# Tarea: Aprendizaje por Refuerzo

1. $f_R(s, a, s_f) = f_R(sf)$, $\gamma = 0.9$:

$$f_{M_T} = \begin{matrix} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} \begin{matrix} a1 & a2 \\ \end{matrix} \begin{bmatrix} s_2 & s_2 \\ s_1 & s_3 \\ s_3 & s_1 \\ s_1 & s_4 \end{bmatrix} \quad f_R(s_f) = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{matrix} \begin{bmatrix} 2 \\ 1 \\ -1 \\ 10 \end{bmatrix}$$

In [ ]:
```
import numpy as np

fmt: np.ndarray = np.array([
    np.array([2, 2]),
    np.array([1, 3]),
    np.array([3, 1]),
    np.array([1, 4]),
])
fr = [2, 1, -1, 10]
gamma = .9
```

a. $V(s)$

In [ ]:
```
a_size = len(fr)
vs = np.zeros(a_size)
iteration = 0
while True:
    i = 0
    new_vs = np.zeros(a_size)
    for s in fmt:
        a1, a2 = s
        vsi = [fr[a1-1] + (gamma * vs[a1-1]), fr[a2-1] + (gamma * vs[a2-1])]
        new_vs[i] = max(vsi)
        i += 1
    print(iteration, new_vs)
    diff = new_vs - vs
    if len(diff[diff>.1]) == 0:
        break
    vs = new_vs
    iteration += 1
```

```
0 [ 1.  2.  2. 10.]
1 [ 2.8  2.9  2.9 19. ]
2 [ 3.61  4.52  4.52 27.1 ]
3 [ 5.068  5.249  5.249 34.39 ]
4 [ 5.7241  6.5612  6.5612 40.951 ]
5 [ 6.90508  7.15169  7.15169 46.8559 ]
6 [ 7.436521  8.214572  8.214572 52.17031 ]
7 [ 8.3931148  8.6928689  8.6928689 56.953279 ]
8 [ 8.82358201  9.55380332  9.55380332 61.2579511 ]
9 [ 9.59842299  9.94122381  9.94122381 65.13215599]
10 [ 9.94710143 10.63858069 10.63858069 68.61894039]
11 [10.57472262 10.95239129 10.95239129 71.75704635]
```

```
12 [10.85715216 11.51725036 11.51725036 74.58134172]
13 [11.36552532 11.77143694 11.77143694 77.12320755]
14 [11.59429325 12.22897279 12.22897279 79.41088679]
15 [12.00607551 12.43486392 12.43486392 81.46979811]
16 [12.19137753 12.80546796 12.80546796 83.3228183 ]
17 [12.52492116 12.97223978 12.97223978 84.99053647]
18 [12.6750158  13.27242905 13.27242905 86.49148282]
19 [12.94518614 13.40751422 13.40751422 87.84233454]
20 [13.0667628  13.65066753 13.65066753 89.05810109]
21 [13.28560078 13.76008652 13.76008652 90.15229098]
22 [13.38407787 13.9570407  13.9570407  91.13706188]
23 [13.56133663 14.04567008 14.04567008 92.02335569]
24 [13.64110307 14.20520297 14.20520297 92.82102012]
25 [13.78468267 14.27699276 14.27699276 93.53891811]
26 [13.84929349 14.4062144  14.4062144  94.1850263 ]
27 [13.96559296 14.46436414 14.46436414 94.76652367]
28 [14.01792773 14.56903367 14.56903367 95.2898713 ]
29 [14.1121303  14.61613495 14.61613495 95.76088417]
30 [14.15452146 14.70091727 14.70091727 96.18479576]
31 [14.23082554 14.73906931 14.73906931 96.56631618]
32 [14.26516238 14.80774299 14.80774299 96.90968456]
33 [14.32696869 14.83864614 14.83864614 97.21871611]
34 [14.35478153 14.89427182 14.89427182 97.4968445 ]
35 [14.40484464 14.91930338 14.91930338 97.74716005]
36 [14.42737304 14.96436017 14.96436017 97.97244404]
37 [14.46792416 14.98463573 14.98463573 98.17519964]
38 [14.48617216 15.02113174 15.02113174 98.35767967]
39 [14.51901857 15.03755494 15.03755494 98.52191171]
40 [14.53379945 15.06711671 15.06711671 98.66972054]
41 [14.56040504 15.08041951 15.08041951 98.80274848]
42 [14.57237755 15.10436454 15.10436454 98.92247363]
43 [14.59392808 15.1151398  15.1151398  99.03022627]
44 [14.60362582 15.13453527 15.13453527 99.12720364]
```

