Question 1

The sample and state spaces are given by:

$$\Omega = [0,1]^{qn}$$
$$\mathbb{S} = \{0, 1, \ldots, qn\}$$

X_k is a random variable because we can express X_k as a function from Ω to S. Since a stochastic process is defined as a collection of random variables, $X = \{X_k\}_{k \in \mathbb{N}}$ is a stochastic process.

Question 2

The possible transitions of X for any state $x \in \mathbb{S}$ are $x-1, x, \text{ and } x+1$. In the case of transition to x-1, we take a fast particle from the left chamber and move it to the right. In the case of transition to x, we pick a slow particle from the left or right chamber and because of the demon, it will always end up in the right chamber. In the case of transition to x+1, we take a fast particle from the right chamber and move it to the left.

$$\text{For } q \neq 0, 1 \text{ and } n \neq 0, \text{we have :}$$
$$x \rightarrow \begin{cases} x - 1 \text{ w.p.} \frac{x}{n} \\ x \text{ w.p.} 1 - q \\ x + 1 \text{ w.p.} \frac{qn - x}{n} \end{cases}$$

We can represent these values in a transition diagram (see Figure 1 at the end of this document).

Question 3

We can write our transition probabilities for X_k like so:

$$p(x, z) = \begin{cases} \frac{x}{n} \text{ where } z = x - 1 \\ 1 - q \text{ where } z = x \\ \frac{qn - x}{n} \text{ where } z = x + 1 \end{cases}$$

X_k is a Markov Chain because we can assign transition probabilities from $x \rightarrow z$ only dependant on the current state X_k and constants.

Question 4

First, let us show the state space for Y_k:

$$\mathbb{S} = \{0, 1, \ldots, (1-q)n\}$$

Next, the possible transitions of Y for any state y are $y-1$ and y. In the case of transition to $y-1$, we take a slow particle from the left chamber and move it to the right. In the case of transition to y, we either pick a fast particle and move it or we pick a slow particle from the right chamber and the demon stops it from being moved to the left chamber.

$$\text{For } n \neq 0 \text{, we have :}$$
$$y \to \begin{cases} y - 1 \text{ w.p.} \frac{y}{n} \\ y \text{ w.p.} \frac{n-y}{n} \end{cases}$$

We can represent these values in a transition diagram (see Figure 2 at the end of this document).

Next, here is the corresponding transition probabilities according to the diagram and our earlier calculations:

$$p(y, z) = \begin{cases} \frac{y}{n} \text{ where } z = y - 1 \\ \frac{n-y}{n} \text{ where } z = y \end{cases}$$

Y is a Markov Chain because we can assign transition probabilities from $y \to z$ only dependent on the current state Y_k and constants.

Question 5

First, let's show that L_k is a random variable by defining the sample and state spaces:

$$\Omega = [0,1]^n$$
$$\mathbb{S} = \{0, 1, \ldots, n\}$$

Since we can express L_k as a function from $\Omega \to \mathbb{S}$, we know that L_k is a random variable. Further, we are given that $L = \{L_k\}_{k \in \mathbb{N}}$ so we can conclude that L is a stochastic process because L is a collection of random variables.

Now, let's try and show the possible transitions for L_k to see if L is a Markov Chain. Our possible transitions are to $l-1, l, \text{ and } l+1$. In the case of transition to $l-1$, we take either a slow or fast particle from the left chamber and move it to the right. In the case of transition to l, we take a slow particle from the right chamber and the demon stops it from going to the left chamber. In the case of transition to $l+1$, we take a fast particle from the right chamber and move it to the left.

$$\text{For } l \neq 0, n, \text{ and } n \neq 0 \text{ we have}:$$
$$l \to \begin{cases} l-1 \text{ w.p.} \frac{x+y}{n} \\ l \text{ w.p.} \frac{(1-q)n-y}{n} \\ l+1 \text{ w.p.} \frac{x-qn}{n} \end{cases}$$

Since these transitions depend on states X_k and Y_k which are other than the current state of L_k, L is not a Markov Chain.

