ECS455 Chapter 3

Call Blocking Probability

Dr.Prapun Suksompong
prapun.com/ecs455

Office Hours:
BKD 3601-7

Tuesday	**9:30-10:30**
Tuesday	**13:30-14:30**
Thursday	**13:30-14:30**

Introduction

- The English dictionary word with the most consecutive vowels (six) is **EUOUAE**.
  - It is also the longest English word consisting only of vowels

- Imagine a word with **five** consecutive vowels.

Introduction

- Words with five consecutive vowels include **AIEEE**, C**OOEEI**NG, M**IAOUE**D, Z**AOUIA**, JUSS**IEUEA**N, Z**OOEAE**, Z**OAEAE**.

Introduction

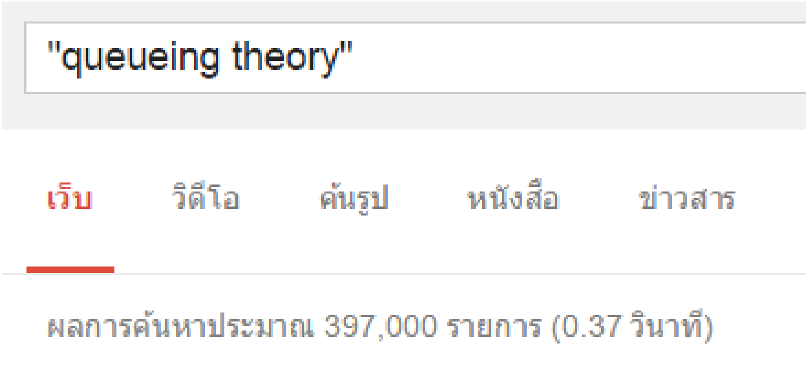
- Words with five consecutive vowels include **AIEEE**, C**OOEEI**NG, M**IAOUE**D, Z**AOUIA**, JUSS**IEUEA**N, Z**OOEAE**, Z**OAEAE**.

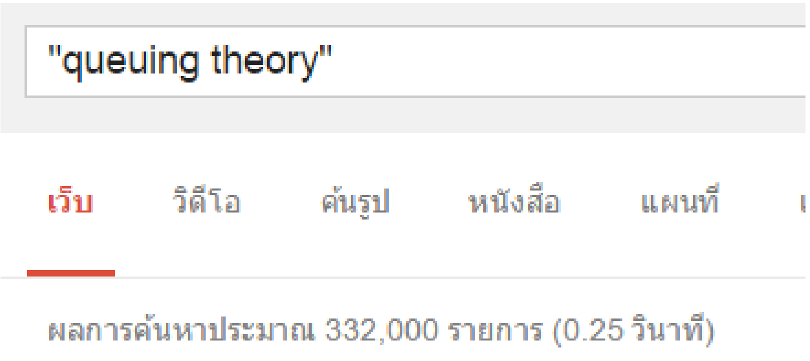
- Our new topic: Q**UEUEI**NG THEORY.
  - This is the only common word in the English language with five consecutive vowels.
- Note: The longest common word without any of the five vowels is RHYTHMS.
  - There are longer rare words: SYMPHYSY, NYMPHLY, GYPSYRY, GYPSYFY, and TWYNDYLLYNGS. WPPWRMWSTE and GLYCYRRHIZIN are long words with very few vowels.

