1. The key assumpeion underlying Markov Chain is that the transition probabolity p_{ij} apply whenever state i is visited, no matter what happened in the past, and no matter how state i was reached. Mathematically, we have.

$$P(X_{n+1}=j \mid X_n=i, X_{n-1}=i_{n-1}, \dots, X_0=i_0) = P(X_{n+1}=j \mid X_n=i) = p_{ij}$$

Focusing on this problem. we have.

$$P(X_{n+1}=R \mid X_n=L, X_{n-1}=R) = 1/2$$

$$P(X_{n+1}=R \mid X_n=L, X_{n-1}=L, X_{n-2}=R) = 0$$

We can see that the value of state n not only depends on X_n, so the generated sequence of signals L and R $\boxed{\text{can't}}$ be describe as a Markov Chain with states L and R.

2. Solution:

(a) Let interger n denotes the initial instance.

According to the Question, the spider captures the fly only when they are in the same position, instead of the meet each other (in case spider $\rightleftarrows$ fly, the spider won't capture the fly), so we have.

$p_{00}=1$, $p_{0i}=0$ (for $i\neq 0$). $p_{10}=0.4$. $p_{11}=0.3+0.3=0.6$. $p_{1i}=0$ (for $i\neq 0,1$)

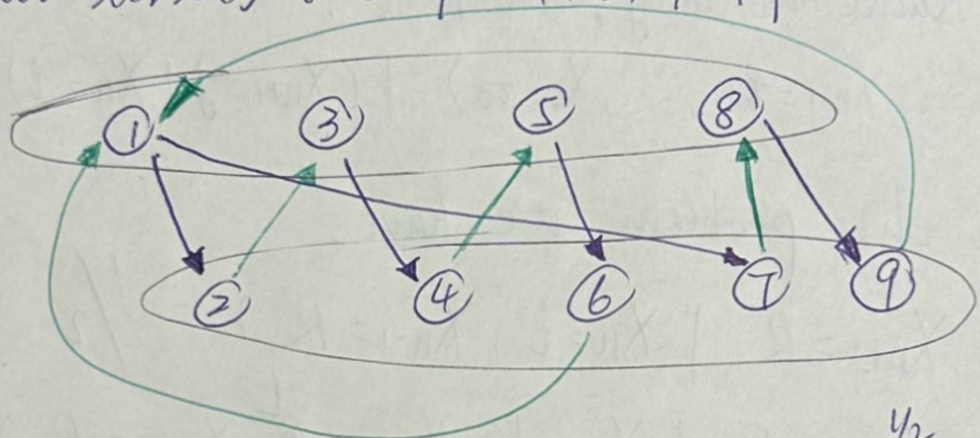
for all $i\neq 0,1$, we have $p_{ii}=0.3$. $p_{i(i-1)}=0.4$. $p_{i(i-2)}=0.3$

(b) state 0 is recurrent, others are all transient.

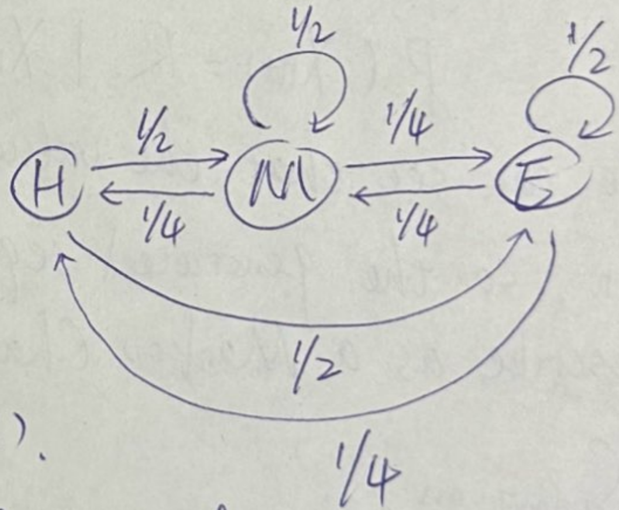
3. Solution:

It is periodic with period 2.

The two subsets are $\{2, 4, 6, 7, 9\}$ and $\{1, 3, 5, 8\}$



4. Solution:



We can model the Markov Chain with 3 states H (hard), M (medium), E (easy).