Question 6

We will use the tower property to calculate the expectation of X_k. Specifically, we will use the property that $\mu_{k+1} = \mathbb{E}[\mathbb{E}[X_{k+1}|X_k]]$ and use reduction to solve for μ_k:

$$\mathbb{E}[X_{k+1}|X_k] = (\frac{qn - x_k}{n})(x_k + 1) + (1-q)x_k + (\frac{x_k}{n})(x_k - 1)$$

Next, let's use this equation to solve for μ_k:

$$\mu_k = (1 - \frac{2}{n})\mu_{k-1} + q$$

$$\text{Let } A = 1 - \frac{2}{n}. \text{ Then we have:}$$

$$\begin{aligned} \mu_k &= A\mu_{k-1} + q \\ &= A(A\mu_{k-2} + q) + q = A^2\mu_{k-2} + q + \\ &= A^2(A\mu_{k-3} + q) + q + Aq = A^3\mu_{k-3} \\ &= A^k\mu_0 + q(1 + A + A^2 \ldots + A^{k-1}) \\ &= A^k\mu_0 + q\frac{1-A^k}{1-A} \end{aligned}$$

$$\text{Therefore } \mathbb{E}[X_k] = (1 - \frac{2}{n})^k z + q\frac{1-A^k}{1-A}$$

$$\lim_{k \to \inf} \mathbb{E}[X_k] = \frac{nq}{2}$$

Explained intuitively, in the case of $q \in (0, 1)$ this means that as time increases, the number of fast moving particles will be split in half between the left and right chambers. Specifically, based on the random variable X_k, we can see that there will be roughly half of the fast particles in the left chamber as time increases. In the case of $q = 0$, there are no fast moving particles in the urn, so as time increases, the number of fast moving particles remains at 0. For $q = 1$, there are only fast moving particles in the urn, so as time increases, the number of fast moving particles approaches $\frac{n}{2}$.

Question 7

First, we construct the sample and state spaces for $(X, L) = \{X_k, L_k\}_{k\in\mathbb{N}}$:

$$\Omega = [0,1]^n$$
$$\mathbb{S} = \{(f,t) | t \geq f, 0 \leq f \leq nq \text{ for } f \in \mathbb{N}, t \in \mathbb{N}\}$$

Now, we want to show that (X, L) is a Markov Chain by finding its transition probabilities:

$$(X, L) \rightarrow \begin{cases} (x+1, l+1) \text{ w.p.} \frac{qn-x}{n} \\ (x, l) \text{ w.p.} \frac{(n-l)-(qn-x)}{n} \\ (x, l-1) \text{ w.p.} \frac{l-x}{n} \\ (x-1, l-1) \text{ w.p.} \frac{x}{n} \end{cases}$$

We can conclude that (X_k, L_k) is a Markov Chain because we can assign transition probabilities from $(x, l) \rightarrow z$ only dependent on the current state (X_k, L_k) and constants.

Question 8

We will start out by writing three initial equations (which we will use and manipulate in our computations to reach the stationary distribution).

$$\mu(x)p(x, x+1) = \mu(x+1)p(x+1, x) \tag{10}$$
$$\mu(x)p(x, x) = \mu(x+1)p(x, x) \tag{11}$$
$$\mu(x)p(x, x-1) = \mu(x-1)p(x-1, x) \tag{12}$$

We'll start by manipulating the equation (11):

$$\mu(x)p(x,x) = \binom{nq}{x}(1-q) = \mu(x)p(x,x) \tag{13}$$

Thus, $\mu(x)p(x,x) = \mu(x)p(x,x)$ for all $x \in \mathbb{S}$.

