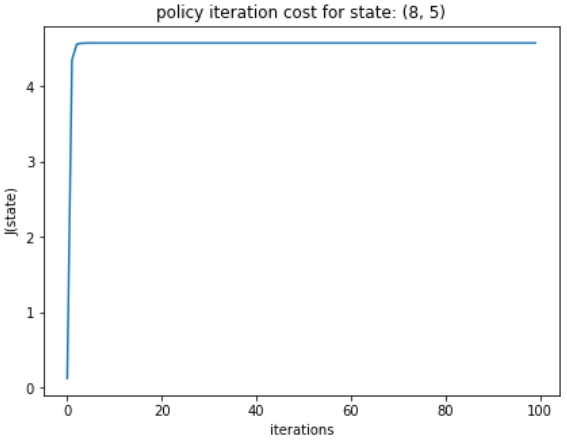










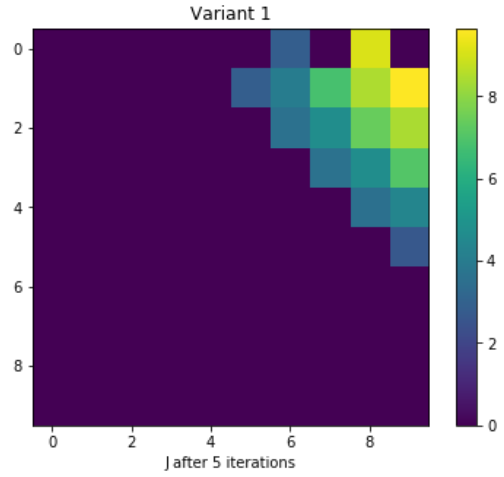


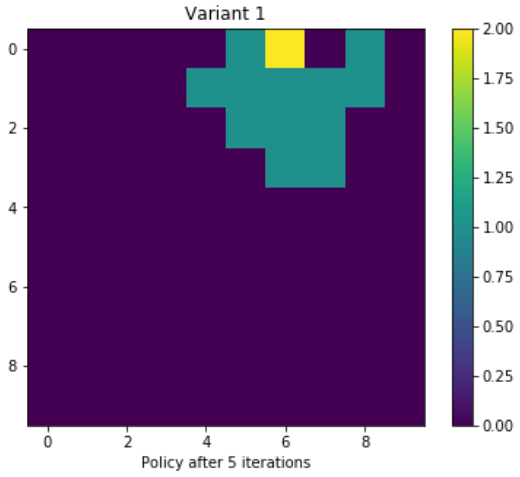
Convergence of policy iteration is faster: Value iteration takes about 20-30 iterations to converge, but policy iteration converges with in 5 iterations

- In case of policy iteration each policy updated policy should be better than it's previous policy
- In case of policy iteration closed loop simultanious equations are solved to find optimal cost (policy evaluation step), but in case of value iteration simultanious equations are solved in iterative fashion

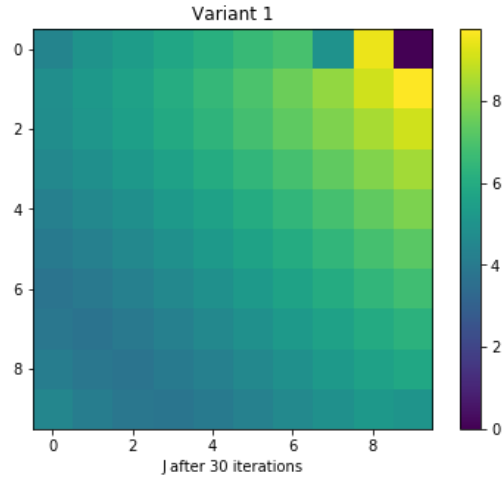
2.c) Show J(s) and greedy policy π(s), ∀s, obtained after 5 iterations, and after you stop value iteration. value iteration stops after 30 iterations...