b. $Q(s, a)$

In [ ]:

```python
import numpy as np

m, n = fmt.shape
qsa = np.zeros((m, n))
iteration = 0
while True:
    new_qsa = []
    for i in range(m):
        idx1, idx2 = fmt[i]
        r1 = fr[idx1-1] + (gamma * max(qsa[idx1-1]))
        r2 = fr[idx2-1] + (gamma * max(qsa[idx2-1]))
        new_qsa.append(np.array([r1, r2]))
    new_qsa_n = np.array(new_qsa)
    diff = new_qsa_n - qsa
    flags = []
    for r in diff:
        flag = True
        if len(r[r>.1]) == 0:
            flag = False
        flags.append(flag)
    print(iteration, new_qsa_n.T)
    if sum(flags) == 0:
        break
    qsa = new_qsa_n
```

```
    iteration += 1
```

```
0 [[ 1.  2. -1.  2.]
 [ 1. -1.  2. 10.]]
1 [[ 2.8  2.9  0.8  2.9]
 [ 2.8  0.8  2.9 19. ]]
2 [[ 3.61  4.52  1.61  4.52]
 [ 3.61  1.61  4.52 27.1 ]]
3 [[ 5.068  5.249  3.068  5.249]
 [ 5.068  3.068  5.249 34.39 ]]
4 [[ 5.7241  6.5612  3.7241  6.5612]
 [ 5.7241  3.7241  6.5612 40.951 ]]
5 [[ 6.90508  7.15169  4.90508  7.15169]
 [ 6.90508  4.90508  7.15169 46.8559 ]]
6 [[ 7.436521  8.214572  5.436521  8.214572]
 [ 7.436521  5.436521  8.214572 52.17031 ]]
7 [[ 8.3931148  8.6928689  6.3931148  8.6928689]
 [ 8.3931148  6.3931148  8.6928689 56.953279 ]]
8 [[ 8.82358201  9.55380332  6.82358201  9.55380332]
 [ 8.82358201  6.82358201  9.55380332 61.2579511 ]]
9 [[ 9.59842299  9.94122381  7.59842299  9.94122381]
 [ 9.59842299  7.59842299  9.94122381 65.13215599]]
10 [[ 9.94710143 10.63858069  7.94710143 10.63858069]
 [ 9.94710143  7.94710143 10.63858069 68.61894039]]
11 [[10.57472262 10.95239129  8.57472262 10.95239129]
 [10.57472262  8.57472262 10.95239129 71.75704635]]
12 [[10.85715216 11.51725036  8.85715216 11.51725036]
 [10.85715216  8.85715216 11.51725036 74.58134172]]
13 [[11.36552532 11.77143694  9.36552532 11.77143694]
 [11.36552532  9.36552532 11.77143694 77.12320755]]
14 [[11.59429325 12.22897279  9.59429325 12.22897279]
 [11.59429325  9.59429325 12.22897279 79.41088679]]
15 [[12.00607551 12.43486392 10.00607551 12.43486392]
 [12.00607551 10.00607551 12.43486392 81.46979811]]
16 [[12.19137753 12.80546796 10.19137753 12.80546796]
 [12.19137753 10.19137753 12.80546796 83.3228183 ]]
17 [[12.52492116 12.97223978 10.52492116 12.97223978]
 [12.52492116 10.52492116 12.97223978 84.99053647]]
18 [[12.6750158  13.27242905 10.6750158  13.27242905]
 [12.6750158  10.6750158  13.27242905 86.49148282]]
19 [[12.94518614 13.40751422 10.94518614 13.40751422]
 [12.94518614 10.94518614 13.40751422 87.84233454]]
20 [[13.0667628  13.65066753 11.0667628  13.65066753]
 [13.0667628  11.0667628  13.65066753 89.05810109]]
21 [[13.28560078 13.76008652 11.28560078 13.76008652]
 [13.28560078 11.28560078 13.76008652 90.15229098]]
22 [[13.38407787 13.9570407  11.38407787 13.9570407 ]
 [13.38407787 11.38407787 13.9570407  91.13706188]]
23 [[13.56133663 14.04567008 11.56133663 14.04567008]
 [13.56133663 11.56133663 14.04567008 92.02335569]]
24 [[13.64110307 14.20520297 11.64110307 14.20520297]
 [13.64110307 11.64110307 14.20520297 92.82102012]]
25 [[13.78468267 14.27699276 11.78468267 14.27699276]
 [13.78468267 11.78468267 14.27699276 93.53891811]]
26 [[13.84929349 14.4062144  11.84929349 14.4062144 ]
 [13.84929349 11.84929349 14.4062144  94.1850263 ]]
27 [[13.96559296 14.46436414 11.96559296 14.46436414]
 [13.96559296 11.96559296 14.46436414 94.76652367]]
28 [[14.01792773 14.56903367 12.01792773 14.56903367]
```

```
 [14.01792773 12.01792773 14.56903367 95.2898713 ]]
29 [[14.1121303  14.61613495 12.1121303  14.61613495]
 [14.1121303  12.1121303  14.61613495 95.76088417]]
30 [[14.15452146 14.70091727 12.15452146 14.70091727]
 [14.15452146 12.15452146 14.70091727 96.18479576]]
31 [[14.23082554 14.73906931 12.23082554 14.73906931]
 [14.23082554 12.23082554 14.73906931 96.56631618]]
32 [[14.26516238 14.80774299 12.26516238 14.80774299]
 [14.26516238 12.26516238 14.80774299 96.90968456]]
33 [[14.32696869 14.83864614 12.32696869 14.83864614]
 [14.32696869 12.32696869 14.83864614 97.21871611]]
34 [[14.35478153 14.89427182 12.35478153 14.89427182]
 [14.35478153 12.35478153 14.89427182 97.4968445 ]]
35 [[14.40484464 14.91930338 12.40484464 14.91930338]
 [14.40484464 12.40484464 14.91930338 97.74716005]]
36 [[14.42737304 14.96436017 12.42737304 14.96436017]
 [14.42737304 12.42737304 14.96436017 97.97244404]]
37 [[14.46792416 14.98463573 12.46792416 14.98463573]
 [14.46792416 12.46792416 14.98463573 98.17519964]]
38 [[14.48617216 15.02113174 12.48617216 15.02113174]
 [14.48617216 12.48617216 15.02113174 98.35767967]]
39 [[14.51901857 15.03755494 12.51901857 15.03755494]
 [14.51901857 12.51901857 15.03755494 98.52191171]]
40 [[14.53379945 15.06711671 12.53379945 15.06711671]
 [14.53379945 12.53379945 15.06711671 98.66972054]]
41 [[14.56040504 15.08041951 12.56040504 15.08041951]
 [14.56040504 12.56040504 15.08041951 98.80274848]]
42 [[14.57237755 15.10436454 12.57237755 15.10436454]
 [14.57237755 12.57237755 15.10436454 98.92247363]]
43 [[14.59392808 15.1151398  12.59392808 15.1151398 ]
 [14.59392808 12.59392808 15.1151398  99.03022627]]
44 [[14.60362582 15.13453527 12.60362582 15.13453527]
 [14.60362582 12.60362582 15.13453527 99.12720364]]
```