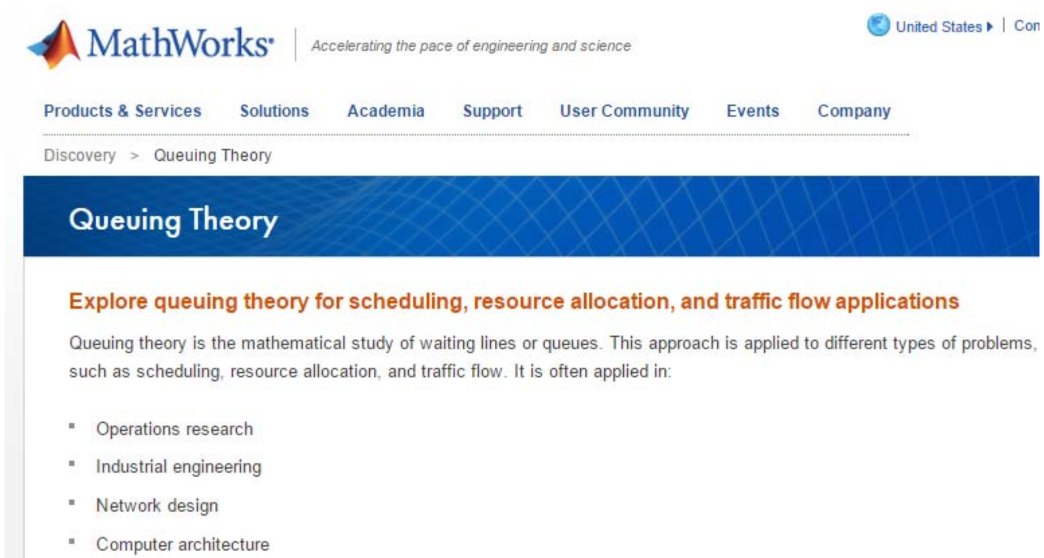
That Second "e"…

- You may recall the rule for changing a verb into its "–ing" form from your English class…
- If the verb ends in an "e" we remove the "e" and add "-ing":
  - browsing, causing, changing, charging, choosing, giving, having, hiring

Queueing Theory







Limiting distribution

```
>> P = [2/5 3/5; 1/2 1/2]
P =
    0.4000    0.6000
    0.5000    0.5000
```

```
>> n1 = [0 1e5]
n1 =
           0      100000
>> n2 = n1*P
n2 =
       50000       50000
>> n3 = n2*P
n3 =
       45000       55000
>> n4 = n3*P
n4 =
       45500       54500
>> n5 = n4*P
n5 =
       45450       54550
>> n6 = n5*P
n6 =
       45455       54545
```

$\underline{n}^{(1)}$

$\underline{n}^{(2)} = \underline{n}^{(1)} P$

$\underline{n}^{(3)} = \underline{n}^{(2)} P = \underline{n}^{(1)} P^2$

$\underline{n}^{(4)} = \underline{n}^{(3)} P = \underline{n}^{(1)} P^3$

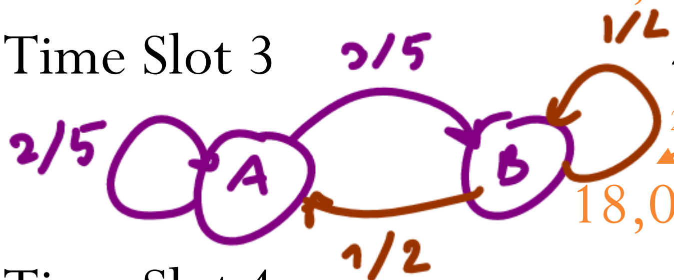
$\underline{n}^{(5)} = \underline{n}^{(1)} P^4$

$\underline{n}^{(6)} = \underline{n}^{(1)} P^5$

```
>> n1 = [1e4 9e4]
n1 =
       10000       90000
>> n2 = n1*P
n2 =
       49000       51000
>> n3 = n2*P
n3 =
       45100       54900
>> n4 = n3*P
n4 =
       45490       54510
>> n5 = n4*P
n5 =
       45451       54549
>> n6 = n5*P
n6 =
   1.0e+04 *
    4.5455    5.4545
```

Limiting distribution

	n_A = # trials in state A	n_B = # trials in state B
Time Slot 1	0	100,000
	0 + 50,000	0 + 50,000
Time Slot 2	50,000	50,000
	20,000 + 25,000	30,000 +25,000
Time Slot 3	45,000	55,000
	18,000 + 27,500	27,000 +27,500
Time Slot 4	45,500	54,500
Time Slot 5	45,450	54,550
Time Slot 6	45,455	54,545
time slot k	$n_A^{(current)}$	$n_B^{(current)}$
k+1	$\frac{2}{5}n_A + \frac{1}{2}n_B = n_A^{(new)}$	$\frac{3}{5}n_A + \frac{1}{2}n_B = n_B^{(new)}$



$$\left[n_A^{(new)} \quad n_B^{(new)}\right] = \left[n_A^{(cur)} \quad n_B^{(cur)}\right]\begin{bmatrix} 2/5 & 3/5 \\ 1/2 & 1/2 \end{bmatrix}$$

