Assignment 9, Question 4, Amogha Sekhar, A53301791, CSE 250A

In [1]:
```
import numpy as np
from collections import defaultdict
```

In [2]:
```
def read_file(filename):
    array_prob = np.zeros((81, 81), dtype = np.float)
    with open(filename, 'r') as f:
        for line in f:
            s, s_, prob = line.strip().split() # s_ is s'
            s, s_, prob = int(s)-1, int(s_)-1, float(prob)
            array_prob[s, s_] = prob
    return array_prob
```

In [3]:
```
prob_a1 = read_file("prob_a1.txt")
prob_a2 = read_file("prob_a2.txt")
prob_a3 = read_file("prob_a3.txt")
prob_a4 = read_file("prob_a4.txt")
```

In [4]:
```
reward = np.zeros((81, 1), dtype = np.float)
with open('rewards.txt', 'r') as f:
    i = 0
    for line in f:
        reward[i] = float(line.strip())
        i += 1
```

In [5]:
```
gamma = 0.9925
```

Policy Iteration

In [6]:
```
policy = np.zeros((81, 1), dtype = np.int) # The policy initialized at ranc

v_pi = np.ones((81, 1), dtype = np.float) # Value function
v_pi_before = np.zeros((81, 1), dtype = np.float) # Value function of prior

iter_ = 0

while max(abs(v_pi - v_pi_before)) > 0.0001:

    p_pi = np.zeros((81, 81), dtype = np.float)

    for s in range(81):
        for s_ in range(81):
            if policy[s] == 0:
                p_pi[s, s_] = prob_a1[s, s_]
            if policy[s] == 1:
                p_pi[s, s_] = prob_a2[s, s_]
            if policy[s] == 2:
                p_pi[s, s_] = prob_a3[s, s_]
            if policy[s] == 3:
                p_pi[s, s_] = prob_a4[s, s_]

    v_pi_before = np.copy(v_pi)
    v_pi = np.linalg.inv(np.identity(81) - gamma * p_pi).dot(reward)

    # argmax part
    for s in range(81):

        sum_a1, sum_a2, sum_a3, sum_a4 = 0, 0, 0, 0

        for s_ in range(81):
            sum_a1 += prob_a1[s, s_] * v_pi[s_]
            sum_a2 += prob_a2[s, s_] * v_pi[s_]
            sum_a3 += prob_a3[s, s_] * v_pi[s_]
            sum_a4 += prob_a4[s, s_] * v_pi[s_]

        list_sum = [sum_a1, sum_a2, sum_a3, sum_a4]
        best_policy = list_sum.index(max(list_sum))

        policy[s] = best_policy

        iter_ += 1

print("Took", iter_, "iterations to converge")
```

```
Took 486 iterations to converge
```

In [7]:
```
v_pi
```

Out[7]:
```
array([[   0.        ],
       [   0.        ],
       [ 100.70098073],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [ 102.3752644 ],
       [ 101.52364515],
       [   0.        ],
       [   0.        ],
       [ 109.48993454],
       [ 110.40903296],
       [ 111.33584663],
       [   0.        ],
       [   0.        ],
       [ 103.23462342],
       [   0.        ],
       [ 106.77826755],
       [ 107.67462643],
       [ 108.57848712],
       [   0.        ],
       [ 112.27044032],
       [   0.        ],
       [   0.        ],
       [ 104.10121204],
       [ 104.97507555],
       [ 105.88853591],
       [   0.        ],
       [   0.        ],
       [ 114.1632295 ],
       [ 113.21287932],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [ 103.78140737],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [ 115.12155727],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [-133.33333333],
       [  90.9853796 ],
       [-133.33333333],
       [   0.        ],
       [-133.33333333],
       [ 116.08792959],
       [ 122.02491241],
       [   0.        ],
       [   0.        ],
       [  81.39949278],
       [  93.67165583],
       [  95.17285726],
       [ 108.34261934],
       [ 109.58365072],
       [ 123.64307021],
       [ 123.1822391 ],
       [   0.        ],
       [   0.        ],
       [-133.33333333],
       [  81.39949278],
       [-133.33333333],
       [   0.        ],
       [-133.33333333],
       [ 125.24978944],
       [ 124.20738563],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [   0.        ],
       [ 133.33333333],
       [   0.        ],
       [   0.        ]])
```

