Solution to HW3

Phudish Prateepamornkul

Question 1(a)

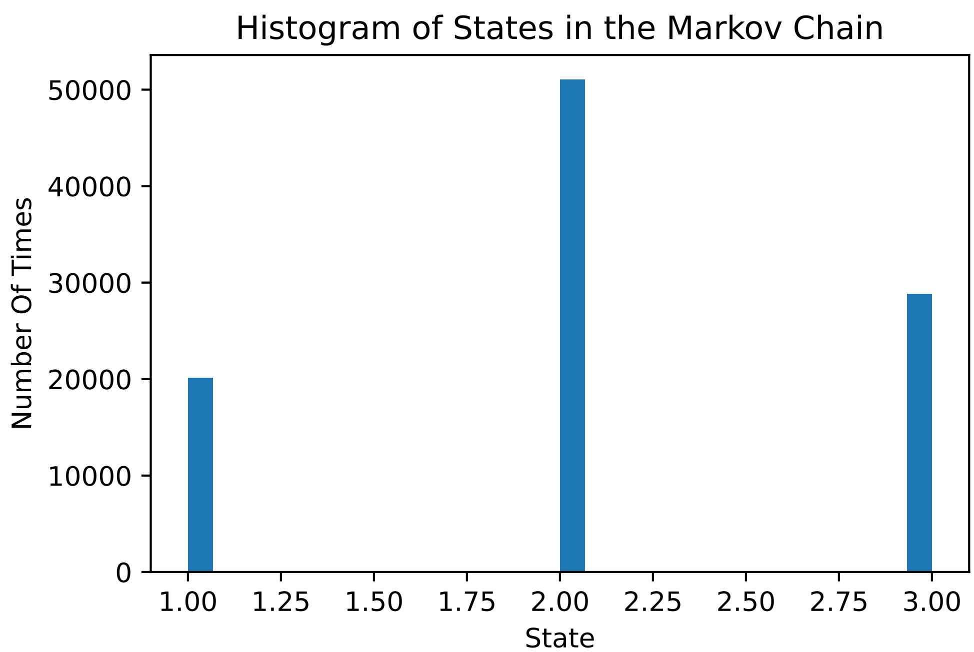
We have that the transition matrix will be

$$\begin{pmatrix} 0.2 & 0.7 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{pmatrix}$$

Question 1(b)

```
import numpy as np
import matplotlib.pyplot as plt
P = np.array([[0.2, 0.7, 0.1], [0.2, 0.5, 0.3], [0.2, 0.4, 0.4]])
states = [1, 2, 3]
current_state = 1
number_time = 100000
state_sequence = [current_state]
for i in range(number_time - 1):
    current_state = np.random.choice(states, p=P[current_state - 1])
    state_sequence.append(current_state)

plt.hist(state_sequence, bins=30)
plt.xlabel("State")
plt.ylabel("Number Of Times")
plt.title("Histogram of States in the Markov Chain")
plt.show()
```



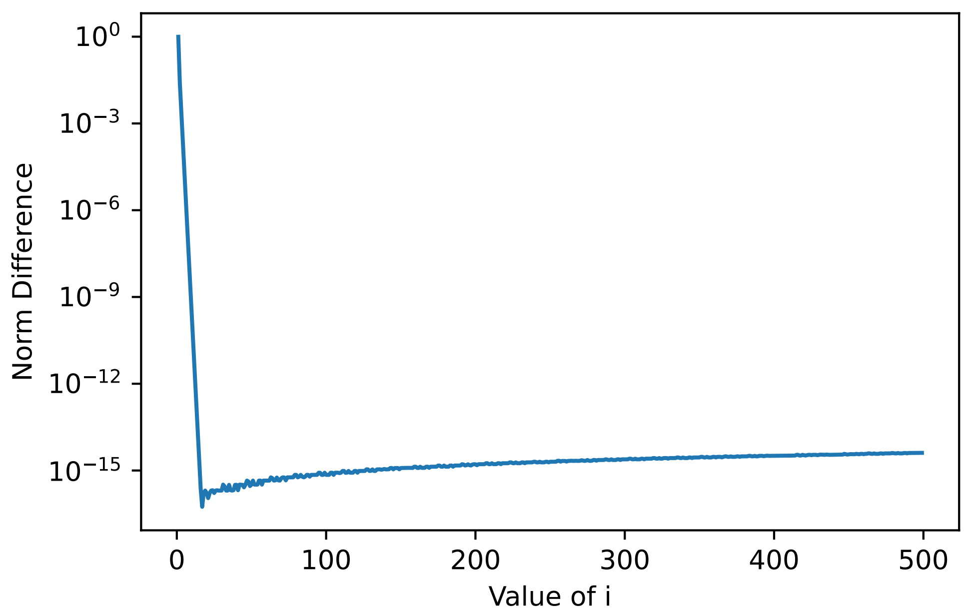
Question 2(a)

```
import scipy
identity = np.identity(np.shape(P)[0])
term_1 = P.T - identity
solution =  scipy.linalg.null_space(term_1)
normalized_solution = solution/(np.sum(solution))
print(f"The stationary state is : {normalized_solution}")
pi_0 = np.array([[1,0,0]])
pi_0 = np.reshape(pi_0,(3,1))
```

```
The stationary state is : [[0.2       ]
 [0.51111111]
 [0.28888889]]
```

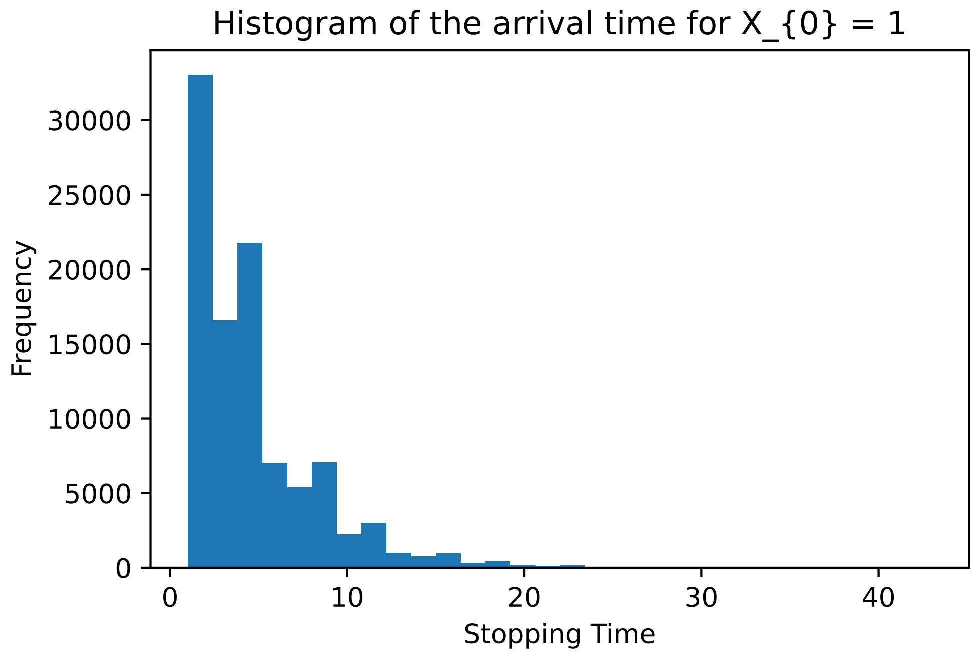
Question 2(b)