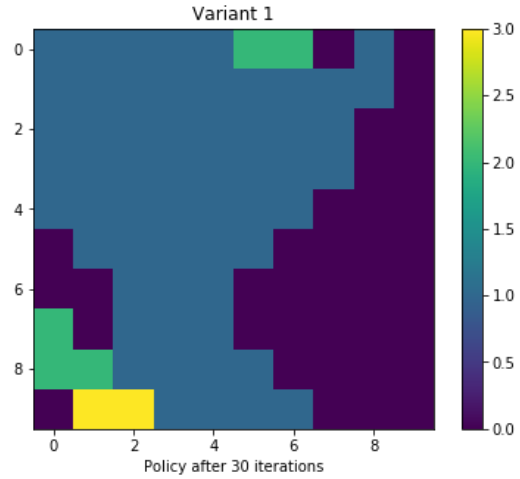
```
========== ValueIteration (N = 5) =============
```






```
--------------------------------
[['^' '^' '^' '^' '^' '>' 'v' 'W' '>' 'T']
 ['^' '^' '^' '^' '>' '>' '>' '>' '>' '^']
 ['^' '^' '^' '^' '^' '>' '>' '>' '^' '^']
 ['^' '^' '^' '^' '^' '^' '>' '>' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['W' '^' '^' '^' '^' '^' '^' '^' '^' '^']]
--------------------------------
==== ValueIteration(N = 30) after convergence of value iteration =====
```