Limiting distribution

```
>> P = [2/5 3/5; 1/2 1/2]
P =
    0.4000    0.6000
    0.5000    0.5000

>> n1 = [0 1e5]
n1 =
           0      100000
>> n2 = n1*P
n2 =
       50000       50000
>> n3 = n2*P
n3 =
       45000       55000
>> n4 = n3*P
n4 =
       45500       54500
>> n5 = n4*P
n5 =
       45450       54550
>> n6 = n5*P
n6 =
       45455       54545
```

```
>> P = [2/5 3/5; 1/2 1/2]
P =
    0.4000    0.6000
    0.5000    0.5000
>> P^2
ans =
    0.4600    0.5400
    0.4500    0.5500
>> P^3
ans =
    0.4540    0.5460
    0.4550    0.5450
>> P^4
ans =
    0.4546    0.5454
    0.4545    0.5455
>> P^5
ans =
    0.4545    0.5455
    0.4546    0.5455
>> P^6
ans =
    0.4545    0.5455
    0.4545    0.5455
```

Limiting distribution

	n_A = # trials in state A	n_B = # trials in state B
Time Slot 1	0	100,000
Time Slot 2	50,000	50,000
Time Slot 3	45,000	55,000
Time Slot 4	45,500	54,500
Time Slot 5	45,450	54,550
Time Slot 6	45,455	54,545

$$\times \frac{1}{100,000}$$

	p_A = proportion of trials in state A	p_B = proportion of trials in state B
Time Slot 1	0	1
Time Slot 2	0.5	0.5
Time Slot 3	0.45000	0.55000
Time Slot 4	0.45500	0.54500
Time Slot 5	0.45450	0.54550
Time Slot 6	0.45455	0.54545

$$[\pi_A \quad \pi_D]\begin{bmatrix} 5/11 & 6/11 \\ 5/11 & 6/11 \end{bmatrix} = \begin{bmatrix} \frac{5}{11}\pi & \frac{6}{11}\pi \end{bmatrix} \qquad P^2 = VDV^{-1}VDV^{-1} = VD^2V^{-1}$$

$$P^3 = VD^2V^{-1}VDV^{-1} = VD^3V^{-1}$$

Limiting distribution

- Start with $\mathbf{P} = \begin{bmatrix} 2/5 & 3/5 \\ 1/2 & 1/2 \end{bmatrix}$
- Want to find $\lim_{n\to\infty} \mathbf{P}^n$.
- To do this, we first decompose $\mathbf{P}$ into $\mathbf{P} = \mathbf{VDV}^{-1}$.
  - In MATLAB, `[V,D] = eig(P)` produces a diagonal matrix $\mathbf{D}$ of eigenvalues and a matrix $\mathbf{V}$ whose columns are the corresponding eigenvectors.
- Finally,

$$\lim_{n\to\infty} \mathbf{P}^n = \lim_{n\to\infty} (\mathbf{VDV}^{-1})^n = \lim_{n\to\infty} \mathbf{VD}^n\mathbf{V}^{-1} = \mathbf{V}\left(\lim_{n\to\infty} \mathbf{D}^n\right)\mathbf{V}^{-1}$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

```
>> P = sym([2/5 3/5; 1/2 1/2])
P =
[ 2/5, 3/5]
[ 1/2, 1/2]

>> [V,D] = eig(P)
V =
[ -6/5, 1]
[    1, 1]
D =
[ -1/10, 0]
[     0, 1]
```

$$P^n = VD^nV^{-1}$$

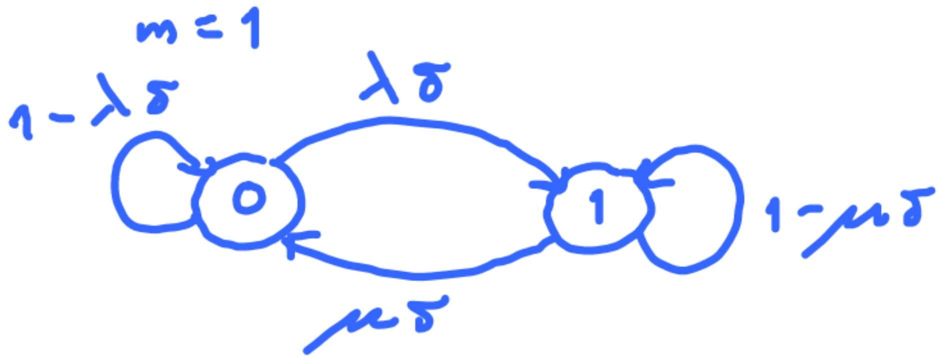
$$\lim_{n\to\infty} \mathbf{D}^n = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

