In [11]:
```
import numpy as np
import matplotlib.pyplot as plt
import sympy
from itertools import product
```

Question 1

In [3]:
```
# Question 1.1
P = np.array([
    [0, 0.5, 0.5, 0, 0, 0, 0, 0],
    [0.3, 0.7, 0.0, 0, 0, 0, 0, 0],
    [0.0, 0.0, 0.0, 0.7, 0, 0, 0, 0.3],
    [0.0, 0.0, 0.0, 0.5, 0.5, 0, 0, 0],
    [0.0, 0.0, 0.0, 0.0, 0.6, 0.4, 0, 0],
    [0.0, 0.0, 0.0, 0.0, 0., 0.5, 0.5, 0],
    [0.0, 0.0, 0.0, 0.0, 0., 0.4, 0., 0.6],
    [0.0, 0.0, 0.0, 0.0, 0., 0.4, 0., 0.6],
])
```

In [4]:
```
A = P.T - np.eye(P.shape[0])
A = np.hstack((A, np.zeros(P.shape[0]).reshape(-1, 1)))
mat = sympy.Matrix(A)
print(mat.rref())
```

```
(Matrix([
[1, 0, 0, 0, 0, 0, 0,                    0, 0],
[0, 1, 0, 0, 0, 0, 0,                    0, 0],
[0, 0, 1, 0, 0, 0, 0, 2.06514699521048e-16, 0],
[0, 0, 0, 1, 0, 0, 0, 2.89120579329468e-16, 0],
[0, 0, 0, 0, 1, 0, 0, 1.44560289664734e-16, 0],
[0, 0, 0, 0, 0, 1, 0,    -1.33333333333333, 0],
[0, 0, 0, 0, 0, 0, 1,   -0.666666666666667, 0],
[0, 0, 0, 0, 0, 0, 0,                    0, 0]]), (0, 1, 2, 3, 4, 5, 6))
```

In [5]:
```
x = np.array([0, 0, 0, 0, 0, 4/3, 2/3, 1]).reshape(-1, 1)
print(P.T @ x)
```

```
[[0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [0.        ]
 [1.33333333]
 [0.66666667]
 [1.        ]]
```

In [6]:
```
# Question 1.3
Q = np.array([
    [0, 0.5, 0.5, 0, 0],
    [0.3, 0.7, 0.0, 0, 0],
    [0.0, 0.0, 0.0, 0.7, 0],
    [0.0, 0.0, 0.5, 0.0, 0.5],
    [0.0, 0.0, 0.0, 0.0, 0.6]
])
Z = np.eye(Q.shape[0]) - Q
```

```
Z_inv = np.linalg.inv(Z)
print(Z_inv)
```

```
[[2.         3.33333333 1.53846154 1.07692308 1.34615385]
 [2.         6.66666667 1.53846154 1.07692308 1.34615385]
 [0.         0.         1.53846154 1.07692308 1.34615385]
 [0.         0.         0.76923077 1.53846154 1.92307692]
 [0.         0.         0.         0.         2.5       ]]
```

In [10]:
```
# Question 1.4
sub_A = P[:7, :7]
sub_A = np.eye(sub_A.shape[0]) - sub_A
sub_mat = np.hstack([sub_A, np.ones(sub_A.shape[0]).reshape(-1, 1)])
sub_mat = sympy.Matrix(sub_mat)
print(sub_mat.rref())
```

```
(Matrix([
[1, 0, 0, 0, 0, 0, 0, 12.9833333333333],
[0, 1, 0, 0, 0, 0, 0, 16.3166666666667],
[0, 0, 1, 0, 0, 0, 0,              7.65],
[0, 0, 0, 1, 0, 0, 0,               9.5],
[0, 0, 0, 0, 1, 0, 0,               7.5],
[0, 0, 0, 0, 0, 1, 0,               5.0],
[0, 0, 0, 0, 0, 0, 1,               3.0]]), (0, 1, 2, 3, 4, 5, 6))
```

Question 2

In [13]:
```
P = np.array([
    [0.1, 0., 0.9],
    [0.7, 0.3, 0.0],
    [0.0, 0.4, 0.6]
])
def visit_cost(state):
    return 2 * state
def terminate_cost(state):
    return -1 * state
```

In [10]:
```
# Question 2.1
p1 = P[0,0] * P[0,0] * P[0,2] * P[2,1]

p2 = P[0,0] * P[0,2] * P[2,1] * P[1,0]

p3 = P[0,0] * P[0,2] * P[2,1] * P[1,1]

p4 = P[0,0] * P[0,2] * P[2,2] * P[2,1]

p5 = P[0,2] * P[2,1] * P[1,0] * P[0,0]

p6 = P[0,2] * P[2,1] * P[1,0] * P[0,2]

p7 = P[0,2] * P[2,1] * P[1,1] * P[1,1]

p8 = P[0,2] * P[2,1] * P[1,1] * P[1,0]

p9 = P[0,2] * P[2,2] * P[2,1] * P[1,0]

p10 = P[0,2] * P[2,2] * P[2,1] * P[1,1]
```

```
p11 = P[0,2] * P[2,2] * P[2,2] * P[2,1]
print(p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 + p11)
```

```
0.7668
```

In [17]:
```
# Question 2.2
states = [0, 1, 2]
T = 4
initial_state = 0

valid_paths = []
valid_path_probs = []
valid_costs = []

# Generate all paths of T=4
for path in product(states, repeat=T):
    full_path = (initial_state,) + path
    if set(full_path) == {0, 1, 2}:
        prob = 1.0
        cost = 0.0
        for i in range(1, len(full_path)):
            prob *= P[full_path[i - 1], full_path[i]]
        valid_paths.append(full_path)
        valid_path_probs.append(prob)
        for i in range(0, len(full_path) - 1):
            cost += visit_cost(full_path[i]+1)
        cost += terminate_cost(full_path[-1] + 1)
        valid_costs.append(cost * prob)
print(sum(valid_costs))
```

```
11.145600000000002
```

In [ ]: