CS6700: Home Work 4

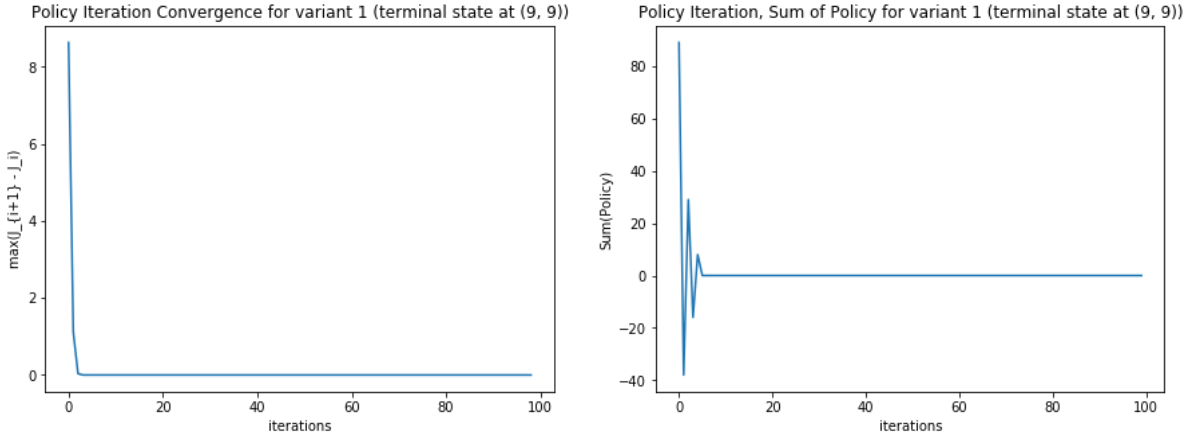
Avinash G. Kori | ED15B006

Question 1

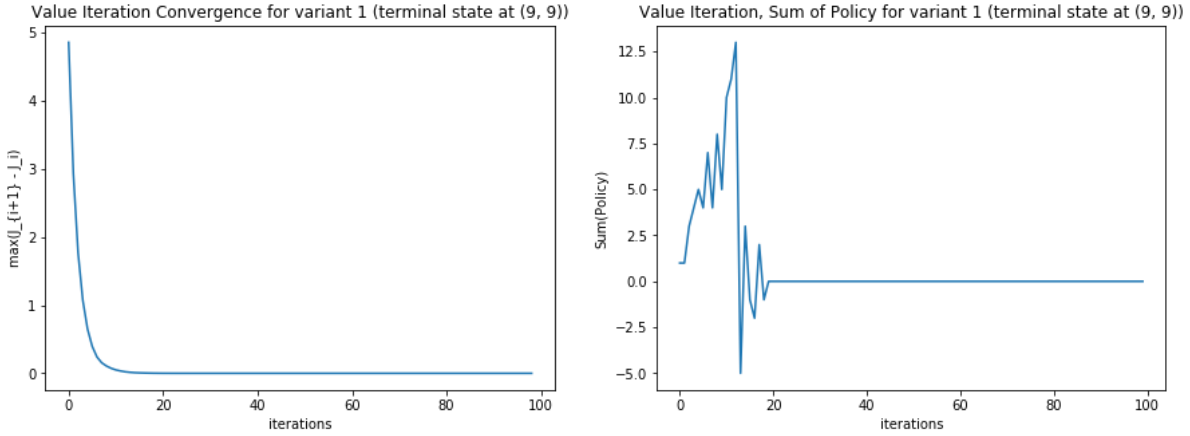
Q1. 1> Value and policy iteration Implementation, in ipython notebook

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.