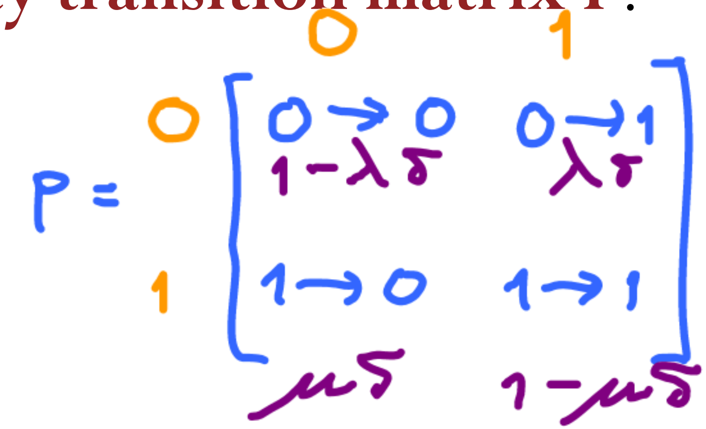
```
>> V*D*(V^(-1))
ans =
[ 2/5, 3/5]
[ 1/2, 1/2]
```

```
>> Dinf = sym([0,0;0,1])
Dinf =
[ 0, 0]
[ 0, 1]
>> V*Dinf*(V^(-1))
ans =
[ 5/11, 6/11]
[ 5/11, 6/11]
```

Review: Discrete-Time Markov Chain

- We model the evolution in time of K by Markov chain.
  - $K(t)$ = the number of channels being occupied at time t
- Time is divided into small slots so that our analysis can be done in discrete time.
  - This only approximate the solution. However, the answers will be accurate in the limit that the slot size δ approaches 0.
- Discrete-time Markov chain can be specified via its **state transition diagram** or its **probability transition matrix P**.





Simulating a Markov Chain in MATLAB

```
function X = MarkovChainGS(n,S,P,X1)
% n = the number of slots to be considered
% S = a row vector containing possible states (usually 1:N)
% P = transition probability matrix
% X1 = initial state for slot 1

N = length(S);      % Number of possible states
T = zeros(1,n);     % Preallocation
T(1) = find(S==X1); % Express the states using indices from 1 to N
                    % instead of the provided support S
for k = 2:n
    T(k) =  randsrc(1,1,[S;P(T(k-1),:)]);
end
X = S(T);           % Express the states using the provided support
end
```

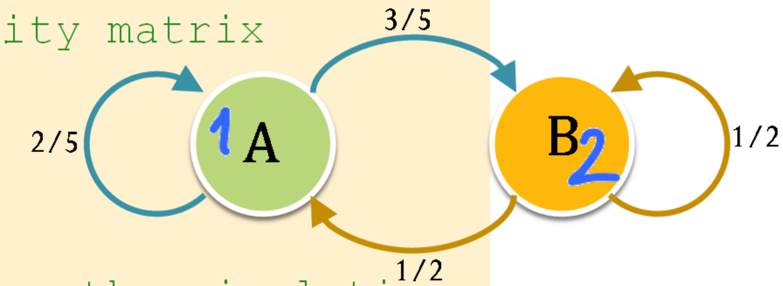
Simulating a Markov Chain in MATLAB

```
n = 1e1;                     % The number of slots to be considered
S = [1,2];                   % Two possible states
P = [2/5 3/5; 1/2 1/2];      % Transition probability matrix
X1 = 2;                      % Initial state

X = MarkovChainGS(n,S,P,X1)

% Approximate the transition probabilities from the simulation
P_sim = []; x = X(1:(n-1)); y = X(2:n);
for k = 1:length(S)
    I = find(x==S(k)); LI = length(I);
    yc = y(I); cond_rel_freq = hist(yc,S)/LI;
    P_sim = [P_sim; cond_rel_freq];
end
P_sim

% Approximate the proportions of time that the states occur
p_sim = hist(X,S)./n
```



$$P = \begin{array}{c} \\ A \\ B \end{array} \begin{array}{cc} A & B \\ \hline \end{array} \begin{bmatrix} 2/5 & 3/5 \\ 1/2 & 1/2 \end{bmatrix}$$