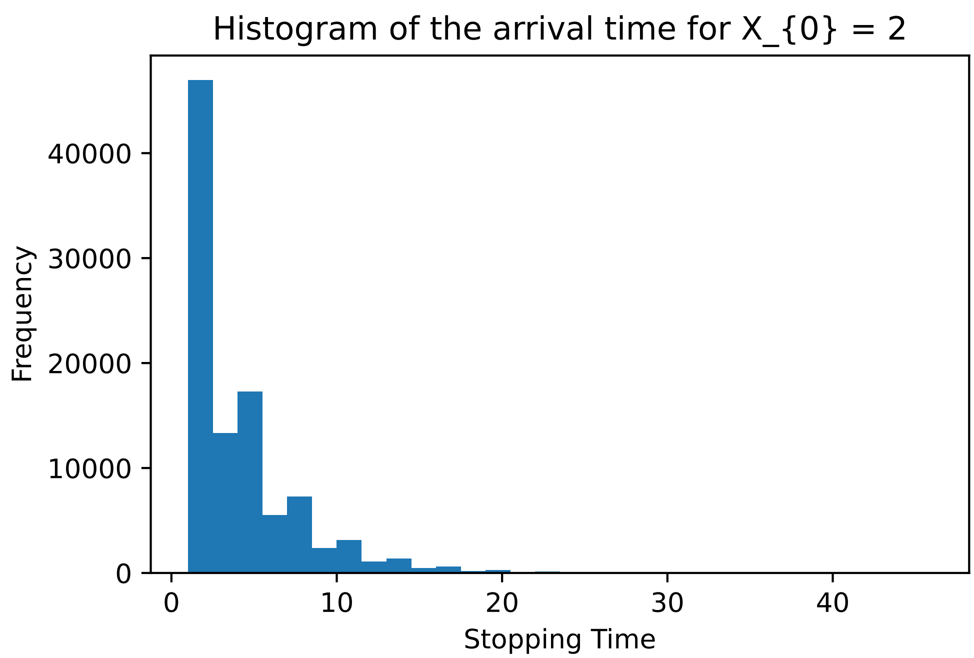
```
number_of_i = 500
value_i =  np.arange(1,number_of_i)
pi_i = [pi_0]
for i in range(2,number_of_i):
    pi_i.append(((pi_0.T)@np.linalg.matrix_power(P,i)).T)
difference = []
for i in range(len(value_i)):
    difference.append(np.linalg.norm(pi_i[i] - normalized_solution))
plt.plot(value_i,difference)
plt.xlabel("Value of i")
plt.ylabel("Norm Difference")
plt.yscale("log")
plt.show()
```



Question 3(a)

```
X_0_start_1 = np.array([1, 0, 0])
X_0_start_2 = np.array([0, 1, 0])
number_time = 100000
state = [1, 2, 3]
P = np.array([[0.2, 0.7, 0.1], [0.2, 0.5, 0.3], [0.2, 0.4, 0.4]])
def arrival_time(initial_probability):
    number_steps = 0
    current_state = np.random.choice(state, p = initial_probability)
    while current_state !=3 :
        state_1 =  np.zeros(3)
        state_1[current_state-1] = 1
        current_state = np.random.choice(state,p = state_1@P)
        number_steps += 1
    return number_steps
first_time = [arrival_time(initial_probability=X_0_start_1) for j in
 ↪  range(number_time)]
second_time = [arrival_time(initial_probability=X_0_start_2) for j in
 ↪  range(number_time)]
plt.hist(first_time,bins = 30)
plt.xlabel("Stopping Time")
plt.ylabel("Frequency")
plt.title("Histogram of the arrival time for X_{0} = 1")
plt.show()
plt.hist(second_time,bins = 30)
plt.xlabel("Stopping Time")
plt.ylabel("Frequency")
plt.title("Histogram of the arrival time for X_{0} = 2")
plt.show()
print(f"The mean for the stopping time when initial is 1 :
 ↪  {np.mean(first_time)}")
print(f"The mean for the stopping time when initial is 2 :
 ↪  {np.mean(second_time)}")
```






```
The mean for the stopping time when initial is 1 : 4.6136
The mean for the stopping time when initial is 2 : 3.84274
```

Question 3(b)

We have that $\mu_1 = 1 + \sum_{j=1}^{3} p_{1j}\mu_j$ and $\mu_2 = 1 + \sum_{j=1}^{3} p_{2j}\mu_j$ So we have that $\mu_1 = 1 + p_{11}\mu_1 + p_{12}\mu_2$ and $\mu_2 = 1 + p_{21}\mu_1 + p_{22}\mu_2$ and $p_{11} = 0.2, p_{12} = 0.7, p_{21} = 0.2, p_{22} = 0.5$. So we have $0.8\mu_1 - 0.7\mu_2 = 1$ and $-0.2\mu_1 + 0.5\mu_2 = 1$ then we multiply 4 to the second equation and we add them up we will have that $1.3\mu_2 = 5$ so $\mu_2 = \frac{50}{13} = 3.846$ and $\mu_1 = \frac{60}{13} = 4.6153$ which is close to what we have on part(a)