It's easy to find that this Markov Chain is aperiodic with a single recurrent class. According to the balance equation

$$\left.\begin{aligned} \pi_H &= \frac{1}{4}\pi_M + \frac{1}{4}\pi_E \\ \pi_M &= \frac{1}{2}\pi_H + \frac{1}{2}\pi_M + \frac{1}{2}\pi_E \\ \pi_E &= \frac{1}{2}\pi_H + \frac{1}{4}\pi_M + \frac{1}{2}\pi_E \end{aligned}\right.$$

And according to the normalization equation

$$\left.\pi_H + \pi_M + \pi_E = 1 \right\} \Rightarrow \pi_H = \frac{1}{5}, \quad \pi_M = \frac{2}{5}, \quad \pi_E = \frac{2}{5}$$

5. Solution:

(a) According to the local balance equation, we have.

$$\begin{cases} 0.6\pi_1 = 0.3\pi_2 \\ 0.2\pi_2 = 0.2\pi_3 \end{cases} \Longrightarrow \pi_2 = \pi_3 = 2\pi_1$$

According to the Normalization equation, we have

$$\pi_1 + \pi_2 + \pi_3 = 1 \Longrightarrow \boxed{\pi_1 = \frac{1}{5}, \ \pi_2 = \pi_3 = \frac{2}{5}}$$

(b) Transitions that result in a state with higher index are birth(s). which means only ① → ② and ② → ③ are birth(s). So the probability that the first transition to observe is a birth is

$$P(X_{n+1}=2 \mid X_n=1) + P(X_{n+1}=3 \mid X_n=2) = 0.6\pi_1 + 0.2\pi_2 = \boxed{\frac{1}{5}}$$

(since the Markov Chain is already in steady-state, $n \to \infty$)

(c)

Current state	Next state changes	
	Birth	Death
1	$P_{1B} = 1$	0
2	$P_{2B} = \frac{0.2}{0.2+0.3} = \frac{2}{5}$	$\frac{0.3}{0.2+0.3}$
3	$P_{3B} = 0$	1

So the probability that the first change of state to observe is a birth is

$$P_{1B}\pi_1 + P_{2B}\pi_2 + P_{3B}\pi_3$$

$$= \pi_1 + \frac{2}{5}\pi_2 = \frac{1}{5} + \frac{2}{5} \times \frac{2}{5} = \boxed{\frac{9}{25}}$$

(d)

P(state was 2 | first transition is a birth)

$$= \frac{P(\text{state was 2 and first transition is a birth})}{P(\text{first transition is a birth})}$$

$$= \frac{\pi_2 \cdot 1/5}{1/5} = \pi_2 = \boxed{\frac{2}{5}}$$

(e) According to (c), we ~~have~~ obtain the probability that the state is 2 and the first change of state is a birth is

$P_{2B}\pi_2 = \frac{2}{5}\pi_2 = \frac{4}{25}$

$\therefore$ The desired probability is $\frac{4/25}{9/25} = \boxed{\frac{4}{9}}$

(f) $P(\text{birth} \mid \text{state changes}) = \dfrac{\cancel{1/5}\ P(\text{first observe transition is a birth})}{\pi_1 \cdot P(X_{n+1} \neq 1 \mid \cancel{\varphi} X_n = 1) + \pi_2 \cdot P(X_{n+1} \neq 2 \mid X_n = 2) + \pi_3 \cdot P(X_{n+1} \neq 3 \mid X_n = 3)}$

$= \dfrac{1/5}{1/5 \times 0.6 + 2/5 \times (1-0.5) + 2/5 \times (1-0.8)} = \boxed{\frac{1}{2}}$

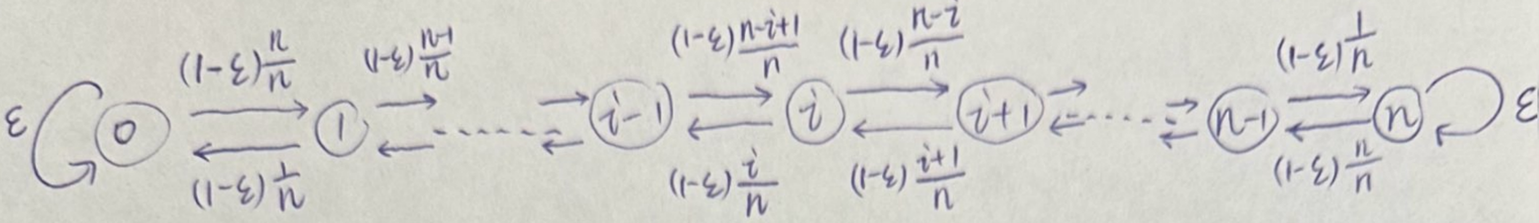
(g) The desired probability is

$\dfrac{P(\text{observed transition leads to state 2})}{P(\text{state changes})} = \dfrac{\pi_1 \times 0.6 + \pi_3 \times (1-0.8)}{\pi_1 \times 0.6 + \pi_2 \times (1-0.5) + \pi_3 \times (1-0.8)}$