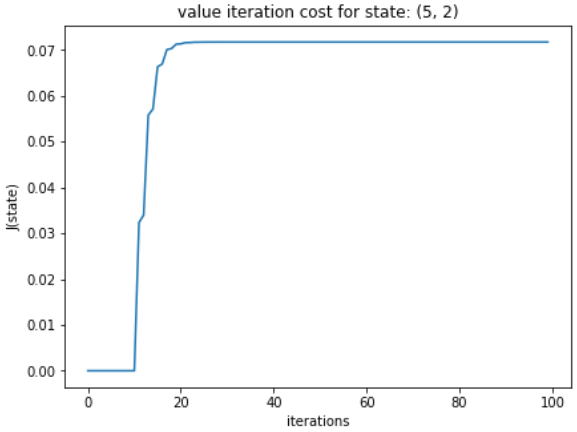
```
================== Value Iteration Plots ===============
```

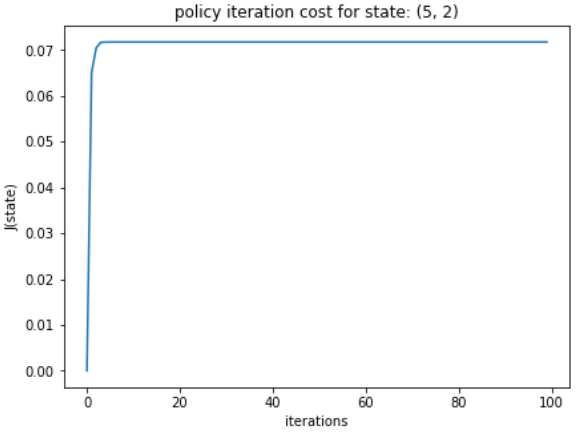


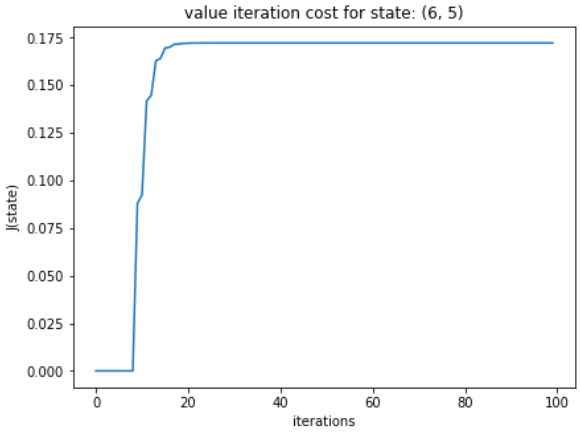

```
================== Policy Iteration Plots ===============
```

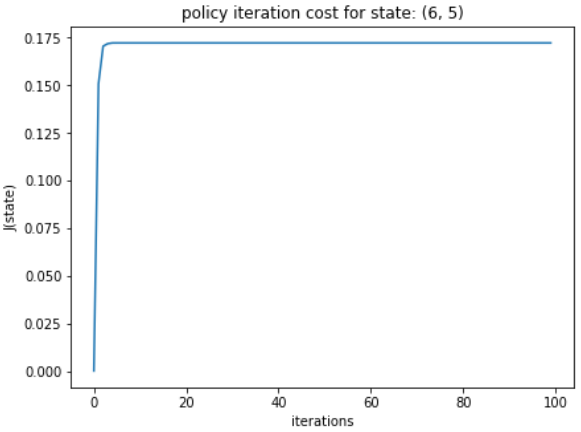


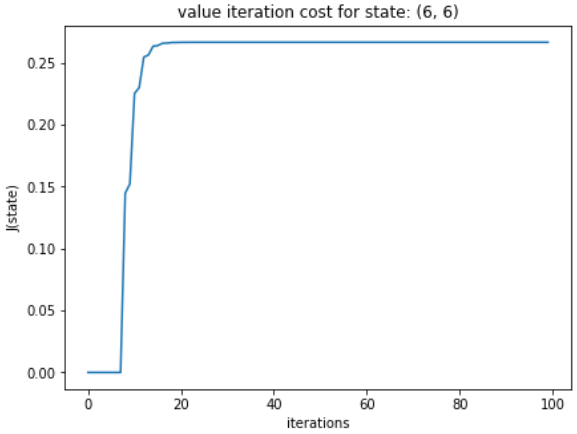
2.b) Compare value iteration and policy iteration by plotting J(s) vs iterations for three random states. Which converges faster? Why?