Next, let's look at equation (12):

$$\begin{aligned}
\mu(x)p(x,x-1) &= \binom{nq}{x}\frac{x}{n} && (14)\\
&= \frac{(nq)!}{(nq-x)!x!}\frac{x}{n} && (15)\\
&= \frac{(nq)!}{n(nq-x)!(x-1)!} && (16)
\end{aligned}$$

Next, let's look at equation (10):

$$\begin{aligned}
\mu(x)p(x,x+1) &= \binom{nq}{x}\frac{nq-x}{n} && (17)\\
&= \frac{(nq)!}{(nq-x)!x!} * \frac{nq-x}{n} && (18)\\
&= \frac{(nq)!}{n(nq-x-1)!x!} && (19)\\
& && (20)\\
\mu(x+1)p(x,x+1) &= \binom{nq}{x+1}\frac{x+1}{n} && (21)\\
&= \frac{(nq)!}{nq-x-1)!(x+1)!} * \frac{x+1}{n} && (22)\\
&= \frac{(nq)!}{n(nq-x-1)!x!} && (23)
\end{aligned}$$

Thus, $\mu(x)p(x,x+1) = \mu(x+1)p(x+1,x)$ for all $x \in \mathbb{S}$.

Finally, let's go back to the right side of equation (12):

$$\begin{aligned}
\mu(x-1)p(x-1,x) &= \binom{nq}{x-1}\frac{nq-x+1}{n} && (24)\\
&= \frac{(nq)!}{nq-x+1)!(x-1)!} * \frac{nq-x+1}{n} && (25)\\
&= \frac{(nq)!}{n(nq-x-1)!x!} && (26)
\end{aligned}$$

Thus, $\mu(x)p(x,x-1) = \mu(x-1)p(x-1,x)$ for all $x \in \mathbb{S}$.

Since $\mu_x = \binom{nq}{x}$ is a stationary measure for x,

$$\sum_{z \in \mathbb{S}} \mu(z) = \sum_{i=0}^{nq} \binom{nq}{x} = 2^{nq}$$

$$\text{Therefore, } \pi(x) = \frac{\binom{nq}{x}}{2^{nq}}.$$

Question 9

First, let's calculate the expectation of L_k using the tower property as we did in exercise 6.

$$\begin{aligned}
\mathbb{E}[L_{k+1}|L_k, X_k] &= \frac{qn - X_k}{n}(L_k + 1) + \frac{(n - L_k) - (qn - x)}{n} L_k + \frac{L_k - X_k}{n}(L_k \\
&= \frac{nL_k - L_k + nq - X_k}{n} \\
&= L_k(1 - \frac{1}{n}) - \frac{X_k}{n} + q
\end{aligned}$$

Now, applying the tower property, we get:

$$\begin{aligned}
\mathbb{E}[L_k] &= \mathbb{E}[\mathbb{E}[L_k|L_{k-1}, X_{k-1}]] \\
&= \mathbb{E}[L_{k-1}](1 - \frac{1}{n}) - \frac{1}{n}\mathbb{E}[X_{k-1}] + q \\
&= [\mathbb{E}[L_{k-2}](1 - \frac{1}{n}) - \frac{1}{n}\mathbb{E}[X_{k-2}] + q](1 - \frac{1}{n}) - \frac{1}{n}\mathbb{E}[X_{k-1}] + q \\
&= [\mathbb{E}[L_{k-2}](1 - \frac{1}{n})^2 - (1 - \frac{1}{n})(q - \frac{1}{n}\mathbb{E}[X_{k-2}]] + q - \frac{1}{n}\mathbb{E}[X_{k-1}] + q \\
&= \mathbb{E}[L_{k-2}](1 - \frac{1}{n})^2 - (1 - \frac{1}{n})(q - \frac{1}{n}\mathbb{E}[X_{k-2}] + q - \frac{1}{n}\mathbb{E}[X_{k-1}] + q \\
&= \mathbb{E}[L_{k_0}](1 - \frac{1}{n})^k - \sum_{i=0}^{k-1}(1 - \frac{1}{n})^i(q - \frac{1}{n}\mathbb{E}[X_{k-1-i}]
\end{aligned}$$