$= \dfrac{0.6\pi_1 + 0.2\pi_3}{0.6\pi_1 + 0.5\pi_2 + 0.2\pi_3}$

$= \boxed{\frac{1}{2}}$

Problem 6. Solution:



This ~~is~~ is a Markov Chain with a single recurrent ~~state~~ class, and it is aperiodic. It is a birth-death process.

According to the local balance equation, we have

$$\pi_i (1-\varepsilon)\frac{n-i}{n} = \pi_{i+1}(1-\varepsilon)\frac{i+1}{n} \quad (\text{for } i = 0, 1, \cdots, n-1)$$

$$\Longrightarrow \pi_{i+1} = \frac{n-i}{i+1}\pi_i \Longrightarrow \pi_i = \frac{n-i+1}{i}\pi_{i-1}$$

$$\Longrightarrow \pi_i = \frac{n-i+1}{i}\pi_{i-1} = \frac{n-i+1}{i}\cdot\frac{n-i+2}{i-1}\pi_{i-2} = \cdots\cdots = \frac{(n-i+1)(n-i+2)\cdots(n-1)n}{i(i-1)\cdots 1}\pi$$

$$= \frac{n!}{i!\,(n-i)!}\pi_0 = \binom{n}{i}\pi_0$$

According to the normalization equation, we have.

$$\sum_{i=0}^{n}\pi_i = \pi_0 + \pi_1 + \cdots + \pi_{n-1} + \pi_n = 1$$

$$\Longrightarrow \pi_0 + \binom{n}{1}\pi_0 + \cdots + \binom{n}{i}\pi_0 + \cdots\cdots + \binom{n}{n-1}\pi_0 + \binom{n}{n}\pi_0 = 1$$

$$\Longrightarrow \left[\sum_{i=0}^{n}\binom{n}{i}\right]\cdot\pi_0 = 1$$

$$\Longrightarrow 2^n\cdot\pi_0 = 1$$

$$\Longrightarrow \pi_0 = \frac{1}{2^n}$$

$$\boxed{\begin{aligned}(1+1)^n &= \binom{n}{0}+\binom{n}{1}+\cdots\binom{n}{i}+\cdots\binom{n}{n}\\ &= \sum_{i=0}^{n}\binom{n}{i}\\ \therefore \sum_{i=0}^{n}\binom{n}{i} &= (1+1)^n = 2^n\end{aligned}}$$

thus $\pi_i = \binom{n}{i}\cdot\frac{1}{2^n}$ (for $i = 0, 1, \cdots n$)

Problem 7. Solution:

Let i denotes the number of white balls in the first urn

$$p_{i,i-1} = \left(\frac{i}{m}\right)^2, \quad p_{i,i+1} = \left(\frac{m-i}{m}\right)^2, \quad p_{ii} = \frac{i(m-i)}{m^2} + \frac{(m-i)i}{m^2} = \frac{2i(m-i)}{m^2}$$

This is a birth-death process.

The Markov Chain is aperiodic and has a single recurrent class.

So according to the local balance equation, we have

$$\pi_{i-1}\left(\frac{m-i+1}{m}\right)^2 = \pi_i\left(\frac{i}{m}\right)^2 \quad (\text{for } i = 1, 2, \cdots m)$$

$$\Rightarrow \pi_i = \left(\frac{m-i+1}{i}\right)^2 \pi_{i-1} = \left(\frac{(m-i+1)(m-i+2)}{i(i-1)}\right)^2 \pi_{i-2} = \cdots \cancel{m(m-1)}$$

$$= \left[\frac{(m-i+1)(m-i+2)\cdots\cdots(m-1)m}{i(i-1)\cdots\cdots 2\times 1}\right]^2 \pi_0 = \left(\frac{m!}{i!\,(m-i)!}\right)^2 \pi_0$$

$$= \binom{m}{i}^2 \pi_0 \qquad (\text{similar to Problem 6})$$

According to the normolization equation

$$\sum_{i=0}^{m} \pi_i = \pi_0 + \pi_1 + \cdots \pi_m = \left[\sum_{i=0}^{m}\binom{m}{i}^2\right] \cdot \pi_0 = 1$$

$$\Rightarrow \binom{2m}{m}\pi_0 = 1 \Rightarrow \pi_0 = \frac{1}{\binom{2m}{m}}$$