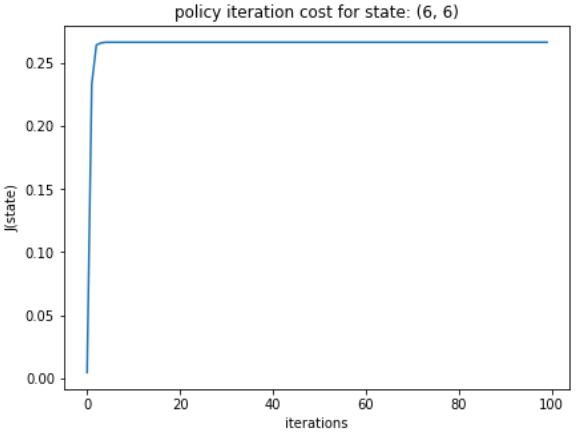












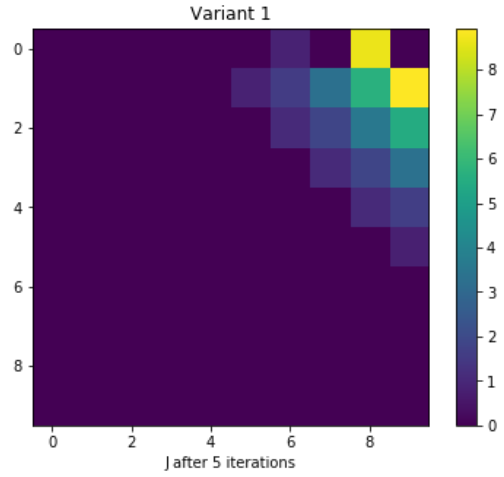
Convergence of policy iteration is faster: Value iteration takes about 15-20 iterations to converge, but policy iteration converges with in 5 iterations

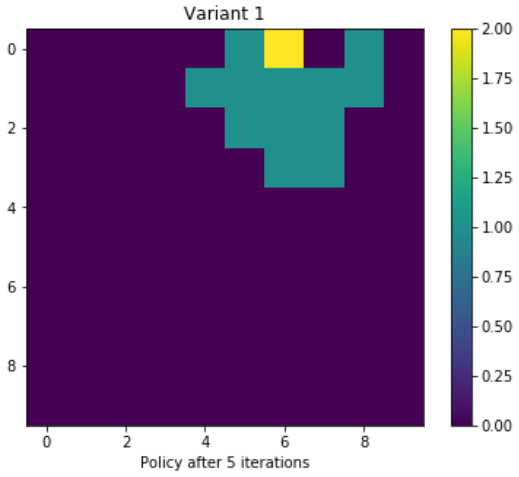
- In case of policy iteration each policy updated policy should be better than it's previous policy
- In case of policy iteration closed loop simultanious equations are solved to find optimal cost (policy evaluation step), but in case of value iteration simultanious equations are solved in iterative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

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

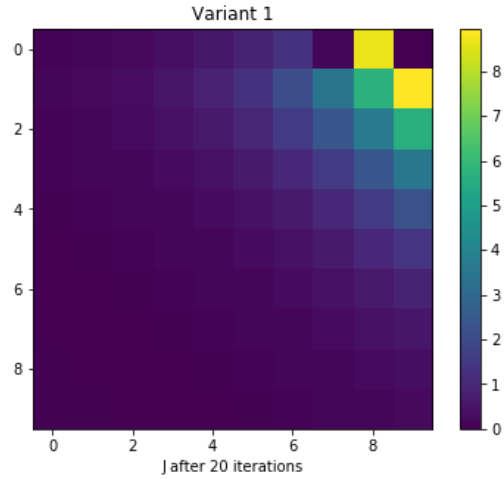
value iteration stops after 20 iterations...