Therefore,

$$\mathbb{E}[X_k] = (1 - \frac{2}{n})^k z + \frac{qn}{2}(1 - (1 - \frac{2}{n})^k)$$

and further

$$\mathbb{E}[X_{k-1-i}] = (1 - \frac{2}{n})^{k-1-i} z + \frac{qn}{2}(1 - (1 - \frac{2}{n})^{k-1-i})$$

Let $A = (1 - \frac{2}{n})^{k-1-i}$. Then we have,

$$\frac{1}{n}\mathbb{E}[X_{k-1-i}] = \frac{z}{n}A + \frac{q}{2}(1-A) \quad (36)$$
$$= \frac{q}{2} + A(\frac{z}{n} - \frac{q}{2}) \quad (37)$$

and then (Latex isn't working so we uploaded our handwritten calculations on the last page)

[Math Processing Error]
Finally,

$$\mathbb{E}[L_k] = \mathbb{E}[L_0](1-\frac{1}{n})^k + \frac{nq}{2}(1-(1-\frac{1}{n})^k) - (\frac{z}{n} - \frac{q}{2})(n-1)[(1-(1-\frac{1}{n})^{k-1}$$

and in terms of the limit as k approaches infinity:

$$lim_{k\to\inf}\mathbb{E}[L_k] = \mathbb{E}[L_0](0) + \frac{nq}{2}(1-0) - (\frac{z}{n} - \frac{q}{2})(n-1)(0-0) \quad (42)$$
$$= \frac{nq}{2} \quad (43)$$

Our stationary distribution can then be expressed as:

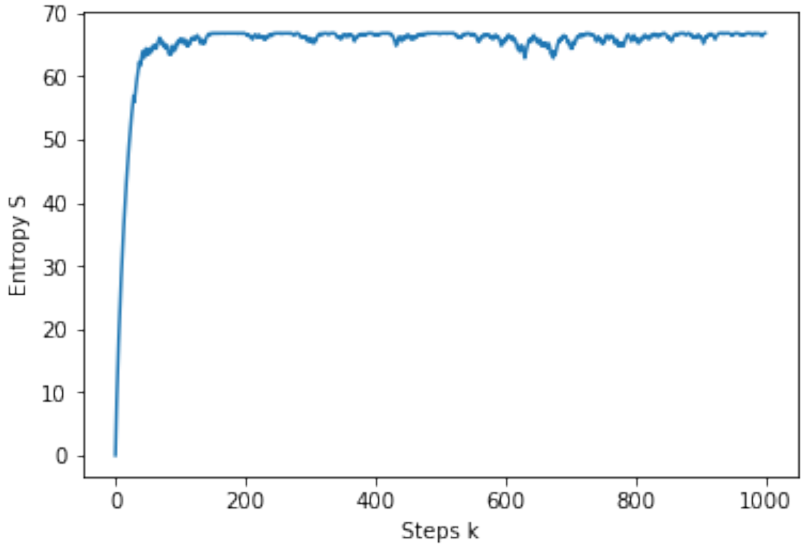
$$\pi(x,l) = \begin{cases} 0 \text{ for } x \neq l \\ \binom{n}{x} \text{ for } x = l \end{cases}$$

Question 10

In [7]:
```
import random
import numpy as np
import matplotlib.pyplot as plt
import math
```