[MarkovChain_Demo1.m]

Simulating a Markov Chain in MATLAB

```
>> MarkovChain_Demo1
X =
     2     2     1     1     1     2     2     1     2     1
P_sim =
    0.5000    0.5000
    0.6000    0.4000
p_sim =
    0.5000    0.5000
```

slot 1 slot 2 slot 3 ... slot 10

P_X

The proportion of time that state 1 (A) occurs: $\frac{5}{10} = \frac{1}{2}$

The proportion of time that state 2 (B) occurs: $\frac{5}{10} = \frac{1}{2}$

Exercises

```
>> MarkovChain_Demo1
X =
     2     1     2     1     2     1     2     1     1     1
P_sim =
```

	1	2
1	2/5 = 0.4	0.6
2	1	0

There are 5 transitions from state 1

```
p_sim =
```

0.6	0.4

```
>> MarkovChain_Demo1
X =
     2     2     2     2     1     2     2     2     1     1
P_sim =
```

0.5	0.5
2/7	5/7

```
p_sim =
```

0.3	0.7

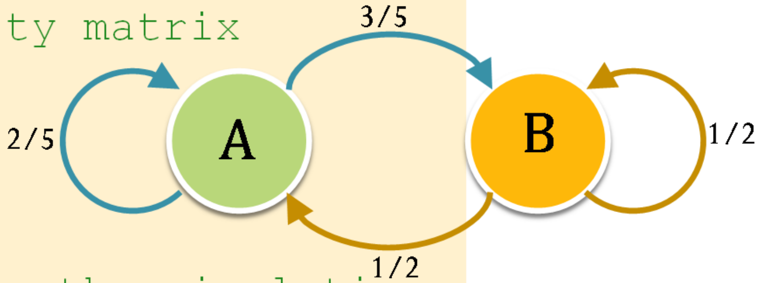
Simulating a Markov Chain in MATLAB

```
n = 1e4;                   % The number of slots to be considered
S = [1,2];                 % Two possible states
P = [2/5 3/5; 1/2 1/2];    % Transition probability matrix
X1 = 2;                    % Initial state

X = MarkovChainGS(n,S,P,X1);

% Approximate the transition probabilities from the simulation
P_sim = []; x = X(1:(n-1)); y = X(2:n);
for k = 1:length(S)
    I = find(x==S(k)); LI = length(I);
    yc = y(I); cond_rel_freq = hist(yc,S)/LI;
    P_sim = [P_sim; cond_rel_freq];
end
P_sim

% Approximate the proportions of time that the states occur
p_sim = hist(X,S)./n
```




```
>> MarkovChain_Demo2
P_sim =
      0.4007     0.5993
      0.5055     0.4945
p_sim =
      0.4575     0.5425
```