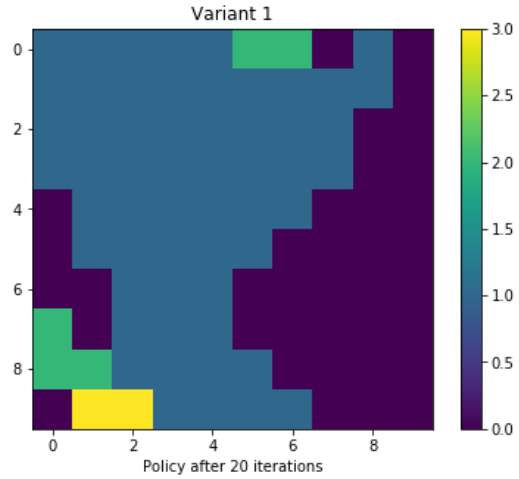
```
========== ValueIteration (N = 5) =============
```






```
--------------------------------
[['^' '^' '^' '^' '^' '>' 'v' 'W' '>' 'T']
 ['^' '^' '^' '^' '^' '>' '>' '>' '>' '^']
 ['^' '^' '^' '^' '^' '^' '>' '>' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '>' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['^' '^' '^' '^' '^' '^' '^' '^' '^' '^']
 ['W' '^' '^' '^' '^' '^' '^' '^' '^' '^']]
--------------------------------
==== ValueIteration(N = 20) after convergence of value iteration =====
```



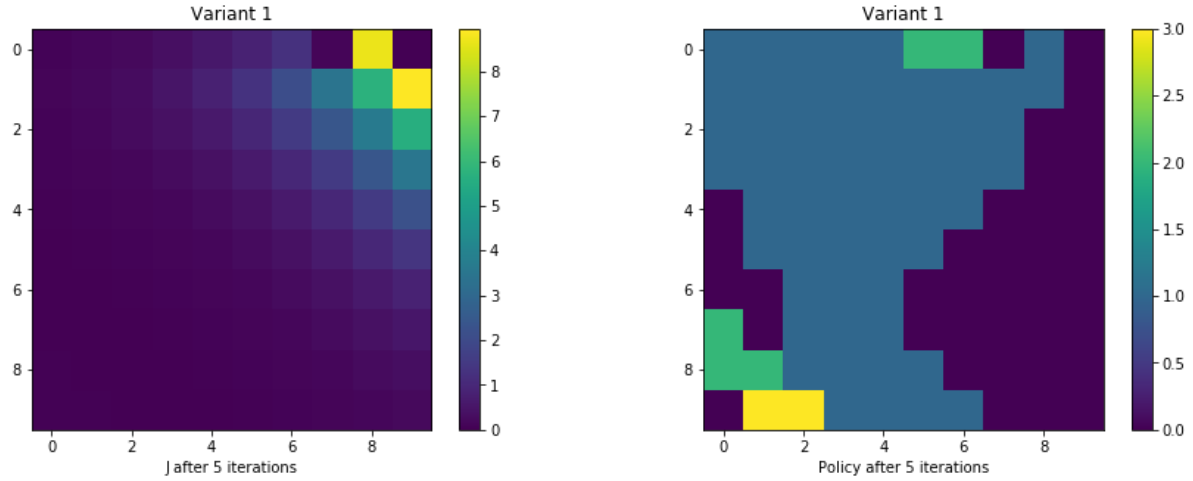



```
--------------------------------
[['>' '>' '>' '>' '>' 'v' 'v' 'W' '>' 'T']
 ['>' '>' '>' '>' '>' '>' '>' '>' '>' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['^' '>' '>' '>' '>' '>' '>' '^' '^' '^']
 ['^' '>' '>' '>' '>' '>' '^' '^' '^' '^']
 ['^'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

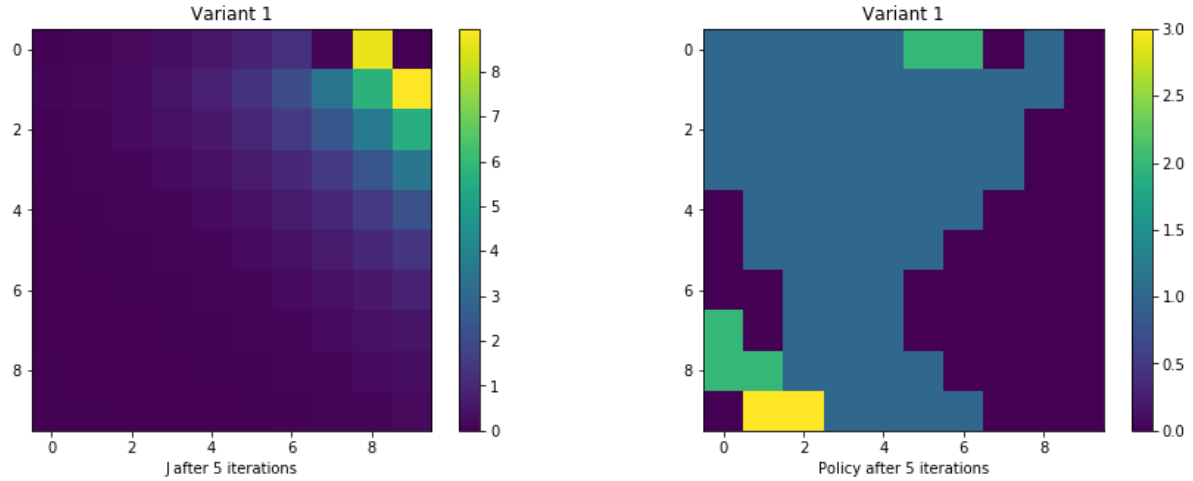
2.d) Show J(s) and greedy policy π(s), ∀s, obtained after 5 iterations, and after you stop policy iteration.

policy iteration stops after 4 iterations

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




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



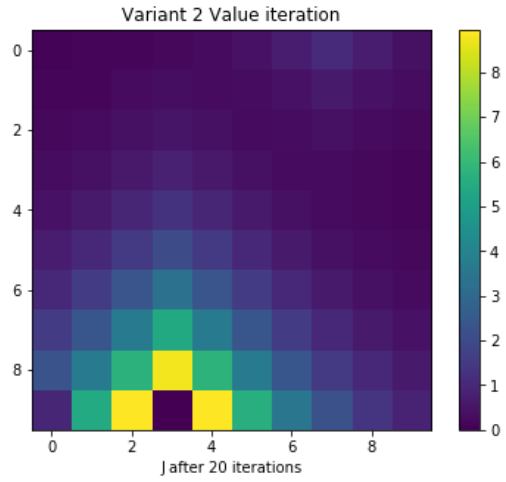