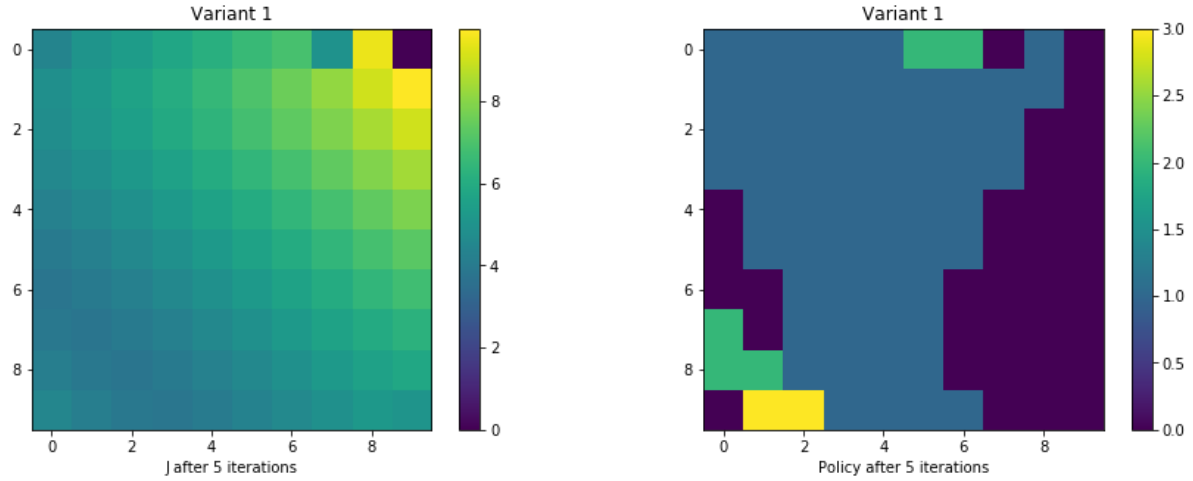




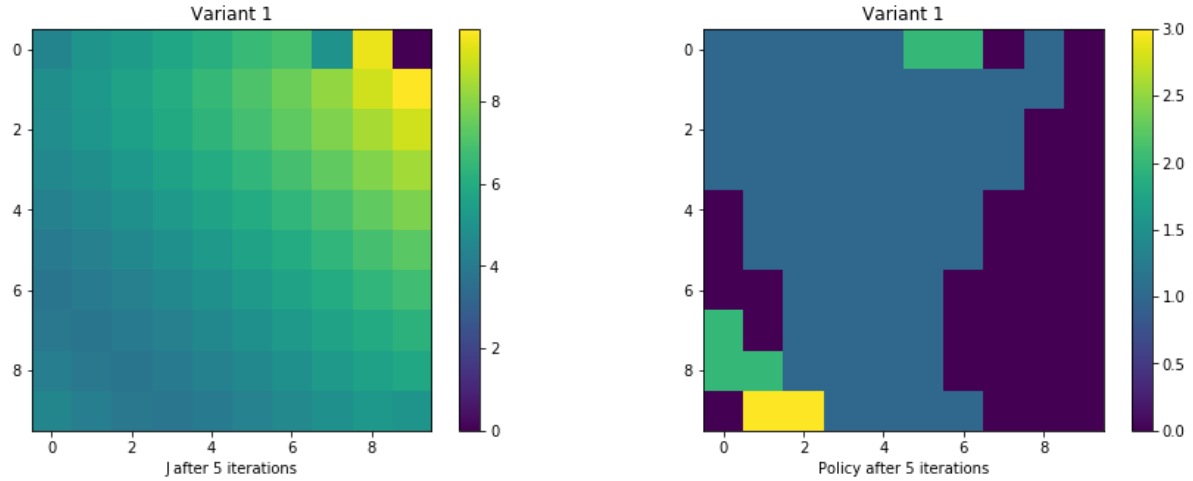

```
--------------------------------
[['>' '>' '>' '>' '>' 'v' 'v' 'W' '>' 'T']
 ['>' '>' '>' '>' '>' '>' '>' '>' '>' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['>' '>' '>' '>' '>' '>' '>' '^' '^' '^']
 ['^' '>' '>' '>' '>' '>' '^' '^' '^' '^']
 ['^' '^' '>' '>' '>' '^' '^' '^' '^' '^']
 ['v' '^' '>' '>' '>' '^' '^' '^' '^' '^']
 ['v' 'v' '>' '>' '>' '>' '^' '^' '^' '^']
 ['W' '<' '<' '>' '>' '>' '>' '^' '^' '^']]
--------------------------------
```

2.d) Show J(s) and greedy policy π(s), ∀s, obtained after 5 iterations, and after you stop policy iteration. policy iteration stops after 5 iterations

```
========== PolicyIteration(N = 5) ===============
```




```
---------------------------------
[['>' '>' '>' '>' '>' 'v' 'v' 'W' '>' 'T']
 ['>' '>' '>' '>' '>' '>' '>' '>' '>' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['^' '>' '>' '>' '>' '>' '>' '^' '^' '^']
 ['^' '>' '>' '>' '>' '>' '>' '^' '^' '^']
 ['^' '^' '>' '>' '>' '>' '^' '^' '^' '^']
 ['v' '^' '>' '>' '>' '>' '^' '^' '^' '^']
 ['v' 'v' '>' '>' '>' '>' '^' '^' '^' '^']
 ['W' '<' '<' '>' '>' '>' '>' '^' '^' '^']]
---------------------------------
=== PolicyIteration(N = 5) after convergence of policy iteration =====
```



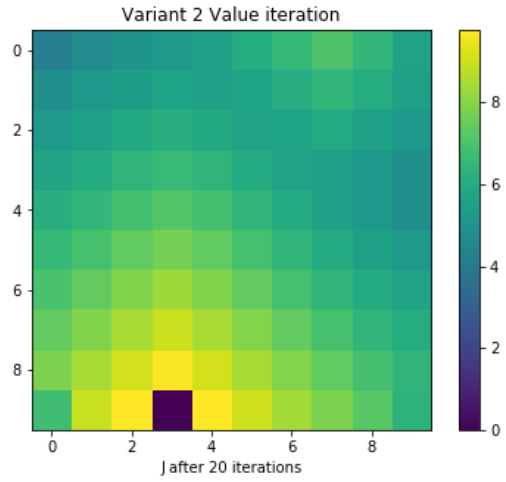

```
---------------------------------
[['>' '>' '>' '>' '>' 'v' 'v' 'W' '>' 'T']
 ['>' '>' '>' '>' '>' '>' '>' '>' '>' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['^' '>' '>' '>' '>' '>' '>' '^' '^' '^']
 ['^' '>' '>' '>' '>' '>' '>' '^' '^' '^']
 ['^' '^' '>' '>' '>' '>' '^' '^' '^' '^']
 ['v' '^' '>' '>' '>' '>' '^' '^' '^' '^']
 ['v' 'v' '>' '>' '>' '>' '^' '^' '^' '^']
 ['W' '<' '<' '>' '>' '>' '>' '^' '^' '^']]
---------------------------------
```