≈ steady-state probabilities

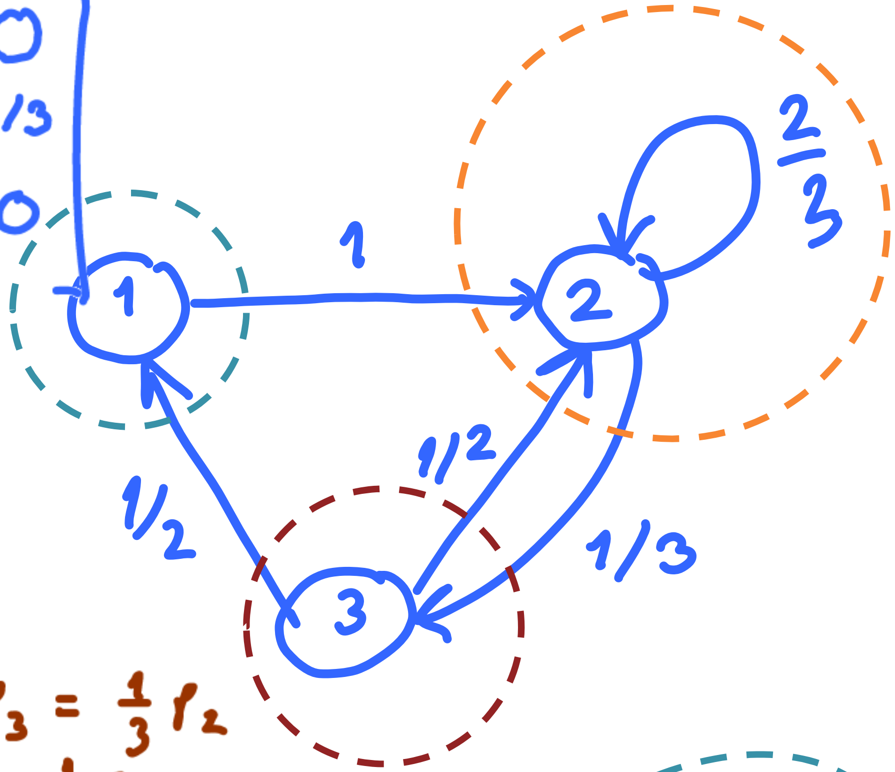
[MarkovChain_Demo2.m]

Review: Steady-State Probabilities

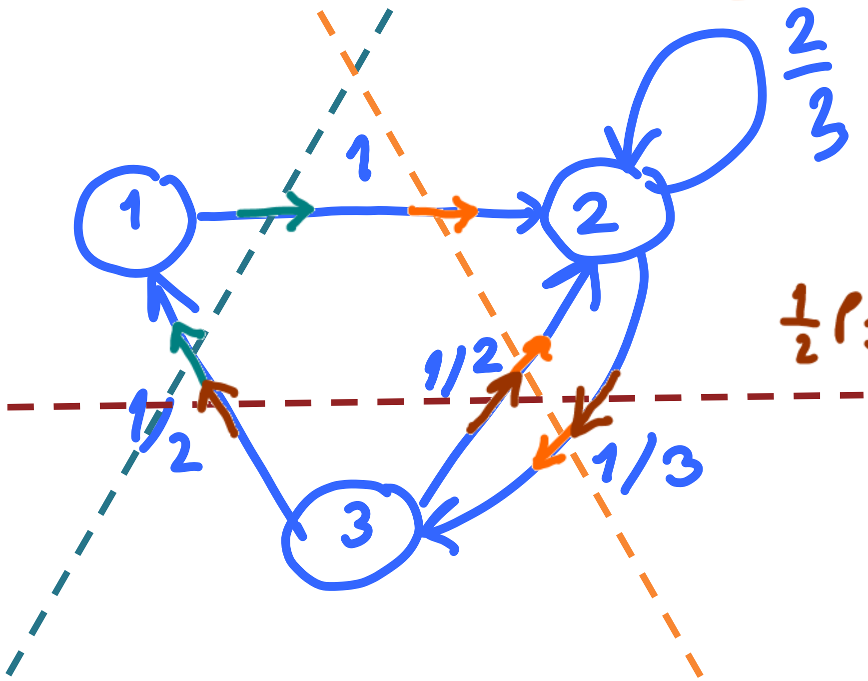
- Long-term behavior of a discrete-time Markov chain can be studied in terms of its **steady-state** (or limiting or equilibrium) probabilities.
  - To find these probabilities, we use balance equations together with the fact that
- To write down a **balance equation**,
  - first define a boundary,
  - then consider the transfer of probabilities "in" and "out" of the boundary.
  - To be at equilibrium, there should not be any net transfer.