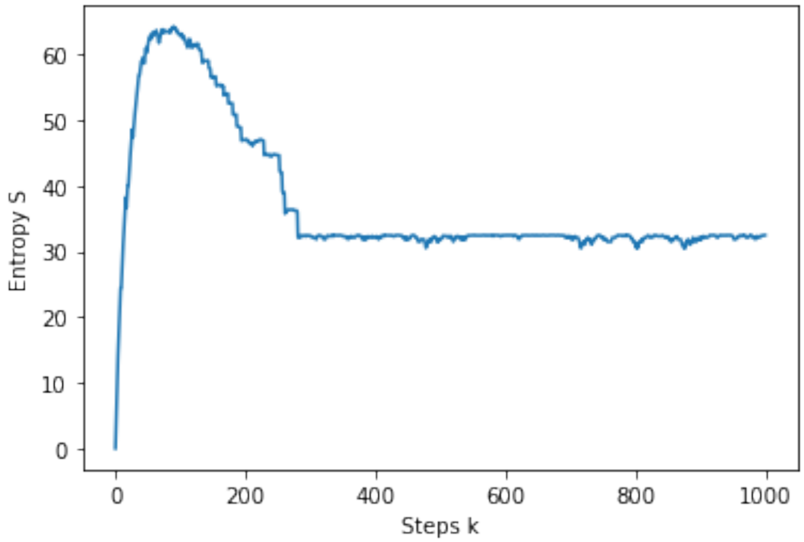
n = 100, q = 1

In [8]:
```
# n=100, q=1
n = 100
q = 1
n_fast = q*n
n_slow = (1-q)*n
steps = 1000   #total steps = 10^3
X = np.zeros(steps+1)
Y = np.zeros(steps+1)
S = np.zeros(steps+1)

## simulation of the process
## label p[0->n_fast] being fast particles, p[n_fast+1->n] being slow partic
## the value of p[i] being 0 (in left urn) or 1 (right urn)
## initially,  all n particales are in left urn

p = np.zeros(n) #initial state
X[0] = np.count_nonzero(p[:n_fast]==0)
Y[0] = np.count_nonzero(p[n_fast:n]==0)
S[0] = math.log(math.comb(n_fast, int(X[0]))) + math.log(math.comb(n_slow, i

## simulation: at each step, uniform randomly pick a particle. First, we che
## Next, we check if it is in right urn (value = 1); Finally, to determine i
## X_k = # of 0's in p[0->n_fast]
## Y_k = # of 0's in p[n_fast+1->n]

for k in range(steps):
    a = random.randint(1, n)
    if a <= n_fast:
        p[a-1] = 1 - p[a-1] # change from 0 to 1 or 1 to 0
    elif p[a-1] == 0:
        p[a-1] = 1 - p[a-1] # change from 0 to 1 or 1 to 0
    # update X_k and Y_k
    X[k+1] = np.count_nonzero(p[:n_fast]==0)
    Y[k+1] = np.count_nonzero(p[n_fast:n]==0)
    S[k+1] = math.log(math.comb(n_fast, int(X[k+1]))) + math.log(math.comb(n


plt.plot(range(steps+1), S)
plt.xlabel('Steps k')
plt.ylabel('Entropy S')
plt.show()
```



