In [11]:
```
import numpy as np
import pandas as pd
from random import seed
from random import random
import matplotlib.pyplot as plt
```

# Problem 1

### (a)

In [33]:
```
P = np.array([[0.4, 0.38, 0.22],
              [0.12, 0.7, 0.18],
              [0.2, 0.5, 0.3]])
states_arr = np.array([1,2,3])
```

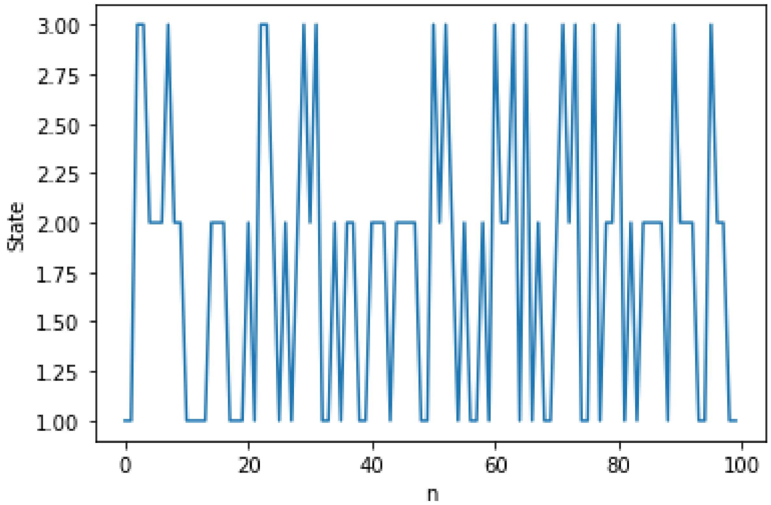
In [34]:
```
n = 100
X = np.zeros(100)
X[0] = 1
states.shape
```

Out[34]:
```
(3,)
```

In [51]:
```
def state_dist(n,P):
    current = X[0]
    states = [X[0]]
    count = 1
    while count < n:
        if current == 1:
            next_state = np.random.choice(states_arr,replace=True,p=P[0])
            states.append(next_state)
        elif current == 2:
            next_state = np.random.choice(states_arr,replace=True,p=P[1])
            states.append(next_state)
        elif current == 3:
            next_state = np.random.choice(states_arr,replace=True,p=P[2])
            states.append(next_state)
        count += 1
    return states
```

In [52]:
```
plt.plot(np.arange(0,100) ,state_dist(100,P))
plt.xlabel('n')
plt.ylabel('State')
```

Out[52]: Text(0, 0.5, 'State')




**(b)**

In [92]:
```
X = state_dist(1000,P)
print("Sum X: ", np.sum(X)/1000)
```

```
Sum X:  1.819
```

In [97]:
```
X = state_dist(1000,P)
X2 = np.square(X)
B = [0.9**i for i in range(1000)]
print("Sum B*X^2:" , np.dot(X2, B))
```

```
Sum B*X^2: 38.4147074904899
```

**(c)**

In [81]:
```
def average_state():
    result = []
    for i in range(1000):
        X = state_dist(1000,P)
        sums = np.sum(X)/1000
        result.append(sums)
    return np.mean(result)
```

In [98]:
```
print("Average state:", average_state())
```

```
Average state: 1.819986
```

In [84]:
```
def average_stateB():
    result = []
    B = [0.9**i for i in range(1000)]
    for i in range(1000):
        X = state_dist(1000,P)
        X2 = np.square(X)
        result.append(np.dot(X2, B))
    return np.mean(result)
```

In [99]:
```
print("Long run reward:", average_stateB())
```

```
Long run reward: 35.97793051594115
```

**(d)**

1. $P = \begin{bmatrix} 0.4 & 0.38 & 0.22 \\ 0.12 & 0.7 & 0.18 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}$

$\pi_1 = 0.4\pi_1 + .12\pi_2 + .2\pi_3$
$\pi_2 = .38\pi_1 + .7\pi_2 + .5\pi_3$
$\pi_3 = .22\pi_1 + .18\pi_2 + .3\pi_3$
$1 = \pi_1 + \pi_2 + \pi_3$

(d) $E\left[\sum_{K=0}^{n} \beta^K (X_K)^2 \mid X_0 = 1\right] = V_K$

$\pi_j = \sum_i \pi_i P_{ij}$ & $\sum_j \pi_j = 1$

$V(S) = R(S) + \beta \sum_{S'} P(S'|S) V(S')$

$V(1) = X_1^2 + \beta(P_{12}V(2) + P_{13}V(3) + P_{11}(V_1))$
$V(2) = X_2^2 + \beta(P_{21}V(1) + P_{22}V(2) + P_{23}V(3))$
$V(3) = X_3^2 + \beta(P_{31}V(1) + P_{32}V(2) + P_{33}V(3))$

$V(1) = X_1^2 + (.9)(.38)V(2) + (.9)(.22)V(3) + (.9)(.4)V(1)$
$V(2) = X_2^2 + (.9)(.12)V(1) + (.9)(.7)V(2) + (.9)(.18)V(3)$
$V(3) = X_3^2 + (.9)(.2)V(1) + (.9)(.5)V(2) + (.9)(.3)V(3)$

$V(1) = X_1^{2} \; 1 + .36V(1) + .342V(2) + .198V(3)$
$V(2) = X_2^{2} \; 4 + .108V(1) + .63V(2) + .162V(3)$
$V(3) = X_3^{2} \; 9 + .18V(1) + .45V(2) + .27V(3)$

Solved in Python:

$V(1) = \frac{1693885}{41539} = 40.778$ $V(2) = \frac{1,849,885}{41,539} = 44,534$ $V(3) = \frac{2,070,135}{41,539} = 49,836$

In the simulation, we got 36.019 which is slightly lower than $V_1 = 40.778$.