Example

$$P = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 0 \\ 2 & 0 & 2/3 & 1/3 \\ 3 & 1/2 & 1/2 & 0 \end{array}$$

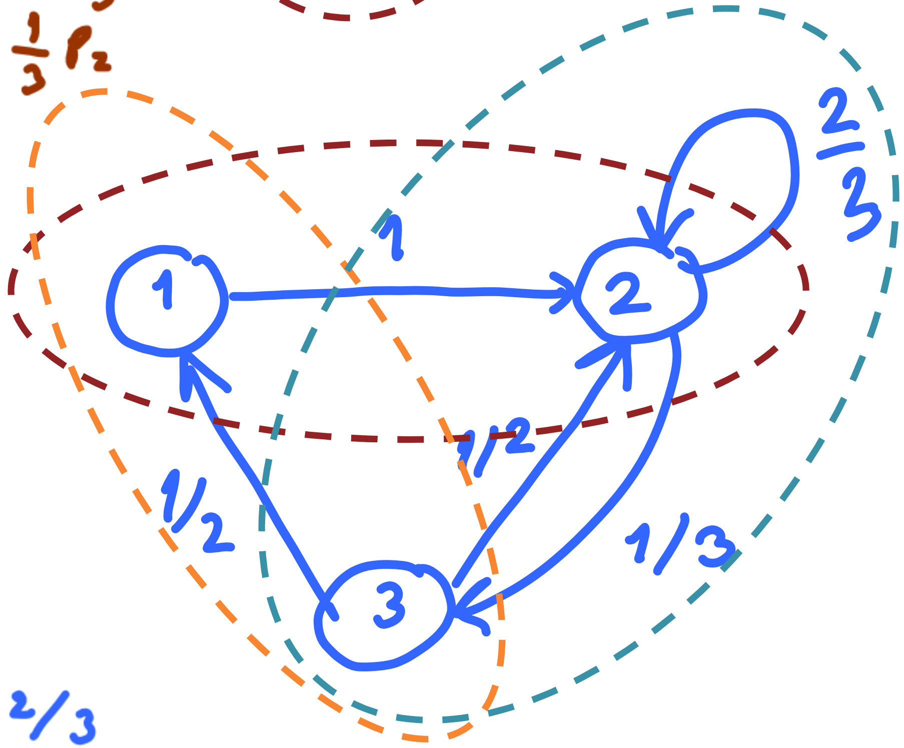


$$p_1 \times 1 + p_3 \times \frac{1}{2} = p_2 \times \frac{1}{3}$$



$$\frac{1}{2} p_3 + \frac{1}{2} p_3 = \frac{1}{3} p_2$$

$$p_3 = \frac{1}{3} p_2$$



$$p_1 \times 1 = \frac{1}{2} p_3$$

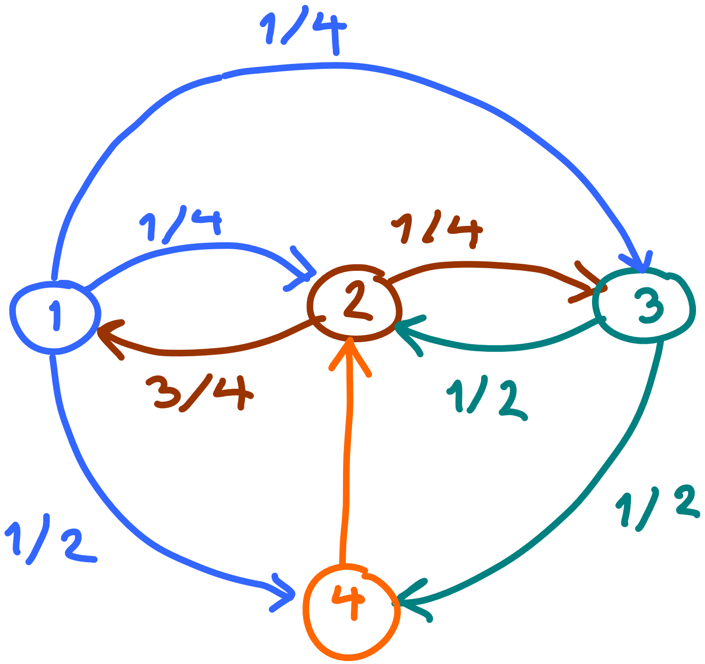
$$p_1 + p_2 + p_3 = 1$$

$$p_1 + 6p_1 + 2p_1 = 1$$

$$p_1 = 1/9$$

$$p_3 = 2p_1 = 2/9 \qquad p_2 = 6p_1 = 6/9 = 2/3$$

Exercise

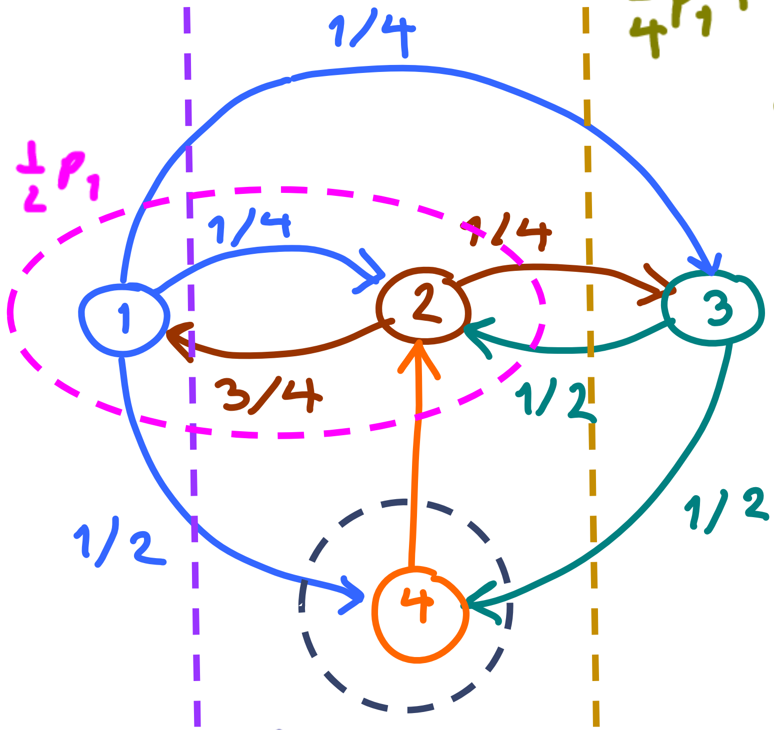


$$p_1 = \frac{3}{4} p_2$$

$$\frac{1}{4} p_1 + \frac{1}{4} p_1 + \frac{1}{2} p_1 = \frac{3}{4} p_2$$

$$\frac{1}{4} p_1 + \frac{1}{4} p_2 + \frac{1}{2} p_1 = \frac{1}{2} p_3 + p_4$$

$$\frac{1}{4} p_1 + \frac{1}{4} p_2 = \frac{1}{2} p_3 + \frac{1}{2} p_3$$



$$\frac{1}{2} p_3 + \frac{1}{2} p_1 = p_4$$

Review: Problem Solving

- For any question that requires you to get your answers (call blocking probability or steady-state probabilities) "from the Markov chain" or "via the Markov chain", make sure that you
  - draw the Markov chain
  - set up the boundaries and write down the corresponding balance equations

Review: Two Interpretations

- Two Interpretations of steady-state probabilities:
  When we let a system governed by a Markov chain evolve for a long time
  - at a particular slot, the probability that we will find the system in a particular state can be approximated by its corresponding steady-state probability,