$(x+y)^n \cdot (x+y)^n = (x+y)^{2n}$

coefficient of $x^{n+k}y^{n-k}$ is $\sum_{i=0}^{n}\binom{n}{i}\binom{n}{n+k-i}$

let $k=0 \Rightarrow \sum_{i=0}^{n}\binom{n}{i}\binom{n}{n-i} = \sum_{i=0}^{n}\binom{n}{i}^2$

$(x^n y^n)$

$\Rightarrow \sum_{i=0}^{n}\binom{n}{i}^2 = \binom{2n}{n}$

Thus $$\pi_i = \frac{\binom{m}{i}^2}{\binom{2m}{m}} \quad (\text{for } i = 0, 1, \cdots m)$$

Problem 8. Solution:

(a) State 4 and 5 are transient

State 1, 2, 3, 6, 7 are recurrent.

The recurrent class $\{1,2,3\}$ is aperiodic

The recurrent class $\{6,7\}$ is periodic

(b) If the process starts from state 1, it will stay in the recurrent class $\{1,2,3\}$. According to the local balance equation, we have
$$\begin{cases} 0.5\pi_1 = 0.5\pi_2 \\ 0.1\pi_2 = 0.6\pi_3 \end{cases}$$

According to the normalization equation, we have $\pi_1+\pi_2+\pi_3=1$

$$\Rightarrow \boxed{\pi_1=\pi_2=\frac{6}{13} \quad \pi_3=\frac{1}{13}}$$

(c) ~~(3)~~ The recurrent class $\{6,7\}$ is periodic, so there are $\boxed{\text{no}}$ state-steady probabilities. $r_{66}(n) = \begin{cases} 0, & n \text{ is odd} \\ 1, & n \text{ is even} \end{cases}$

(d) (i) The probability we desired is $p_{12}\pi_1 + p_{23}\pi_2 = 0.5\times\frac{6}{13}+0.1\times\frac{6}{13} = \boxed{\frac{18}{65}}$

(ii) The desired probability is $\frac{p_{23}\pi_2}{p_{12}\pi_1+p_{23}\pi_2} = \frac{0.1\times\frac{6}{13}}{0.5\times\frac{6}{13}+0.1\times\frac{6}{13}} = \boxed{\frac{1}{6}}$

(iii)

prob	Current State	State changes: increase	State changes: decrease
$\pi_1 = 6/13$	1	$p_{1I} = \frac{0.5}{(1-0.5)} = 1$	0
$\pi_2 = 6/13$	2	$p_{2I} = \frac{0.1}{(1-0.4)} = \frac{1}{6}$	$\frac{0.5}{(1-0.4)} = \frac{5}{6}$
$\pi_3 = 1/13$	3	$p_{3I} = 0$	$\frac{0.6}{(1-0.4)} = 1$

Thus the desired probability is

$$p_{1I}\pi_1 + p_{2I}\pi_2 + p_{3I}\pi_3 = 1\times\frac{6}{13}+\frac{1}{6}\times\frac{6}{13}+0\times\frac{1}{13} = \boxed{\frac{7}{13}}$$

(e) (i) $a_4 = \sum_{j=1}^{m} p_{4j} a_j = p_{43} a_3 + p_{44} a_4 + p_{45} a_5 = 0.2 \times 1 + 0.4 a_4 + 0.2 a_5$

$a_5 = \sum_{j=1}^{m} p_{5j} a_j = p_{54} a_4 + p_{57} a_7 = 0.7 a_4 + 0.3 \times 0 = 0.7 a_4$

Let a_4, a_5 be the probability the the recurrent class $\{1,2,3\}$ is finally reached starting from state 4, 5, respectively

$\Rightarrow a_4 = 0.2 + 0.4 a_4 + 0.14 a_4 \Rightarrow \begin{cases} a_4 = \frac{10}{23} \\ a_5 = \frac{7}{23} \end{cases}$