1. $f_R(s, a, s_f) = f_R(s_f), \gamma = 0.6$:

$$f_{M_T} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} s_f = s_1 & \\ a_1 & a_2 \end{array} \\ \begin{bmatrix} 0.4 & 0.2 \\ 0.5 & 0.0 \\ 1.0 & 0.3 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} s_f = s_2 & \\ a_1 & a_2 \end{array} \\ \begin{bmatrix} 0.5 & 0.8 \\ 0.0 & 0.0 \\ 0.0 & 0.6 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cc} s_f = s_3 & \\ a_1 & a_2 \end{array} \\ \begin{bmatrix} 0.1 & 0.0 \\ 0.5 & 1.0 \\ 0.0 & 0.1 \end{bmatrix} \end{array}$$

$$f_R(s_f) = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \\ \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} \end{array}$$

In [ ]:

```
fmt = [[
    [0.4, 0.2],
    [0.5, 0.0],
    [1.0, 0.3],
],[
    [0.5, 0.8],
    [0.0, 0.0],
    [0.0, 0.6],
],[
    [0.1, 0.0],
```

```
    [0.5, 1.0],
    [0.0, 0.1],
]]
fr = [2, 1, -1]
gamma = .6
```

a. $V(s)$

In [ ]:

```
m = len(fmt)
n = len(fmt[0])
w = len(fmt[0][0])
vs = np.zeros(n)
iteration = 0

while True:
    new_vs = vs.copy()
    for si in range(n):
        r = np.zeros(w)
        for sfi in range(m):
            for ai in range(w):
                r[ai] += fmt[sfi][si][ai] * (fr[sfi] + (gamma * vs[sfi]))
        new_vs[si] = max(r)
    print(iteration, new_vs)
    diff = new_vs - vs
    if len(diff[diff>.1]) == 0:
        break
    vs = new_vs.copy()
    iteration += 1
```

```
0 [1.2 0.5 2. ]
1 [1.758 1.46  2.72 ]
2 [2.22312 1.8434  3.0548 ]
3 [2.4698568 2.083376  3.333872 ]
4 [2.61781075 2.24111864 3.48191408]
5 [2.70952502 2.32991745 3.57068645]
```

b. $Q(s, a)$

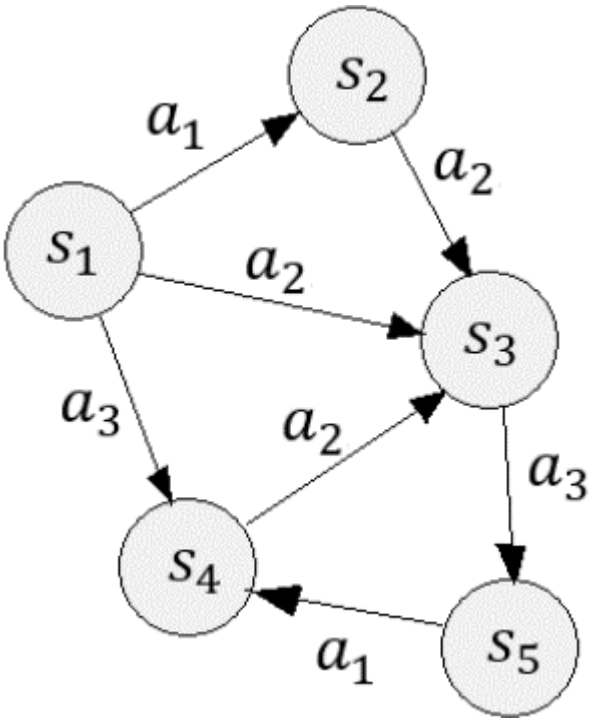
In [ ]:

```
m = len(fmt)
n = len(fmt[0])
w = len(fmt[0][0])
qsa = np.zeros((n, w))
iteration = 0
while True:
    new_qsa = np.zeros((n, w))
    for si in range(n):
        r = np.zeros(w)
        for sfi in range(m):
            for ai in range(w):
                r[ai] += fmt[sfi][si][ai] * (fr[sfi] + (gamma * max(qsa[sfi])))
        new_qsa[si] = r
    print(iteration)
    print(new_qsa.T)
    diff = new_qsa - qsa
    diff = diff.reshape(diff.size).copy()
    if len(diff[diff>.1]) == 0:
        break
```

```
    qsa = new_qsa
    iteration += 1
```

```
0
[[ 1.2  0.5  2. ]
 [ 1.2 -1.   1.1]]
1
[[1.758 1.46  2.72 ]
 [1.584 0.2   1.616]]
2
[[2.22312 1.8434  3.0548 ]
 [2.11176 0.632   2.10524]]
3
[[2.4698568 2.083376  3.333872 ]
 [2.3516064 0.83288   2.3470736]]
4
[[2.61781075 2.24111864 3.48191408]
 [2.4964033  1.0003232  2.4946219 ]]
5
[[2.70952502 2.32991745 3.57068645]
 [2.58987424 1.08914845 2.58692349]]
```