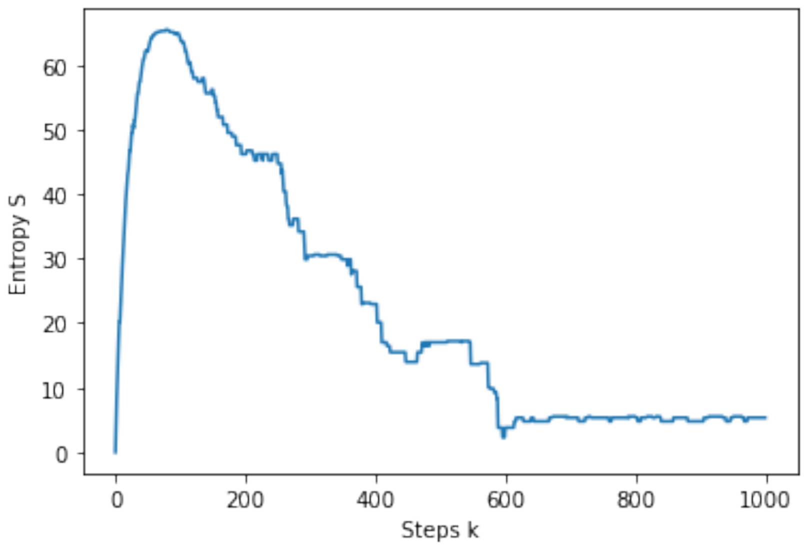
$n = 100, q = .5$

In [9]:
```
# n=100, q=.5
n = 100
q = .5
n_fast = int(q*n)
n_slow = int((1-q)*n)
steps = 1000   #total steps = 10^3
X = np.zeros(steps+1)
Y = np.zeros(steps+1)
S = np.zeros(steps+1)

## simulation of the process
## label p[0->n_fast] being fast particles, p[n_fast+1->n] being slow partic
## the value of p[i] being 0 (in left urn) or 1 (right urn)
## initially,  all n particales are in left urn

p = np.zeros(n) #initial state
X[0] = np.count_nonzero(p[:n_fast]==0)
Y[0] = np.count_nonzero(p[n_fast:n]==0)
S[0] = math.log(math.comb(n_fast, int(X[0]))) + math.log(math.comb(n_slow, i

## simulation: at each step, uniform randomly pick a particle. First, we che
## Next, we check if it is in right urn (value = 1); Finally, to determine i
## X_k = # of 0's in p[0->n_fast]
## Y_k = # of 0's in p[n_fast+1->n]

for k in range(steps):
    a = random.randint(1, n)
    if a <= n_fast:
        p[a-1] = 1 - p[a-1] # change from 0 to 1 or 1 to 0
    elif p[a-1] == 0:
        p[a-1] = 1 - p[a-1] # change from 0 to 1 or 1 to 0
    # update X_k and Y_k
    X[k+1] = np.count_nonzero(p[:n_fast]==0)
    Y[k+1] = np.count_nonzero(p[n_fast:n]==0)
    S[k+1] = math.log(math.comb(n_fast, int(X[k+1]))) + math.log(math.comb(n


plt.plot(range(steps+1), S)
plt.xlabel('Steps k')
plt.ylabel('Entropy S')
plt.show()
```