  - considering the whole evolution up to a particular time, the **proportion of time that the system is in a particular state** can be approximated by its corresponding steady-state probability.

Example

```
n = 1e4;                        % The number of slots to be considered
S = [1,2,3];                    % Three possible states
P = [0 1 0; 0 2/3 1/3; 1/2 1/2 0]; % Transition probability matrix
X1 = 2;                         % Initial state

X = MarkovChainGS(n,S,P,X1);

% Approximate the transition probabilities from the simulation
P_sim = []; x = X(1:(n-1)); y = X(2:n);
for k = 1:length(S)
    I = find(x==S(k)); LI = length(I);
    yc = y(I); cond_rel_freq = hist(yc,S)/LI;
    P_sim = [P_sim; cond_rel_freq];
end
P_sim

% Approximate the proportions of time that the states occur
p_sim = hist(X,S)./n
```

```
>> MarkovChain_Demo3
P_sim =
         0    1.0000         0
         0    0.6620    0.3380
    0.4858    0.5142         0
p_sim =
    0.1093    0.6657    0.2250
```

Review: Call–Blocking Probability

- **Call blocking probability P_b** is the (long-term) proportion of calls that get blocked by the system because all channels are occupied.
- For M/M/m/m system, the (long-term) call blocking probability P_b is given by P_m

= the steady-state probability for state *m*
= the (long-term) proportion of time that the system will be in state *m*