1. $\gamma = 0.8$




$$f_{M_T} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccc} a1 & a2 & a3 \end{array} \\ \begin{bmatrix} s_2 & s_3 & s_4 \\ & s_3 & \\ & & s_5 \\ & s_3 & \\ s_4 & & \end{bmatrix} \end{array}$$

$$f_R(s,a,s_f) = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} s_f = s_1 \\ \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} s_f = s_2 \\ \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} -2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} s_f = s_3 \\ \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0 & 5 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} s_f = s_4 \\ \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0 & 0 & -3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

In [ ]:

```
fmt = [
    [2,3,4],
    [0,3,0],
    [0,0,5],
    [0,3,0],
    [4,0,0],
]

fr = [[
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
],[
    [-2, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
],[
    [0, 5, 0],
    [0, 4, 0],
    [0, 0, 0],
    [0, -1, 0],
    [0, 0, 0],
],[
    [0, 0, -3],
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
    [1, 0, 0],
],[
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, -6],
    [0, 0, 0],
    [0, 0, 0],
]]

gamma = .8
```

a. $V(s)$

In [ ]:

```
m = len(fr)
n = len(fr[0])
```

```
w = len(fr[0][0])
vs: np.ndarray = np.zeros(m)
iteration = 0

while True:
    new_vs: np.ndarray = vs.copy()
    for si in range(n):
        r: np.ndarray = np.zeros(m)
        a_sf: list = fmt[si].copy()
        r = []
        for ai in range(w):
            sf_idx = fmt[si][ai]-1
            if sf_idx != -1:
                ri = fr[sf_idx][si][ai] + (gamma * vs[sf_idx])
                r.append(ri)
        new_vs[si] = max(r)
    print(iteration, new_vs)
    diff = new_vs - vs
    if len(diff[diff>.1]) == 0:
        break
    vs = new_vs.copy()
    iteration += 1
```

```
0 [ 5.  4. -6. -1.  1.]
1 [ 1.2 -0.8 -5.2 -5.8  0.2]
2 [ 0.84 -0.16 -5.84 -5.16 -3.64]
3 [ 0.328 -0.672 -8.912 -5.672 -3.128]
4 [-2.1296 -3.1296 -8.5024 -8.1296 -3.5376]
5 [-1.80192 -2.80192 -8.83008 -7.80192 -5.50368]
6 [ -2.064064  -3.064064 -10.402944  -8.064064  -5.241536]
7 [ -3.3223552  -4.3223552 -10.1932288  -9.3223552  -5.4512512]
8 [ -3.15458304  -4.15458304 -10.36100096  -9.15458304  -6.45788416]
9 [ -3.28880077  -4.28880077 -11.16630733  -9.28880077  -6.32366643]
10 [ -3.93304586  -4.93304586 -11.05893315  -9.93304586  -6.43104061]
11 [ -3.84714652  -4.84714652 -11.14483249  -9.84714652  -6.94643669]
```

b. $Q(s, a)$

In [ ]:

