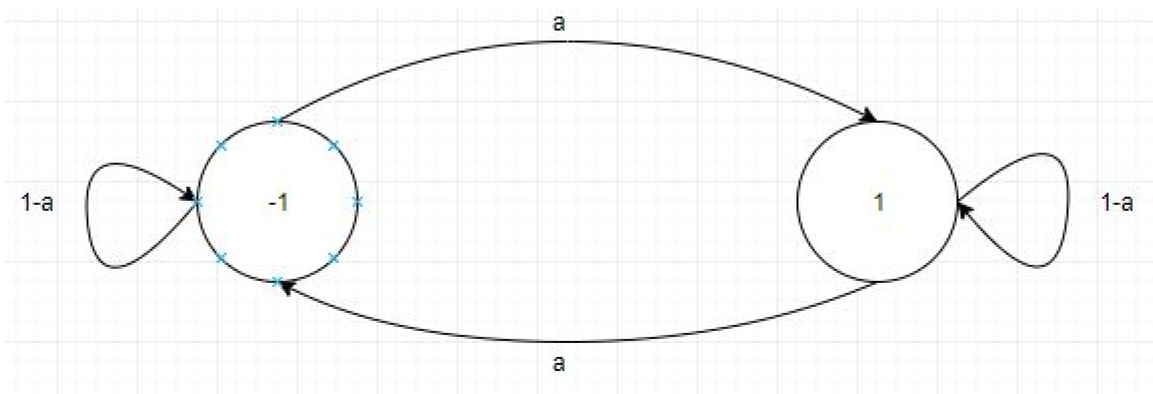
Matthew Bennett
Homework 3

1. A.
$P(X_{n+2}=3|X_n=4)=P(X_2=3|X_n=4)=P(X_2=3,X_1=4|X_0=4)+P(X_2=3,X_1=3|X_0=4)$
$\Rightarrow \frac{P(X_2=3,X_1=4,X_0=4)+P(X_2=3,X_1=3|X_0=4)}{P(X_0=4)}=\frac{(0.75)(0.25)(0.25)+(0.25(0.25)(0.5)}{(0.25)}=0.3125$
B. $P(X_{n+2}=1,X_{n+3}=2|X_n=0)$

$=P(X_2=1,X_3=2|X_0=0)=P(X_2=1,X_3=2,X_1=0|X_0=0)+P(X_2=1,X_3=2,X_1=1|X_0=0)+$
$P(X_2=1,X_3=2,X_1=2|X_0=0)=\frac{(0.63)[(0.25)(0.5)(0.25)+(0.5)(0.5)(0.25)+(0.25)(0.25)(0.25)]}{0.63}=0.1094$
C.
$P(X_0=3,X_1=3,X_2=4)+P(X_0=3,X_1=4,X_2=4)=(0.25)[(0.5)(0.25)+(0.25)(0.75)]=0.07813$
D. $=(0.5)(0.25)+(0.25)(0.75)=0.1875$
E. $\sum_{n=1}^{\infty}(0.5)^n=1day$

2.



.

a. (Diagram above)
b.
   i. $f_0^{(1)}=1-a$
   ii. $f_0^{(2)}=a^2$
   iii. $f_0^{(3)}=a^2(1-a)$
   iv. Greatest common denominator is 1=> period
c. Given the period is 1, intuitively the chance of the of returning to state -1 would be 1 because it is going to recur again within the period.
d. Being that the period is and that there is two states to jump from then the mean will be 2.
e. The limiting case will be independent of the initial system, because of this it will evenly be distributed: (0.5,0.5),(0.5,0.5)
f. This will give a steady state vector as (0.5,0.5)

3.

a. ${f_0}^{(1)} = 1 - p$
b. ${f_0}^{(2)} = p(1 - p)$
c. ${f_0}^{(3)} = 0$
d. ${f_0}^{(4)} = p^2(1 - p)^2$
e. ${f_0}^{(6)} = 2p^3(1 - p)^3$
f. ${f_0}^{(8)} = 5p^4(1 - p)^4$

4. (Program below)

```
import numpy as np

def transistion_p_matrix(matrix, steps):
    matrix_cp = np.copy(matrix)

    for s in range(steps-1):
        matrix = np.matmul(matrix, matrix_cp)

    return matrix
```