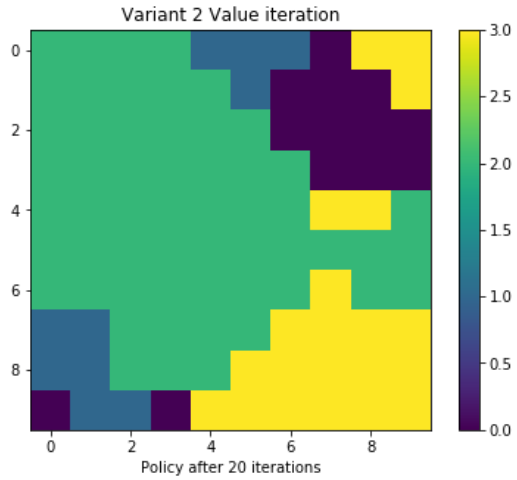
2.e) Explain the behaviour of J and greedy policy π obtained by value iteration and policy iteration.

- Policy in all the states lead to terminal state, it can even be observed in the policies around wormhole, around wormhole (7, 9) any policy is not directed towards (7, 9), while around wormhole (0, 0) all the policies are leading towards (0,0)
- Cost around terminal state high as compared to any other states
- Cost for all the states remain constant after convergence, Convergence in case of value iteration needs about 25 iteration while policy iteration just needs 5 iterations

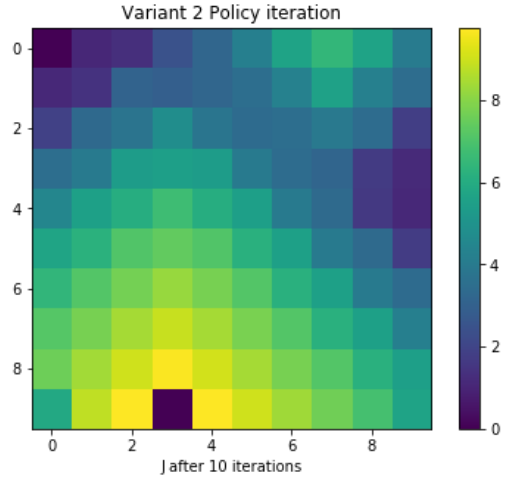
3.a) Show J(s) and policy π(s), ∀s, obtained after you stop value iteration and policy iteration and explain it's behaviour.

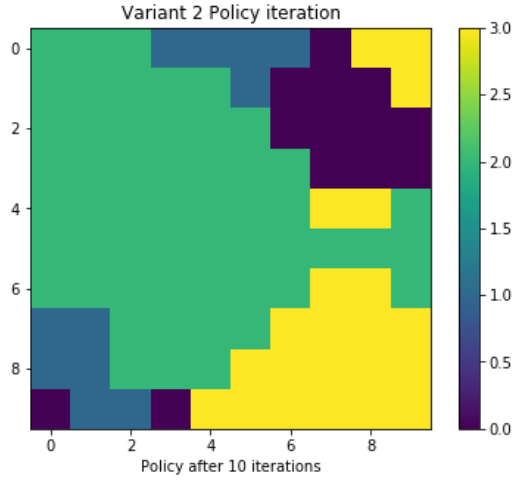
```
======== value iteration stops at 20 iterations ==============
```






```
--------------------------------
[['v' 'v' 'v' 'v' '>' '>' '>' 'W' '<' '<']
 ['v' 'v' 'v' 'v' 'v' '>' '^' '^' '^' '<']
 ['v' 'v' 'v' 'v' 'v' 'v' '^' '^' '^' '^']
 ['v' 'v' 'v' 'v' 'v' 'v' 'v' '^' '^' '^']
 ['v' 'v' 'v' 'v' 'v' 'v' 'v' '<' '<' 'v']
 ['v' 'v' 'v' 'v' 'v' 'v' 'v' 'v' 'v' 'v']
 ['v' 'v' 'v' 'v' 'v' 'v' 'v' '<' 'v' 'v']
 ['>' '>' 'v' 'v' 'v' 'v' '<' '<' '<' '<']
 ['>' '>' 'v' 'v' 'v' '<' '<' '<' '<' '<']
 ['W' '>' '>' 'T' '<' '<' '<' '<' '<' '<']]
--------------------------------
======= policy iteration stops at 10 iterations ==============
```






```
--------------------------------
[['v' 'v' 'v' '>' '>' '>' '>' 'W' '<' '<']
 ['v' 'v' 'v' 'v' 'v' '>' '^' '^' '^' '<']
 ['v' 'v' 'v' 'v' 'v' 'v' '^' '^' '^' '^']
 ['v' 'v' 'v' 'v' 'v' 'v' 'v' '^' '^' '^']
 ['v' 'v' 'v' 'v' 'v' 'v' 'v' '<' '<' 'v']
 ['v' 'v' 'v' 'v' 'v' 'v' 'v' 'v' 'v' 'v']
 ['v' 'v' 'v' 'v' 'v' 'v' 'v' '<' '<' 'v']
 ['>' '>' 'v' 'v' 'v' 'v' '<' '<' '<' '<']
 ['>' '>' 'v' 'v' 'v' '<' '<' '<' '<' '<']
 ['W' '>' '>' 'T' '<' '<' '<' '<' '<' '<']]
--------------------------------
```

- Policy in all the states lead to terminal state, it can even be observed in the policies around wormhole, around wormhole (7, 9) as all the policies are directing towards (7, 9), while around wormhole (0, 0) no policies are directing, as (3, 0) is terminal state
- Cost around terminal state and (7,9) wormhole is high as compared to any other states
- Cost for all the states remain constant after convergence, Convergence in case of value iteration needs about 25 iteration while policy iteration just needs 5 iterations

---

Question 2

1) Find an optimal policy using policy iteration starting with a policy that will always cruise independent of the town. Solve it for discount factors β ranging from 0 to 0.95 with intervals of 0.05. Tabulate the optimal policies and optimal values obtained for different values of β. (5marks)

```
==================== Policy Iteration ========================
('alpha        ', '       cost          ', '       policy')
0.0     [16.  15.   4.5]     [2 2 3]
0.05     [16.5440639  15.75739147  5.2185664 ]     [2 2 3]
0.1     [17.15799375 16.59690146  6.00307659]     [2 2 3]
0.15     [17.85346742 17.53220299  6.86550504]     [2 2 3]
0.2     [18.64454899 18.57953341  7.82021028]     [2 2 3]
0.25     [19.54816605 19.75811775  8.88440947]     [2 2 3]
0.3     [20.58477005 21.0907643  10.07883526]     [2 2 3]
0.35     [21.7792532  22.60470363 11.42864865]     [2 2 3]
0.4     [23.16221888 24.33276447 12.96470426]     [2 2 3]
0.45     [24.77172862 26.31500611 14.72529119]     [2 2 3]
0.5     [26.65567963 28.60095664 16.75850298]     [2 2 3]
0.55     [28.87500064 31.25263978 19.12542434]     [2 2 3]
0.6     [31.50789165 34.34860957 21.90435971]     [2 2 3]
0.65     [34.65537473 37.98925346 25.19637055]     [2 2 3]
0.7     [38.44846791 42.30366766 29.13243307]     [2 2 3]
0.75     [43.05734347 47.45845719 33.88257685]     [2 2 3]
0.8     [48.70288385 53.66886382 39.66741754]     [2 2 2]
0.85     [55.67110533 61.21268058 46.77255266]     [2 2 2]
0.9     [64.33097876 70.44746992 55.56634954]     [2 2 2]
```

2.a) Find an optimal policy using modified policy iteration. Let mk = 5 ∀k. Start with a policy that will always cruise independent of the town. Let β = 0.9. What are the optimal values? (3 marks)

```
============ Modified Policy Iteration with M = 5 ==================
0.9     [64.33097876 70.44746992 55.56634954]     [2 2 2]
```

2.b) Do you find any improvement if you choose mk = 10 ∀k? Explain. (2 marks)

```
============ Modified Policy Iteration with M = 10 ==================
0.9     [ 91.91066062 101.63094332  83.3489995 ]     [2 2 2]
```

3.) Find an optimal policy using value iteration and Gauss-Seidel value iteration starting with a zero vector. Let β = 0.9. What are the optimal values? (5 marks)

```
============ Value Iteration ============
[130.79243435 138.11318907 124.30186832] [2 2 2]
============ Gauss SeidelValue Iteration ============
[130.51816094 137.81893625 124.03704263] [2 2 2]
```

Reference

- Prashanth L. A. CS6700: Reinforcement learning Course notes, 2018
- Dimitri P. Bertsekas. Dynamic Programming and Optimal Control, vol. I. Athena Scientific, 2017.