In [9]:
```
policy
```

Out[9]:
```
array([[0],
       [0],
       [2],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [2],
       [1],
       [0],
       [0],
       [3],
       [3],
       [2],
       [0],
       [0],
       [2],
       [0],
       [3],
       [3],
       [0],
       [0],
       [2],
       [0],
       [0],
       [3],
       [3],
       [0],
       [0],
       [0],
       [2],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [2],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [2],
       [2],
       [0],
       [0],
       [3],
       [3],
       [3],
       [3],
       [3],
       [2],
       [2],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [2],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0]])
```

Value Iteration

In [10]:
```
v_pi = np.ones((81, 1), dtype = np.float) # Value function
v_pi_before = np.zeros((81, 1), dtype = np.float) # Value function of prior

iter_ = 0

while max(abs(v_pi - v_pi_before)) > 0.000001:

    for s in range(81):

        sum_a1, sum_a2, sum_a3, sum_a4 = 0, 0, 0, 0

        for s_ in range(81):
            sum_a1 += prob_a1[s, s_] * v_pi_before[s_]
            sum_a2 += prob_a2[s, s_] * v_pi_before[s_]
            sum_a3 += prob_a3[s, s_] * v_pi_before[s_]
            sum_a4 += prob_a4[s, s_] * v_pi_before[s_]

        v_pi_before[s] = v_pi[s]

        list_sum = [sum_a1, sum_a2, sum_a3, sum_a4]
        v_pi[s] = reward[s] + gamma * max(list_sum)

    iter_ += 1

print("Took", iter_, "iterations to converge")
```

```
Took 3672 iterations to converge
```

In [11]:
```
v_pi
```

Out[11]:
```
array([[ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.00700865e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.02375148e+02],
       [ 1.01523530e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.09489817e+02],
       [ 1.10408916e+02],
       [ 1.11335729e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.03234507e+02],
       [ 9.93633760e-07],
       [ 1.06778151e+02],
       [ 1.07674510e+02],
       [ 1.08578371e+02],
       [ 9.93633760e-07],
       [ 1.12270323e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.04101096e+02],
       [ 1.04974959e+02],
       [ 1.05888420e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.14163112e+02],
       [ 1.13212762e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.03781292e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.15121440e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [-1.33333200e+02],
       [ 9.09852775e+01],
       [-1.33333200e+02],
       [ 9.93633760e-07],
       [-1.33333200e+02],
       [ 1.16087812e+02],
       [ 1.22024788e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 8.13994071e+01],
       [ 9.36715583e+01],
       [ 9.51727592e+01],
       [ 1.08342509e+02],
       [ 1.09583540e+02],
       [ 1.23642946e+02],
       [ 1.23182115e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [-1.33333200e+02],
       [ 8.13994072e+01],
       [-1.33333200e+02],
       [ 9.93633760e-07],
       [-1.33333200e+02],
       [ 1.25249665e+02],
       [ 1.24207261e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 9.93633760e-07],
       [ 1.33333202e+02],
       [ 9.93633760e-07],
       [ 9.93633760e-07]])
```

In [12]:
```
policy = np.zeros((81, 1), dtype = np.int) # The policy initialized at ranc

for s in range(81):
    sum_a1, sum_a2, sum_a3, sum_a4 = 0, 0, 0, 0

    for s_ in range(81):
        sum_a1 += prob_a1[s, s_] * v_pi[s_]
        sum_a2 += prob_a2[s, s_] * v_pi[s_]
        sum_a3 += prob_a3[s, s_] * v_pi[s_]
        sum_a4 += prob_a4[s, s_] * v_pi[s_]

    list_sum = [sum_a1, sum_a2, sum_a3, sum_a4]
    best_policy = list_sum.index(max(list_sum))

    policy[s] = best_policy
```

In [13]:
```
policy
```

Out[13]:
```
array([[0],
       [0],
       [2],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [2],
       [1],
       [0],
       [0],
       [3],
       [3],
       [2],
       [0],
       [0],
       [2],
       [0],
       [3],
       [3],
       [0],
       [0],
       [2],
       [0],
       [0],
       [3],
       [3],
       [0],
       [0],
       [0],
       [2],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [2],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [2],
       [2],
       [0],
       [0],
       [3],
       [3],
       [3],
       [3],
       [3],
       [2],
       [2],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [2],
       [1],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0]])
```

```
In [ ]:
```