A-f are below:

```python
matrix = np.array([[0.4,0.4,0.2,0,0],
                   [0.4,0.4,0.2,0,0],
                   [0,0.4,0.4,0.2,0],
                   [0,0,0.4,0.4,0.2],
                   [0,0,0,0.4,0.6]])
transistion_p_matrix(matrix, 2)
```

```
array([[0.32, 0.4 , 0.24, 0.04, 0.  ],
       [0.32, 0.4 , 0.24, 0.04, 0.  ],
       [0.16, 0.32, 0.32, 0.16, 0.04],
       [0.  , 0.16, 0.32, 0.32, 0.2 ],
       [0.  , 0.  , 0.16, 0.4 , 0.44]])
```

```python
matrix = np.array([[0.4,0.4,0.2,0,0],
                   [0.4,0.4,0.2,0,0],
                   [0,0.4,0.4,0.2,0],
                   [0,0,0.4,0.4,0.2],
                   [0,0,0,0.4,0.6]])
transistion_p_matrix(matrix, 4)
```

```
array([[0.2688, 0.3712, 0.2624, 0.08  , 0.0176],
       [0.2688, 0.3712, 0.2624, 0.08  , 0.0176],
       [0.2048, 0.32  , 0.2752, 0.1376, 0.0624],
       [0.1024, 0.2176, 0.2752, 0.24  , 0.1648],
       [0.0256, 0.1152, 0.2496, 0.3296, 0.28  ]])
```

```python
matrix = np.array([[0.4,0.4,0.2,0,0],
                   [0.4,0.4,0.2,0,0],
                   [0,0.4,0.4,0.2,0],
                   [0,0,0.4,0.4,0.2],
                   [0,0,0,0.4,0.6]])
transistion_p_matrix(matrix, 8)
```

```
array([[0.23441408, 0.34097152, 0.26655744, 0.1123072 , 0.04574976],
       [0.23441408, 0.34097152, 0.26655744, 0.1123072 , 0.04574976],
       [0.21311488, 0.32      , 0.26688512, 0.13344256, 0.06655744],
       [0.17117184, 0.27805696, 0.26688512, 0.1753856 , 0.10850048],
       [0.12988416, 0.23611392, 0.26622976, 0.21700096, 0.1507712 ]])
```

```python
matrix = np.array([[0.4,0.4,0.2,0,0],
                   [0.4,0.4,0.2,0,0],
                   [0,0.4,0.4,0.2,0],
                   [0,0,0.4,0.4,0.2],
                   [0,0,0,0.4,0.6]])
transistion_p_matrix(matrix, 16)
```

```
array([[0.21685184, 0.32351844, 0.2666666 , 0.12981486, 0.06314827],
       [0.21685184, 0.32351844, 0.2666666 , 0.12981486, 0.06314827],
       [0.21333319, 0.32      , 0.26666681, 0.1333334 , 0.0666666 ],
       [0.20629632, 0.31296313, 0.26666681, 0.14037028, 0.07370347],
       [0.19925987, 0.30592625, 0.26666638, 0.14740694, 0.08074056]])
```

```python
matrix = np.array([[0.4,0.4,0.2,0,0],
                   [0.4,0.4,0.2,0,0],
                   [0,0.4,0.4,0.2,0],
                   [0,0,0.4,0.4,0.2],
                   [0,0,0,0.4,0.6]])
transistion_p_matrix(matrix, 32)
```

```
array([[0.21343237, 0.32009904, 0.26666667, 0.1332343 , 0.06656763],
       [0.21343237, 0.32009904, 0.26666667, 0.1332343 , 0.06656763],
       [0.21333333, 0.32      , 0.26666667, 0.13333333, 0.06666667],
       [0.21313526, 0.31980193, 0.26666667, 0.1335314 , 0.06686474],
       [0.21293719, 0.31960386, 0.26666667, 0.13372947, 0.06706281]])
```

```python
matrix = np.array([[0.4,0.4,0.2,0,0],
                   [0.4,0.4,0.2,0,0],
                   [0,0.4,0.4,0.2,0],
                   [0,0,0.4,0.4,0.2],
                   [0,0,0,0.4,0.6]])
transistion_p_matrix(matrix, 64)
```

```
array([[0.21333341, 0.32000008, 0.26666667, 0.13333325, 0.06666659],
       [0.21333341, 0.32000008, 0.26666667, 0.13333325, 0.06666659],
       [0.21333333, 0.32      , 0.26666667, 0.13333333, 0.06666667],
       [0.21333318, 0.31999984, 0.26666667, 0.13333349, 0.06666682],
       [0.21333302, 0.31999969, 0.26666667, 0.13333365, 0.06666698]])
```