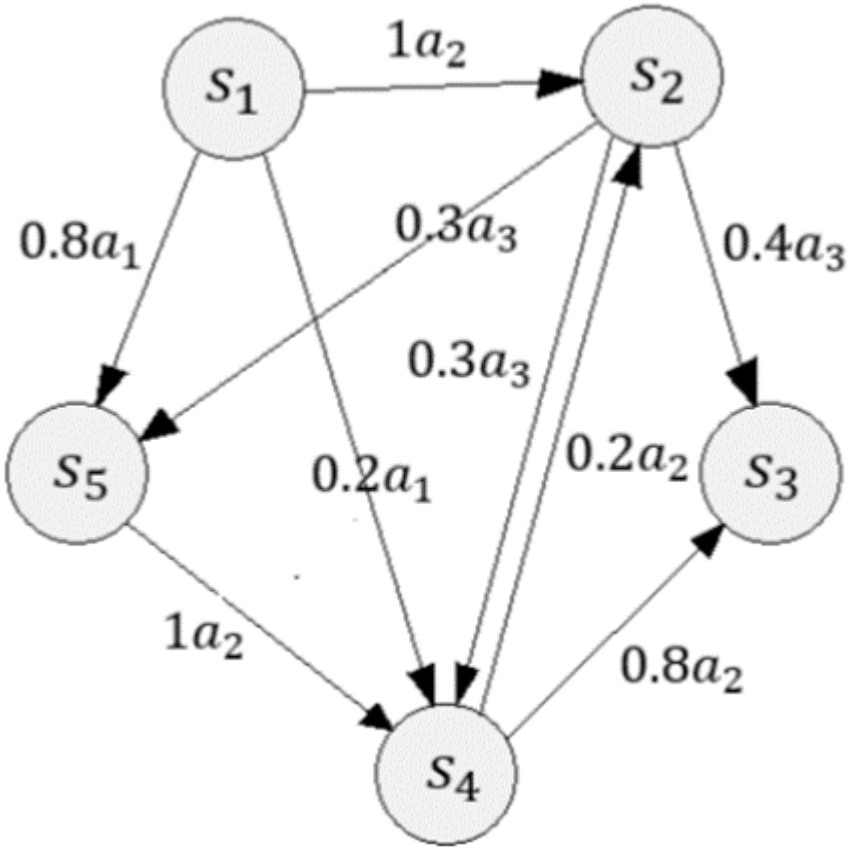
```
qsa: np.ndarray = np.zeros((m, w))
iteration = 0

while True:
    new_qsa: np.ndarray = qsa.copy()
    for si in range(n):
        a_sf = fmt[si]
        r = []
        for ai in range(len(a_sf)):
            sf_idx = a_sf[ai]-1
            ri = fr[sf_idx][si][ai] + (gamma * max(qsa[sf_idx]))
            r.append(ri)
        new_qsa[si] = np.array(r)
    print(iteration)
    print(new_qsa.T)
    diff = new_qsa - qsa
    flags = []
    for r in diff:
        flag = True
        if len(r[r>.1]) == 0:
            flag = False
```

```
        flags.append(flag)
    if sum(flags) == 0:
        break
    qsa = new_qsa
    iteration += 1
```

```
0
[[-2.  0.  0.  0.  1.]
 [ 5.  4.  0. -1.  0.]
 [-3.  0. -6.  0.  0.]]
1
[[ 1.2  0.8  0.8  0.8  1. ]
 [ 5.   4.   0.8 -1.   0.8]
 [-3.   0.8 -5.2  0.8  0.8]]
2
[[ 1.2   0.8   0.8   0.8   1.64]
 [ 5.64  4.64  0.8  -0.36  0.8 ]
 [-2.36  0.8  -5.2   0.8   0.8 ]]
3
[[ 1.712  1.312  1.312  1.312  1.64 ]
 [ 5.64   4.64   1.312 -0.36   1.312]
 [-2.36   1.312 -4.688  1.312  1.312]]
4
[[ 1.712   1.312   1.312   1.312   2.0496]
 [ 6.0496  5.0496  1.312   0.0496  1.312 ]
 [-1.9504  1.312  -4.688   1.312   1.312 ]]
5
[[ 2.03968  1.63968  1.63968  1.63968  2.0496 ]
 [ 6.0496   5.0496   1.63968  0.0496   1.63968]
 [-1.9504   1.63968 -4.36032  1.63968  1.63968]]
6
[[ 2.03968   1.63968   1.63968   1.63968   2.311744]
 [ 6.311744  5.311744  1.63968   0.311744  1.63968 ]
 [-1.688256  1.63968  -4.36032   1.63968   1.63968 ]]
7
[[ 2.2493952  1.8493952  1.8493952  1.8493952  2.311744 ]
 [ 6.311744   5.311744   1.8493952  0.311744   1.8493952]
 [-1.688256   1.8493952 -4.1506048  1.8493952  1.8493952]]
8
[[ 2.2493952   1.8493952   1.8493952   1.8493952   2.47951616]
 [ 6.47951616  5.47951616  1.8493952   0.47951616  1.8493952 ]
 [-1.52048384  1.8493952  -4.1506048   1.8493952   1.8493952 ]]
9
[[ 2.38361293  1.98361293  1.98361293  1.98361293  2.47951616]
 [ 6.47951616  5.47951616  1.98361293  0.47951616  1.98361293]
 [-1.52048384  1.98361293 -4.01638707  1.98361293  1.98361293]]
10
[[ 2.38361293  1.98361293  1.98361293  1.98361293  2.58689034]
 [ 6.58689034  5.58689034  1.98361293  0.58689034  1.98361293]
 [-1.41310966  1.98361293 -4.01638707  1.98361293  1.98361293]]
11
[[ 2.46951227  2.06951227  2.06951227  2.06951227  2.58689034]
 [ 6.58689034  5.58689034  2.06951227  0.58689034  2.06951227]
 [-1.41310966  2.06951227 -3.93048773  2.06951227  2.06951227]]
```

1. $\gamma = 0.7$:




$$f_{M_T} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \quad s_f = s_1$$

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & .2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \quad s_f = s_2$$

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & .4 \\ 0 & 0 & 0 \\ 0 & .8 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \quad s_f = s_3$$

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} .2 & 0 & 0 \\ 0 & 0 & .3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \quad s_f = s_4$$

$$f_R(s, a, s_f) = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 9 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \\ 2 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \quad s_f = s_1$$

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} -2 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \quad s_f = s_2$$

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 0 & 5 & 0 \\ 0 & 4 & -1 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \quad s_f = s_3$$

$$\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{c} a_1 \end{array} \\ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{array} \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \quad s_f$$

In [ ]:

```
fmt = [[
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
],[
    [0, 1, 0],
    [0, 0, 0],
    [0, 0, 0],
```

```
    [0, .2, 0],
    [0, 0, 0],
],[
    [0, 0, 0],
    [0, 0, .4],
    [0, 0, 0],
    [0, .8, 0],
    [0, 0, 0],
],[
    [.2, 0, 0],
    [0, 0, .3],
    [0, 0, 0],
    [0, 0, 0],
    [0, 1, 0],
],[
    [.8, 0, 0],
    [0, 0, .3],
    [0, 0, 0],
    [0, 0, 0],
    [0, 0, 0],
]]

fr = [[
    [9, 0, 0],
    [0, 0, 0],
    [0, 1, 0],
    [0, 0, -2],
    [2, 0, 0],
],[
    [-2, 2, 0],
    [0, 0, 0],
    [0, 0, 0],
    [0, -3, 0],
    [0, 0, 0],
],[
    [0, 5, 0],
    [0, 4, -1],
    [0, 0, 0],
    [0, -1, 0],
    [0, 0, 0],
],[
    [1, 0, -3],
    [0, 0, 2],
    [0, 0, 0],
    [0, 0, 0],
    [1, -1, 0],
],[
    [3, 0, 0],
    [0, 0, 0],
    [0, 0, -6],
    [0, 0, 0],
    [0, 0, 0],
]]

gamma = .8
```