$\therefore$ Starting from state 4,

P(.... eventually reach class $\{1,2,3\}$) $= a_4 = \boxed{\frac{10}{23}}$

P(.... $\{6,7\}$) $= 1 - a_4 = \boxed{\frac{13}{23}}$

(ii) $\mu_4 = 1 + \sum_{j=1}^{m} p_{ij} \mu_j = 1 + p_{43}\mu_3 + p_{44}\mu_4 + p_{45}\mu_5 = 1 + 0.2 \times 0 + 0.4\mu_4 + 0.2\mu_5$

$= 1 + 0.4\mu_4 + 0.2\mu_5$

$\mu_5 = 1 + \sum_{j=1}^{m} p_{ij} \mu_j = 1 + p_{54}\mu_4 + p_{57}\mu_7 = 1 + 0.7\mu_4 + 0.3 \times 0$

$= 1 + 0.7\mu_4$

$\begin{cases} \mu_4 = 1 + 0.4\mu_4 + 0.2\mu_5 \\ \mu_5 = 1 + 0.7\mu_4 \end{cases} \Rightarrow \begin{cases} \mu_4 = \frac{60}{23} \\ \mu_5 = \frac{65}{23} \end{cases}$

$\therefore$ The expected number of transitions until a recurrent state is reached starting from state 4 is $\mu_4 = \boxed{\frac{60}{23}}$

Problem 9 Solution:

We will model this system using a continuous Marcov Chain with state ~~X(t)~~ n equal to the numbers of people waiting at time t, $n=0,1,2,3,4$. For. $n=0,1,2,3$

the transitions from n to $n+1$ have rate 1

the transitions from n to $n-1$ have rate 2

~~This is~~ The local balance equations is $\pi_{n-1}=2\pi_n$

So we have $\pi_n=\frac{1}{2}\pi_{n-1}=(\frac{1}{2})^n\pi_0$ (for $n=1,2,3,4$)

Normalization equation: $\sum_{i=0}^{n}\pi_i=\pi_0(1+\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\frac{1}{2^4})=1$

So we have $\pi_0=\frac{1}{1+\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\frac{1}{2^4}}=\frac{1}{31/16}=\frac{16}{31}$

Then the expected number of passengers found by Penelope (Given she joins the queue)

is $E(N)=\frac{0\times\pi_0+1\times\pi_1+2\times\pi_2+3\times\pi_3}{\pi_0+\pi_1+\pi_2+\pi_3}$

$$=\frac{1\times 16/31\times 1/2+2\times\frac{16}{31}\times 1/4+3\times 16/31\times\frac{1}{8}}{16/31\,(1+1/2+1/4+1/8)}=\frac{11}{15}$$

Since the empty taxis pass by at a Poison rate 2/min, the expected waiting time for a new empty taxi is $\frac{1}{\lambda}=\frac{1}{2}$ min

$\therefore$ The desired expected waiting time is

$$E(T)=\frac{1}{\lambda}E(N)=\frac{1}{2}\times\frac{11}{15}=\frac{11}{30}\text{ min}$$

Problem 10. Solution:

We model this problem using a Markov chain with states $i = 0, 1, 2, 3$ denotes the numbers of umbrellas ~~at~~ in the current location.

Then the transition probability matrix is

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1-p & p \\ 0 & 1-p & p & 0 \\ 1-p & p & 0 & 0 \end{bmatrix}$$

$p_{03} = 1 \quad p_{30} = 1-p \quad p_{31} = p$

$p_{12} = 1-p \quad p_{13} = p \quad p_{22} = p$

$p_{21} = 1-p$

The chain

$$\begin{tikzcd} 0 \arrow[r, shift left, "1"] & 3 \arrow[l, shift left, "1-p"] \arrow[r, shift left, "p"] & 1 \arrow[l, shift left, "p"] \arrow[r, shift left, "1-p"] & 2 \arrow[l, shift left, "1-p"] \arrow[loop right, "p"] \end{tikzcd}$$

has a single recurrent class that is aperiodic

Balance equation:

$$\left.\begin{aligned} \pi_0 &= (1-p)\pi_3 \\ \pi_1 &= p\pi_3 + (1-p)\pi_2 \\ \pi_2 &= (1-p)\pi_1 + p\pi_2 \\ \pi_3 &= \pi_0 + p\pi_1 \end{aligned}\right.$$