```
--------------------------------
[['>' '>' '>' '>' '>' 'v' 'v' 'W' '>' 'T']
 ['>' '>' '>' '>' '>' '>' '>' '>' '>' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['>' '>' '>' '>' '>' '>' '>' '>' '^' '^']
 ['^' '>' '>' '>' '>' '>' '>' '^' '^' '^']
 ['^' '>' '>' '>' '>' '>' '^' '^' '^' '^']
 ['^'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

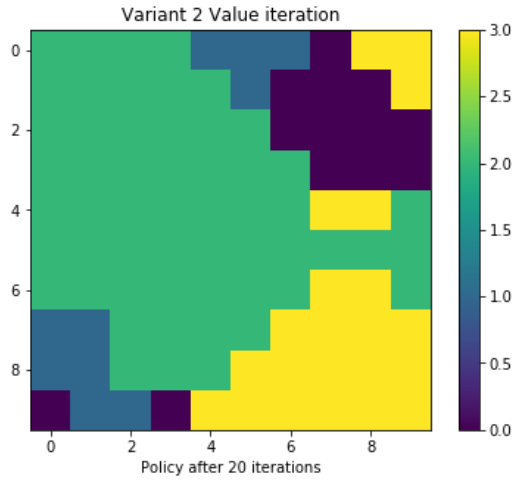
2.e) Explain the behaviour of J and greedy policy π obtained by value iteration and policy iteration.