a. $V(s)$

In [ ]:

```
m = len(fmt)
```

```
n = len(fmt[0])
w = len(fmt[0][0])
vs = np.zeros(m)
iteration = 0

while True:
    new_vs = np.zeros(m)
    for si in range(n):
        r = np.zeros(w)
        for sfi in range(m):
            for ai in range(w):
                r[ai] = fmt[sfi][si][ai] * (fr[sfi][si][ai] + (gamma * vs[sfi]))
        new_vs[si] = max(r)
    print(iteration, new_vs)
    diff = new_vs - vs
    if len(diff[diff>.1]) == 0:
        break
    vs = new_vs.copy()
    iteration += 1
```

```
0 [2.4 0.  0.  0.  0. ]
1 [2.4 0.  0.  0.  0. ]
```

b. $Q(s, a)$

In [ ]:

```
m = len(fmt)
n = len(fmt[0])
w = len(fmt[0][0])
qsa = np.zeros((m, w))
iteration = 0

while True:
    new_qsa = np.zeros((m, w))
    for si in range(n):
        r = np.zeros(w)
        for sfi in range(m):
            for ai in range(w):
                r[ai] += fmt[sfi][si][ai] * (fr[sfi][si][ai] + (gamma *  max(qsa[sfi])))
        new_qsa[si] = r.copy()
    print(iteration)
    print(new_qsa.T)
    diff = new_qsa - qsa
    flags = []
    for r in diff:
        flag = True
        if len(r[r>.1]) == 0:
            flag = False
        flags.append(flag)
    if sum(flags) == 0:
        break
    qsa = new_qsa.copy()
    iteration += 1
```

```
0
[[ 2.6  0.   0.   0.   0. ]
 [ 2.   0.   0.  -1.4 -1. ]
 [ 0.   0.2  0.   0.   0. ]]
1
[[ 2.6    0.     0.     0.     0.   ]
```

```
 [ 2.16   0.     0.    -1.368 -1.   ]
 [ 0.     0.2    0.     0.     0.   ]]
2
[[ 2.6    0.     0.     0.     0.   ]
 [ 2.16   0.     0.    -1.368 -1.   ]
 [ 0.     0.2    0.     0.     0.   ]]
```

1. $\gamma = 0.8$ $ f{M{T}} = \begin{matrix}
   & \begin{matrix}\rightarrow & \leftarrow\end{matrix} \\
   \begin{matrix} s_{F_{1}} \\ s_{1} \\ s_{2} \\ s_{3} \\ s_{F_{2}}
  \end{matrix} &
   \begin{bmatrix}
    s_{1} &  \\
    s_{2} & s_{F_{1}} \\
    s_{3} & s_{1} \\
    s_{F_{2}} & s_{2} \\
     & s_{3} \\
   \end{bmatrix}
  \end{matrix}
  $

$$f_{M_T} = \begin{matrix} s_{F_1} \\ s_1 \\ s_2 \\ s_3 \\ s_{F_2} \end{matrix} \begin{bmatrix} -10 \\ 0 \\ -0.4 \\ -0.4 \\ 10 \end{bmatrix}$$

In [ ]:

```
fmt = [
    [0, 2],
    [1, 3],
    [2, 4],
    [3, 5],
    [4, 0],
]

fr = [
    -10,
    0,
    -.4,
    -.4,
    10,
]

gamma = .8
```

a. $V(s)$

In [ ]:

```
m = len(fmt)
vs = np.zeros(m)

iteration = 0
while True:
    i = 0
    new_vs = np.zeros(m)
```

```
    for si in range(m):
        r = []
        for a in fmt[si]:
            sfi = a-1
            if sfi != -1:
                ri = fr[sfi] + (gamma * vs[sfi])
                r.append(ri)
        new_vs[si] = max(r)
    print(iteration, new_vs)
    diff = new_vs - vs
    if len(diff[diff>.1]) == 0:
        break
    vs = new_vs
    iteration += 1
```

```
0 [ 0.  -0.4  0.  10.  -0.4]
1 [-0.32 -0.4   7.6   9.68  7.6 ]
2 [-0.32   5.68   7.344 16.08   7.344]
3 [ 4.544   5.4752 12.464  15.8752 12.464 ]
4 [ 4.38016  9.5712  12.30016 19.9712  12.30016]
5 [ 7.65696   9.440128 15.57696  19.840128 15.57696 ]
6 [ 7.5521024 12.061568  15.4721024 22.461568  15.4721024]
7 [ 9.6492544  11.97768192 17.5692544  22.37768192 17.5692544 ]
8 [ 9.58214554 13.65540352 17.50214554 24.05540352 17.50214554]
9 [10.92432282 13.60171643 18.84432282 24.00171643 18.84432282]
10 [10.88137314 14.67545825 18.80137314 25.07545825 18.80137314]
11 [11.7403666  14.64109851 19.6603666  25.04109851 19.6603666 ]
12 [11.71287881 15.32829328 19.63287881 25.72829328 19.63287881]
13 [12.26263463 15.30630305 20.18263463 25.70630305 20.18263463]
14 [12.24504244 15.7461077  20.16504244 26.1461077  20.16504244]
15 [12.59688616 15.73203395 20.51688616 26.13203395 20.51688616]
16 [12.58562716 16.01350893 20.50562716 26.41350893 20.50562716]
17 [12.81080714 16.00450173 20.73080714 26.40450173 20.73080714]
18 [12.80360138 16.18464571 20.72360138 26.58464571 20.72360138]
19 [12.94771657 16.17888111 20.86771657 26.57888111 20.86771657]
20 [12.94310489 16.29417326 20.86310489 26.69417326 20.86310489]
21 [13.03533861 16.29048391 20.95533861 26.69048391 20.95533861]
```