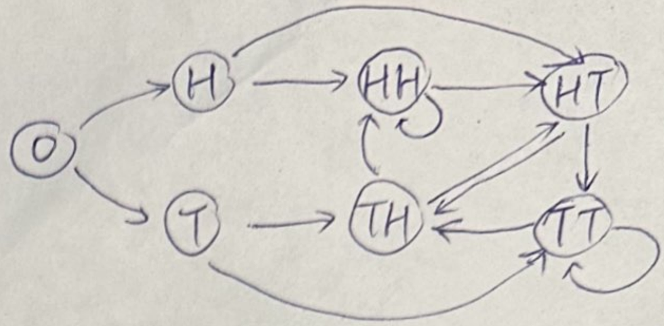
Normalization equation: $\pi_0 + \pi_1 + \pi_2 + \pi_3 = 1$

$$\pi_0 = \frac{1-p}{4-p} \quad \pi_1 = 1/(4-p) \quad \pi_2 = 1/(4-p) \quad \pi_3 = \frac{1}{4-p}$$

According to the steady-state convergence theorem.

$$P(\text{get wet}) = \pi_0 p = \boxed{\frac{(1-p)p}{4-p}}$$

Problem 11 Solution:



(All arcs are with prob $p_{ij} = \frac{1}{2}$)

(a) $s = HT$, $\therefore t_{HT} = 0$

$$
\left.\begin{aligned}
t_0 &= 1 + \tfrac{1}{2}t_H + \tfrac{1}{2}t_T \\
t_H &= 1 + \tfrac{1}{2}t_{HH} + \tfrac{1}{2}t_{HT} \\
t_T &= 1 + \tfrac{1}{2}t_{TH} + \tfrac{1}{2}t_{TT} \\
t_{HH} &= 1 + \tfrac{1}{2}t_{HH} + \tfrac{1}{2}t_{HT} \\
t_{TH} &= 1 + \tfrac{1}{2}t_{HH} + \tfrac{1}{2}t_{HT} \\
t_{TT} &= 1 + \tfrac{1}{2}t_{TH} + \tfrac{1}{2}t_{TT}
\end{aligned}\right\} \Rightarrow
\begin{aligned}
&\boxed{t_0 = 4} \\
&t_H = 2 \\
&t_T = 4 \\
&t_{HH} = 2 \\
&t_{TH} = 2 \\
&t_{TT} = 4
\end{aligned}
$$

$\therefore E($~~exp~~ time to see $HT) = 4$

$(E(\text{time to see } HT) = 2^2 = 4)$

(b) $s = HH$, $\therefore t_{HH} = 0$

$$
\left.\begin{aligned}
t_0 &= 1 + \tfrac{1}{2}t_H + \tfrac{1}{2}t_T \\
t_H &= 1 + \tfrac{1}{2}t_{HH} + \tfrac{1}{2}t_{HT} \\
t_T &= 1 + \tfrac{1}{2}t_{TH} + \tfrac{1}{2}t_{TT} \\
t_{HT} &= 1 + \tfrac{1}{2}t_{TH} + \tfrac{1}{2}t_{TT} \\
t_{TH} &= 1 + \tfrac{1}{2}t_{HH} + \tfrac{1}{2}t_{HT} \\
t_{TT} &= 1 + \tfrac{1}{2}t_{TH} + \tfrac{1}{2}t_{TT}
\end{aligned}\right\}
\begin{aligned}
&\boxed{t_0 = 6} \\
&t_H = 4 \\
&t_T = 6 \\
&\cancel{t_{HH}}\ t_{HT} = 6 \\
&t_{TH} = 4 \\
&t_{TT} = 6
\end{aligned}
$$

$\therefore E(\text{time to see } HH) = t_0 = 6$

$0 \to H \to HH$

$(E(\text{time to see } HH) = 2^2 + 2 = 6)$

(c) $E(\text{time to see } HTHT) = 2^4 + 2^2 = 20$

$0 \to HT \to HTHT$

$0 \to H \to HH \to HHH \to HHHH$

(d) $E(\text{time to see } HHHH) = 2^4 + 2^3 + 2^2 + 2^1 = 30$

1 Code and Result of Problem 11

Here is a Python program for a fair coin tossed:

1.1 Import Necessary Library

```
import numpy as np
from scipy import stats
```

1.2 Function Definition

1.2.1 run(p,n)