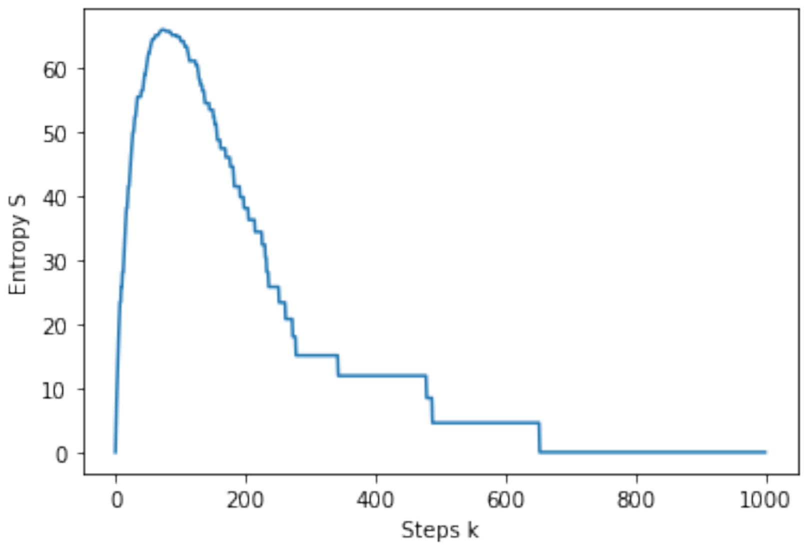
n = 100, q = .1

In [10]:
```
# n=100, q=.1
n = 100
q = .1
n_fast = int(q*n)
n_slow = int((1-q)*n)
steps = 1000   #total steps = 10^3
X = np.zeros(steps+1)
Y = np.zeros(steps+1)
S = np.zeros(steps+1)

## simulation of the process
## label p[0->n_fast] being fast particles, p[n_fast+1->n] being slow partic
## the value of p[i] being 0 (in left urn) or 1 (right urn)
## initially,  all n particales are in left urn

p = np.zeros(n) #initial state
X[0] = np.count_nonzero(p[:n_fast]==0)
Y[0] = np.count_nonzero(p[n_fast:n]==0)
S[0] = math.log(math.comb(n_fast, int(X[0]))) + math.log(math.comb(n_slow, i

## simulation: at each step, uniform randomly pick a particle. First, we che
## Next, we check if it is in right urn (value = 1); Finally, to determine i
## X_k = # of 0's in p[0->n_fast]
## Y_k = # of 0's in p[n_fast+1->n]

for k in range(steps):
    a = random.randint(1, n)
    if a <= n_fast:
        p[a-1] = 1 - p[a-1] # change from 0 to 1 or 1 to 0
    elif p[a-1] == 0:
        p[a-1] = 1 - p[a-1] # change from 0 to 1 or 1 to 0
    # update X_k and Y_k
    X[k+1] = np.count_nonzero(p[:n_fast]==0)
    Y[k+1] = np.count_nonzero(p[n_fast:n]==0)
    S[k+1] = math.log(math.comb(n_fast, int(X[k+1]))) + math.log(math.comb(n


plt.plot(range(steps+1), S)
plt.xlabel('Steps k')
plt.ylabel('Entropy S')
plt.show()
```



n = 100, q = .01

In [11]:
```
# n=100, q=.01
n = 100
q = .01
n_fast = int(q*n)
n_slow = int((1-q)*n)
steps = 1000   #total steps = 10^3
X = np.zeros(steps+1)
Y = np.zeros(steps+1)
S = np.zeros(steps+1)

## simulation of the process
## label p[0->n_fast] being fast particles, p[n_fast+1->n] being slow partic
## the value of p[i] being 0 (in left urn) or 1 (right urn)
## initially,  all n particales are in left urn

p = np.zeros(n) #initial state
X[0] = np.count_nonzero(p[:n_fast]==0)
Y[0] = np.count_nonzero(p[n_fast:n]==0)
S[0] = math.log(math.comb(n_fast, int(X[0]))) + math.log(math.comb(n_slow, i

## simulation: at each step, uniform randomly pick a particle. First, we che
## Next, we check if it is in right urn (value = 1); Finally, to determine i
## X_k = # of 0's in p[0->n_fast]
## Y_k = # of 0's in p[n_fast+1->n]

for k in range(steps):
    a = random.randint(1, n)
    if a <= n_fast:
        p[a-1] = 1 - p[a-1] # change from 0 to 1 or 1 to 0
    elif p[a-1] == 0:
        p[a-1] = 1 - p[a-1] # change from 0 to 1 or 1 to 0
    # update X_k and Y_k
    X[k+1] = np.count_nonzero(p[:n_fast]==0)
    Y[k+1] = np.count_nonzero(p[n_fast:n]==0)
    S[k+1] = math.log(math.comb(n_fast, int(X[k+1]))) + math.log(math.comb(n


plt.plot(range(steps+1), S)
plt.xlabel('Steps k')
plt.ylabel('Entropy S')
plt.show()
```



We find that if all particles are fast particles, i.e. $q = 1$, we find that the entropy will be monontically increasing, that means the system will be increasingly disorder. When there is a portion of slow paticles with Maxwell's demon, i.e. $q = 0.5, 0.1, 0.01$, we find that entropy will first increase and then decrease. Finally, it will approach to some fixed non-negative value. This is due to the portion of slow particles and existences of Maxwell's demon. If we have a larger portion of slow particles, then the system will have a lower entropy (order) eventually. For example, when $q = 0.01$, entropy will eventually approach to 0 for about 400 steps.