b. $Q(s, a)$

In [ ]:

```
m = len(fmt)
n = len(fmt[0])
qsa = np.zeros((m, n))
iteration = 0

while True:
    new_qsa = []
    for i in range(m):
        idx1, idx2 = fmt[i]
        r1 = fr[idx1-1] + (gamma * max(qsa[idx1-1]))
        r2 = fr[idx2-1] + (gamma * max(qsa[idx2-1]))
        new_qsa.append(np.array([r1, r2]))
    new_qsa_n = np.array(new_qsa)
    diff = new_qsa_n - qsa
    flags = []
    for r in diff:
        flag = True
        if len(r[r>.1]) == 0:
            flag = False
```

```
        flags.append(flag)
    print(iteration, new_qsa_n.T)
    if sum(flags) == 0:
        break
    qsa = new_qsa_n
    iteration += 1
```

```
0 [[ 10.  -10.    0.   -0.4  -0.4]
 [  0.   -0.4  -0.4  10.   10. ]]
1 [[18.   -2.   -0.32 -0.4   7.6 ]
 [-0.32 -0.4   7.6  18.   18.  ]]
2 [[24.4   4.4  -0.32  5.68 14.  ]
 [-0.32  5.68 14.   24.4  24.4 ]]
3 [[29.52   9.52   4.544 10.8   19.12 ]
 [ 4.544 10.8   19.12  29.52  29.52 ]]
4 [[33.616 13.616  8.64  14.896 23.216]
 [ 8.64  14.896 23.216 33.616 33.616]]
5 [[36.8928 16.8928 11.9168 18.1728 26.4928]
 [11.9168 18.1728 26.4928 36.8928 36.8928]]
6 [[39.51424 19.51424 14.53824 20.79424 29.11424]
 [14.53824 20.79424 29.11424 39.51424 39.51424]]
7 [[41.611392 21.611392 16.635392 22.891392 31.211392]
 [16.635392 22.891392 31.211392 41.611392 41.611392]]
8 [[43.2891136 23.2891136 18.3131136 24.5691136 32.8891136]
 [18.3131136 24.5691136 32.8891136 43.2891136 43.2891136]]
9 [[44.63129088 24.63129088 19.65529088 25.91129088 34.23129088]
 [19.65529088 25.91129088 34.23129088 44.63129088 44.63129088]]
10 [[45.7050327 25.7050327 20.7290327 26.9850327 35.3050327]
 [20.7290327 26.9850327 35.3050327 45.7050327 45.7050327]]
11 [[46.56402616 26.56402616 21.58802616 27.84402616 36.16402616]
 [21.58802616 27.84402616 36.16402616 46.56402616 46.56402616]]
12 [[47.25122093 27.25122093 22.27522093 28.53122093 36.85122093]
 [22.27522093 28.53122093 36.85122093 47.25122093 47.25122093]]
13 [[47.80097674 27.80097674 22.82497674 29.08097674 37.40097674]
 [22.82497674 29.08097674 37.40097674 47.80097674 47.80097674]]
14 [[48.2407814 28.2407814 23.2647814 29.5207814 37.8407814]
 [23.2647814 29.5207814 37.8407814 48.2407814 48.2407814]]
15 [[48.59262512 28.59262512 23.61662512 29.87262512 38.19262512]
 [23.61662512 29.87262512 38.19262512 48.59262512 48.59262512]]
16 [[48.87410009 28.87410009 23.89810009 30.15410009 38.47410009]
 [23.89810009 30.15410009 38.47410009 48.87410009 48.87410009]]
17 [[49.09928007 29.09928007 24.12328007 30.37928007 38.69928007]
 [24.12328007 30.37928007 38.69928007 49.09928007 49.09928007]]
18 [[49.27942406 29.27942406 24.30342406 30.55942406 38.87942406]
 [24.30342406 30.55942406 38.87942406 49.27942406 49.27942406]]
19 [[49.42353925 29.42353925 24.44753925 30.70353925 39.02353925]
 [24.44753925 30.70353925 39.02353925 49.42353925 49.42353925]]
20 [[49.5388314 29.5388314 24.5628314 30.8188314 39.1388314]
 [24.5628314 30.8188314 39.1388314 49.5388314 49.5388314]]
21 [[49.63106512 29.63106512 24.65506512 30.91106512 39.23106512]
 [24.65506512 30.91106512 39.23106512 49.63106512 49.63106512]]
```

1. $\gamma = 0.6$

$$
f_{M_T} = \begin{array}{c} \\ s_{F_1} \\ s_1 \\ s_2 \\ s_3 \\ s_{F_2} \end{array}
\begin{array}{c}
\begin{array}{cc} s_f = s_{F_1} & \\ \rightarrow & \leftarrow \end{array} \\
\begin{bmatrix} 0 & 0 \\ .2 & .8 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}
\end{array}
\begin{array}{c} \\ s_{F_1} \\ s_1 \\ s_2 \\ s_3 \\ s_{F_2} \end{array}
\begin{array}{c}
\begin{array}{cc} s_f = s_1 & \\ \rightarrow & \leftarrow \end{array} \\
\begin{bmatrix} .8 & .2 \\ 0 & 0 \\ .2 & .8 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}
\end{array}
\begin{array}{c} \\ s_{F_1} \\ s_1 \\ s_2 \\ s_3 \\ s_{F_2} \end{array}
\begin{array}{c}
\begin{array}{cc} s_f = s_2 & \\ \rightarrow & \leftarrow \end{array} \\
\begin{bmatrix} 0 & 0 \\ .8 & .2 \\ 0 & 0 \\ .2 & .8 \\ 0 & 0 \end{bmatrix}
\end{array}
\begin{array}{c} \\ s_{F_1} \\ s_1 \\ s_2 \\ s_3 \\ s_{F_2} \end{array}
\begin{array}{c}
\begin{array}{cc} s_f = s_3 & \\ \rightarrow & \leftarrow \end{array} \\
\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ .8 & .2 \\ 0 & 0 \\ .2 & .8 \end{bmatrix}
\end{array}
\begin{array}{c} \\ s_{F_1} \\ s_1 \\ s_2 \\ s_3 \\ s_{F_2} \end{array}
\begin{array}{c}
\begin{array}{cc} s_f = s_{F_2} & \\ \rightarrow & \leftarrow \end{array} \\
\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ .8 & .2 \\ 0 & 0 \end{bmatrix}
\end{array}
$$