- returns one simulated value of time to see HH...H (n H, denoted by W_H,n)

```
def run(p, n):
    """Returns one simulated value of W_H,n
    in i.i.d. Bernoulli (p) trials"""

    tosses = 0                    # Number of tosses
    in_a_row = 0                  # Number of consecutive heads observed

    while in_a_row < n:           # While fewer than n consecutive heads
        tosses += 1              # update tosses
        if stats.bernoulli.rvs(p, size=1).item(0) == 1:
            in_a_row +=1                # update in_a_row
        else:
            in_a_row = 0                # reset in_a_row

    return tosses
```

1.2.2 simulate_run(p, n, repetitions)

- Returns an array of length equal to repetitions, whose entries are independent simulated values of W_H,n in i.i.d. Bernoulli (p) trials

```
def simulate_run(p, n, repetitions):
    """Returns an array of length equal to repetitions,
    whose entries are independent simulated values of W_H,n
    in i.i.d. Bernoulli (p) trials"""
    results = []
    i=0
    while i < repetitions:
        i+=1
        results.append(run(p,n))
    return results
```

1.2.3 HT_run(p)

- Returns one simulated value of W_HT in i.i.d. Bernoulli (p) trials

```python
def HT_run(p):
    """Returns one simulated value of W_HT
    in i.i.d. Bernoulli (p) trials"""

    Heads = 0
    Tails = 0
    tosses = 0
    while Tails == 0:  # While no Tails has been observed after a Heads
        while Heads == 0:
            tosses += 1
            if stats.bernoulli.rvs(p, size=1).item(0) == 1:
                Heads += 1       #Got a heads, break out of the heads loop
        tosses += 1

        if stats.bernoulli.rvs(p, size=1).item(0) == 0:
            Tails +=1
    return tosses
```

1.2.4 simulate_HT(p, N)

- Returns an array of length equal to repetitions, whose entries are independent simulated values of W_HT in i.i.d. Bernoulli (p) trials

```python
def simulate_HT(p, N):
    """Returns an array of length equal to repetitions,
    whose entries are independent simulated values of W_HT
    in i.i.d. Bernoulli (p) trials"""
    results = []
    i = 0
    while i< N:
        i += 1
        results.append(HT_run(p))
    return results
```

1.2.5 HTHT_run(p)

- Returns one simulated value of W_HTHT in i.i.d. Bernoulli (p) trials

```python
def HTHT_run(p):
    """Returns one simulated value of W_HTHT
    in i.i.d. Bernoulli (p) trials"""

    Heads = 0
    Tails = 0
    tosses = 0
    while Tails < 2 :  # While no Tails has been observed after a Heads
        while Heads == 0 and Tails == 0 :
            tosses += 1
            if stats.bernoulli.rvs(p, size=1).item(0) == 1:
                Heads += 1       #Got a heads (H), break out of the heads loop

        while Heads == 1 and Tails == 0 :
            tosses += 1
            if stats.bernoulli.rvs(p, size=1).item(0) == 0:
                Tails +=1 # Got T (HT), break out of the tails loop

        while Heads == 1 and Tails == 1:
            tosses += 1
            if stats.bernoulli.rvs(p, size=1).item(0) == 1:
                Heads += 1       #Got a heads (HTH), break out of the heads loop
            else:
                Heads = 0
                Tails = 0

        while Heads == 2 and Tails == 1:
            tosses += 1
            if stats.bernoulli.rvs(p, size=1).item(0) == 0:
                Tails += 1       #Got a T (HTHT), break out of the whole loop
            else:
                Heads = 1
                Tails = 0
    return tosses
```

1.2.6 simulate_HTHT(p, N)

- Returns an array of length equal to repetitions, whose entries are independent simulated values of W_HTHT in i.i.d. Bernoulli (p) trials

```python
def simulate_HTHT(p, N):
    """Returns an array of length equal to repetitions,
    whose entries are independent simulated values of W_HTHT
    in i.i.d. Bernoulli (p) trials"""
    results = []
    i = 0
    while i< N:
        i += 1
        results.append(HTHT_run(p))
    return results
```