- 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

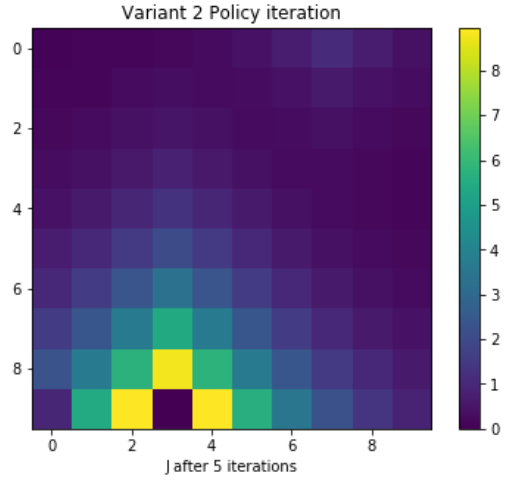
3.a) Show J(s) and policy π(s), ∀s, obtained after you stop value iteration and policy iteration and explain it's behaviour.

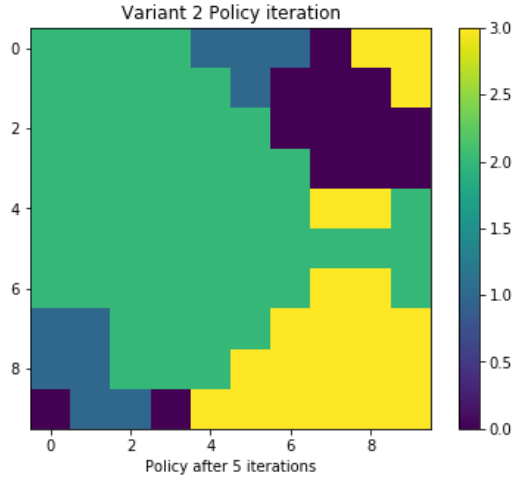
```
======== value iteration stops at 20th iteration ==============
```






```
---------------------------------
[['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' '<' '<' '<' '<' '<']
 ['W' '>' '>' 'T' '<' '<' '<' '<' '<' '<']]
---------------------------------
======= policy iteration stops at 5th iteration ==============
```






```
---------------------------------
[['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' '<' '<' '<' '<' '<']
 ['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     [ 8. 16.  7.]     [1 1 1]
0.05     [ 8.51152729 16.40025991  7.49886907]     [1 1 1]
0.1     [ 9.07650615 16.85636856  8.05086512]     [1 1 1]
0.15     [ 9.70812149 17.46450304  8.66916045]     [1 2 1]
0.2     [10.43703008 18.48214286  9.3843985 ]     [1 2 1]
0.25     [11.27407407 19.62962963 10.20740741]     [1 2 1]
0.3     [12.24383724 20.93406593 11.16275616]     [1 2 1]
0.35     [13.37871444 22.43076923 12.28282403]     [1 2 1]
0.4     [14.72222222 24.16666667 13.61111111]     [1 2 1]
0.45     [16.33413127 26.2055336  15.20737071]     [1 2 1]
0.5     [18.2987013  28.63636364 17.15584416]     [1 2 1]
0.55     [20.78998864 31.6073967  19.83072522]     [1 2 2]
0.6     [24.02568645 35.32772365 23.45881311]     [1 2 2]
0.65     [28.27669207 40.09628059 28.1299788 ]     [1 2 2]
0.7     [34.06193078 46.43541617 34.36604102]     [1 2 2]
0.75     [42.3174114  55.28505393 43.10631741]     [1 2 2]
0.8     [55.07936507 68.55820105 56.26984126]     [2 2 2]
0.85     [77.24650453 90.81169305 78.43344815]     [2 2 2]
0.9     [121.64997255 135.30277695 122.8334045 ]     [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     [  9.80458517 121.26398216  10.59387195]     [2 2 2]
```

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

- Both the policies are same, but cost needs to be converged
- Modified policy iteration freezes the policies in finite number of steps, in this case as number of actions are just 3

```
============ Modified Policy Iteration with M = 10 ==================
Cost and Policy for state [A, B, C] are: [  9.80458517 121.26398216  10.59387195] and [2
2 2] respectively.
```

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 ============
Cost and Policy for state [A, B, C] are: [121.65345309 135.30625749 122.83688504] and [2
2 2] respectively.
============ Gauss SeidelValue Iteration ============
Cost and Policy for state [A, B, C] are: [121.65347112 135.30627552 122.83690308] and [2
2 2] respectively.
```

Reference

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