$$
f_R(s_f) = \begin{array}{c} \\ s_{F_1} \\ s_1 \\ s_2 \\ s_3 \\ s_{F_2} \end{array}
\begin{bmatrix} -10 \\ 0 \\ -0.4 \\ -0.4 \\ 10 \end{bmatrix}
$$

In [ ]:

```
fmt = [
    [
        [0, 0],
        [.2, .8],
        [0, 0],
        [0, 0],
        [0, 0],
    ],
    [
        [.8, .2],
        [0, 0],
        [.2, .8],
        [0, 0],
        [0, 0],
    ],
    [
        [0, 0],
        [.8, .2],
        [0, 0],
        [.2, .8],
        [0, 0],
    ],
    [
        [0, 0],
        [0, 0],
        [.8, .2],
        [0, 0],
        [.2, .8],
    ],
    [
        [0, 0],
        [0, 0],
        [0, 0],
        [.8, .2],
        [0, 0],
    ],
]

fr = [-10, 0, -0.4, -0.4, 10]

gamma = 0.6
```

a. $V(s)$

In [ ]:
```
m = len(fmt)
n = len(fmt[0])
w = len(fmt[0][0])
vs = np.zeros(n)
iteration = 0

while True:
    new_vs = vs.copy()
    for si in range(n):
        r = np.zeros(w)
        for sfi in range(m):
            for ai in range(w):
                r[ai] += fmt[sfi][si][ai] * (fr[sfi] + (gamma * vs[sfi]))
        new_vs[si] = max(r)
    print(iteration, new_vs)
    diff = new_vs - vs
    if len(diff[diff>.1]) == 0:
        break
    vs = new_vs.copy()
    iteration += 1
```

```
0 [ 0.   -2.32 -0.08  7.92 -0.08]
1 [-0.2784 -2.3584  3.2032  7.872   3.4816]
2 [-0.283008 -0.815872  3.175552  9.975552  3.45856 ]
3 [-0.09790464 -0.829696    4.37036032  9.96117504  4.46826496]
4 [-0.09956352 -0.2339756   4.3618005  10.58921042  4.46136402]
5 [-0.02807707 -0.23828338  4.73474393 10.58487079  4.762821  ]
6 [-0.02859401 -0.05069216  4.73214397 10.77432335  4.76073798]
7 [-6.08305954e-03 -5.20021737e-02  4.84559215e+00  1.07730115e+01
  4.85167521e+00]
8 [-6.24026085e-03  5.15426459e-03  4.84480526e+00  1.08302752e+01
  4.85104552e+00]
```

b. $Q(s,a)$

In [ ]:
```
m = len(fmt)
n = len(fmt[0])
w = len(fmt[0][0])
qsa = np.zeros((n, w))
iteration = 0
while True:
    new_qsa = np.zeros((n, w))
    for si in range(n):
        r = np.zeros(w)
        for sfi in range(m):
            for ai in range(w):
                r[ai] += fmt[sfi][si][ai] * (fr[sfi] + (gamma * max(qsa[sfi])))
        new_qsa[si] = r
    print(iteration)
    print(new_qsa.T)
    diff = new_qsa - qsa
    diff = diff.reshape(diff.size).copy()
    if len(diff[diff>.1]) == 0:
        break
    qsa = new_qsa
    iteration += 1
```

```
0
[[ 0.   -2.32 -0.32  7.92 -0.08]
 [ 0.   -8.08 -0.08  1.68 -0.32]]
1
[[-1.1136 -2.3584  3.2032  7.872   0.8704]
 [-0.2784 -8.0896 -0.2432  1.632   3.4816]]
2
[[-1.132032 -0.815872  3.175552  9.975552  0.86464 ]
 [-0.283008 -7.829248 -0.267392  3.635328  3.45856 ]]
3
[[-0.39161856 -0.829696    4.37036032  9.96117504  1.11706624]
 [-0.09790464 -7.8347776   0.72544768  3.61929216  4.46826496]]
4
[[-0.39825408 -0.2339756   4.3618005  10.58921042  1.115341  ]
 [-0.09956352 -7.60255099  0.71708692  4.31396475  4.46136402]]
5
[[-0.11230829 -0.23828338  4.73474393 10.58487079  1.19070525]
 [-0.02807707 -7.60437443  1.07839696  4.30902792  4.762821  ]]
6
[[-0.11437602 -0.05069216  4.73214397 10.77432335  1.19018449]
 [-0.02859401 -7.52530772  1.07580847  4.52421561  4.76073798]]
7
[[-2.43322382e-02 -5.20021737e-02  4.84559215e+00  1.07730115e+01
   1.21291880e+00]
 [-6.08305954e-03 -7.52586785e+00  1.18858656e+00  4.52271766e+00
   4.85167521e+00]]
8
[[-2.49610434e-02  5.15426459e-03  4.84480526e+00  1.08302752e+01
   1.21276138e+00]
 [-6.24026085e-03 -7.50144881e+00  1.18780034e+00  4.58808526e+00
   4.85104552e+00]]
```