1.3 Testing Code (100 repetitions)

- calculates the mean value of 100 repetitions and prints them on the screen

```
sim_W_HH = simulate_run(0.5, 2, 100)
sim_W_HT = simulate_HT(0.5, 100)
sim_W_HTHT = simulate_HTHT(0.5, 100)
sim_W_HHHH = simulate_run(0.5, 4, 100)

print("E(time to see HT) = ", np.mean(sim_W_HT))
print("E(time to see HH) = ", np.mean(sim_W_HH))
print("E(time to see HTHT) = ", np.mean(sim_W_HTHT))
print("E(time to see HHHH) = ", np.mean(sim_W_HHHH))
```

1.4 Some Result of 100 repetitions test

- Some results of 100 repetitions test are shown below:

```
# Result 1
E(time to see HT) =  4.31
E(time to see HH) =  5.76
E(time to see HTHT) =  18.5
E(time to see HHHH) =  25.6

# Result 2
E(time to see HT) =  4.04
E(time to see HH) =  5.62
E(time to see HTHT) =  20.46
E(time to see HHHH) =  31.5

# Result 3
E(time to see HT) =  3.81
E(time to see HH) =  5.1
E(time to see HTHT) =  20.22
E(time to see HHHH) =  27.95

# Result 4
E(time to see HT) =  3.88
E(time to see HH) =  6.48
E(time to see HTHT) =  21.11
E(time to see HHHH) =  27.1

# Result 5
E(time to see HT) =  3.89
E(time to see HH) =  5.52
E(time to see HTHT) =  17.72
E(time to see HHHH) =  35.57
```

- Since the repetitions 100 is not large enough, it is hard to conclude and verify the Expectation to see a specific string. Then let's set a 10000 repetitions.

1.5 Testing Code (10000 repetitions)

- calculates the mean value of 10000 repetitions and prints them on the screen

```
sim_W_HH = simulate_run(0.5, 2, 10000)
sim_W_HT = simulate_HT(0.5, 10000)
sim_W_HTHT = simulate_HTHT(0.5, 10000)
sim_W_HHHH = simulate_run(0.5, 4, 10000)

print("E(time to see HT) = ", np.mean(sim_W_HT))
print("E(time to see HH) = ", np.mean(sim_W_HH))
print("E(time to see HTHT) = ", np.mean(sim_W_HTHT))
print("E(time to see HHHH) = ", np.mean(sim_W_HHHH))
```

1.6 One Result of 10000 repetitions test

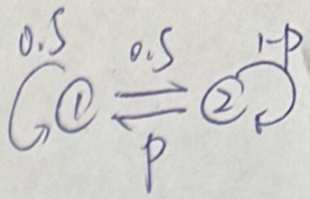
- One of the results of 10000 repetitions test is shown below

```
E(time to see HT) =  4.0084
E(time to see HH) =  6.0531
E(time to see HTHT) =  20.1087
E(time to see HHHH) =  30.1223
```

- According to the result, we can see that

$$E[\text{time to see } HT] \approx 4$$
$$E[\text{time to see } HH] \approx 6$$
$$E[\text{time to see } HTHT] \approx 20$$
$$E[\text{time to see } HHHH] \approx 30$$

Problem 12. Solution:

The transition matrix $P = \begin{bmatrix} 0.5 & 0.5 \\ p & 1-p \end{bmatrix}$ 

Balance Equation: $\pi_1 = \frac{1}{2}\pi_1 + p\pi_2 \Rightarrow \pi_1 = 2p\pi_2$

According to the statement of problem, we know

~~that $\pi_1 = \frac{1}{10}$, $\pi_2 = \frac{9}{10}$ (or $\pi_2 = 9\pi_1 \Rightarrow \pi_1 = \frac{1}{9}\pi_2$)~~

$\cancel{\Rightarrow \begin{cases} \pi_1 = 2p\pi_2 \\ \pi_1 = \frac{1}{9}\pi_2 \end{cases} \Rightarrow 2p = \frac{1}{9} \Rightarrow p = \frac{1}{18}}$

Normalization Equation: $\pi_1 + \pi_2 = 1 \Rightarrow 2p\pi_2 + \pi_2 = 1$

$(1+2p)\pi_2 = 1$

$p = \dfrac{1-\pi_2}{2\pi_2}$

if $\hat{\pi}_2$ is an estimate of π_2

then an estimate of $\hat{p} = \dfrac{1-\hat{\pi}_2}{2\hat{\pi}_2}$

we have $\hat{\pi}_2 = 9/10$

then $\hat{p} = \dfrac{1 - 9/10}{2 \times 9/10} = \dfrac{1}{18}$