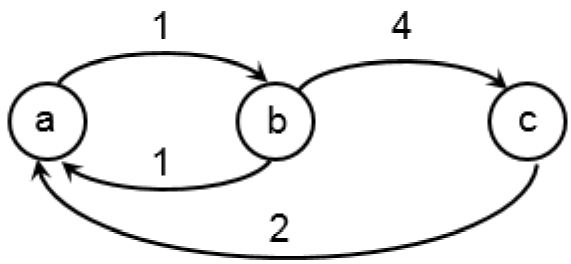
Solutions to Mini-Project #2 for Computer Networks (Fall 2014)

1) For the CTMC below do the following.

- Write and then solve as three equations in three unknowns the Q matrix.
- Use uniformization to convert the Q matrix to a P matrix. Give the P matrix and then solve it as three equations in three unknowns
- Solve the P matrix using the power (iterative) method. Hint: You can use `iter.c` found on the Christensen Tools page and show a screenshot of the execution.



From this we can directly write the transition rate matrix (Q matrix) as:

$$Q = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -5 & 4 \\ 2 & 0 & -2 \end{bmatrix}$$

where the rates are in transitions per day. We know that $0 = \pi Q$ where π_1, π_2, and π_3 (where for the states 1 = a, 2 = b, and 3 = c) are the steady state probabilities that we seek. From the Q matrix we can write the following equations:

$$\begin{aligned} -1\pi_1 + 1\pi_2 + 2\pi_3 &= 0 \\ 1\pi_1 - 5\pi_2 &= 0 \\ 4\pi_2 - 2\pi_3 &= 0 \end{aligned}$$

These equations are dependent, so we break the dependence with the identify (sum of steady state probabilities must be 1). So,

$$\begin{aligned} -1\pi_1 + 1\pi_2 + 2\pi_3 &= 0 \\ 1\pi_1 - 5\pi_2 &= 0 \\ \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned}$$

This we can solve and we get $\pi_1 = 0.625$, $\pi_2 = 0.125$, and $\pi_3 = 0.250$. Using uniformization, we convert the Q matrix into a P matrix:

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} -1/-5 & 1/-5 & 0/-5 \\ 1/-5 & -5/-5 & 4/-5 \\ 2/-5 & 0/-5 & -2/-5 \end{bmatrix} = \begin{bmatrix} 0.80 & 0.20 & 0 \\ 0.20 & 0 & 0.80 \\ 0.40 & 0 & 0.60 \end{bmatrix}$$

From the P matrix we write the following equations following from $\pi = \pi P$:

$$
\begin{aligned}
0.80\pi_1 + 0.20\pi_2 + 0.40\pi_3 &= \pi_1 \\
0.20\pi_1 &= \pi_2 \\
0.80\pi_2 + 0.60\pi_3 &= \pi_3
\end{aligned}
$$

These equations are dependent, so we break the dependence with the identify (sum of steady state probabilities must be 1). So,

$$
\begin{aligned}
0.80\pi_1 + 0.20\pi_2 + 0.40\pi_3 &= \pi_1 \\
0.20\pi_1 &= \pi_2 \\
\pi_1 + \pi_2 + \pi_3 &= 1
\end{aligned}
$$

This we can solve and we get $\pi_1 = 0.625$, $\pi_2 = 0.125$, and $\pi_3 = 0.250$ (as expected!). Finally, we solve the P matrix iteratively (note that the index starts at 0 and not 1):

```
VC++ cmd

c:\work>type in.dat
P 3
& Assignment #2, problem #1     &
& ---------------- column #0    &
0       0.80
1       0.20
2       0.40
-1      zero
& ---------------- column #1    &
0       0.20
-1      zero
& ---------------- column #2    &
1       0.80
2       0.60
-1      zero
& ---------------- END FLAG     &
-999    zero

c:\work>iter < in.dat
------------------------------------------------- iter.c -----
  Iteration count = 10 -- Convergence mean = 0.333333
  Iteration count = 20 -- Convergence mean = 6.88355e-07
  Iteration count = 30 -- Convergence mean = 1.40378e-13
  Iteration count = 40 -- Convergence mean = 0
--------------------------------------------------------------
  Results for P matrix (size = 3 and tolerance = 1e-15)
    Pi[0] = 0.625
    Pi[1] = 0.125
    Pi[2] = 0.25
--------------------------------------------------------------

c:\work>
```

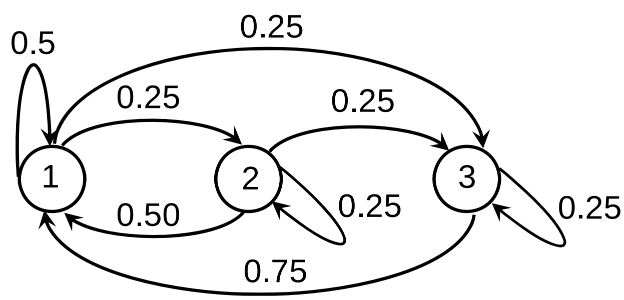
2) You are given a system or component (for example, a transceiver) with three states, BUSY, IDLE, and SLEEP (corresponding to the system or component being powered-on and busy, powered-on and idle, and sleeping, respectively). Let the power draw of each state be as follows:

- BUSY = 1 W
- IDLE = 0.8 W
- SLEEP = 0.1 W

Periodically, once per minute the system transitions between states (including to the same state) with a known probability as given below. What is the average power draw of this system or component? Write the state equations and solve them both directly and iteratively (e.g., using `iter.c` found on the Christensen tools page).

- BUSY to BUSY with probability 0.50, BUSY to IDLE with probability 0.25, and BUSY to SLEEP with probability 0.25
- IDLE to IDLE with probability 0.25, IDLE to BUSY with probability 0.50, and IDLE to SLEEP with probability 0.25.
- SLEEP to SLEEP with probability 0.25 and SLEEP to BUSY with probability 0.75.

First we draw the Markov chain (letting 1 = BUSY, 2 = IDLE, and 3 = SLEEP) with transition probabilities.



From this we can directly write the transition rate matrix (P matrix) as:

$$P = \begin{bmatrix} 0.50 & 0.25 & 0.25 \\ 0.50 & 0.25 & 0.25 \\ 0.75 & 0 & 0.25 \end{bmatrix}$$

We know that $\pi = \pi P$ where π_1, π_2, and π_3 (where for the states 1 = BUSY, 2 = IDLE, and 3 = SLEEP). From the P matrix we can write the following equations:

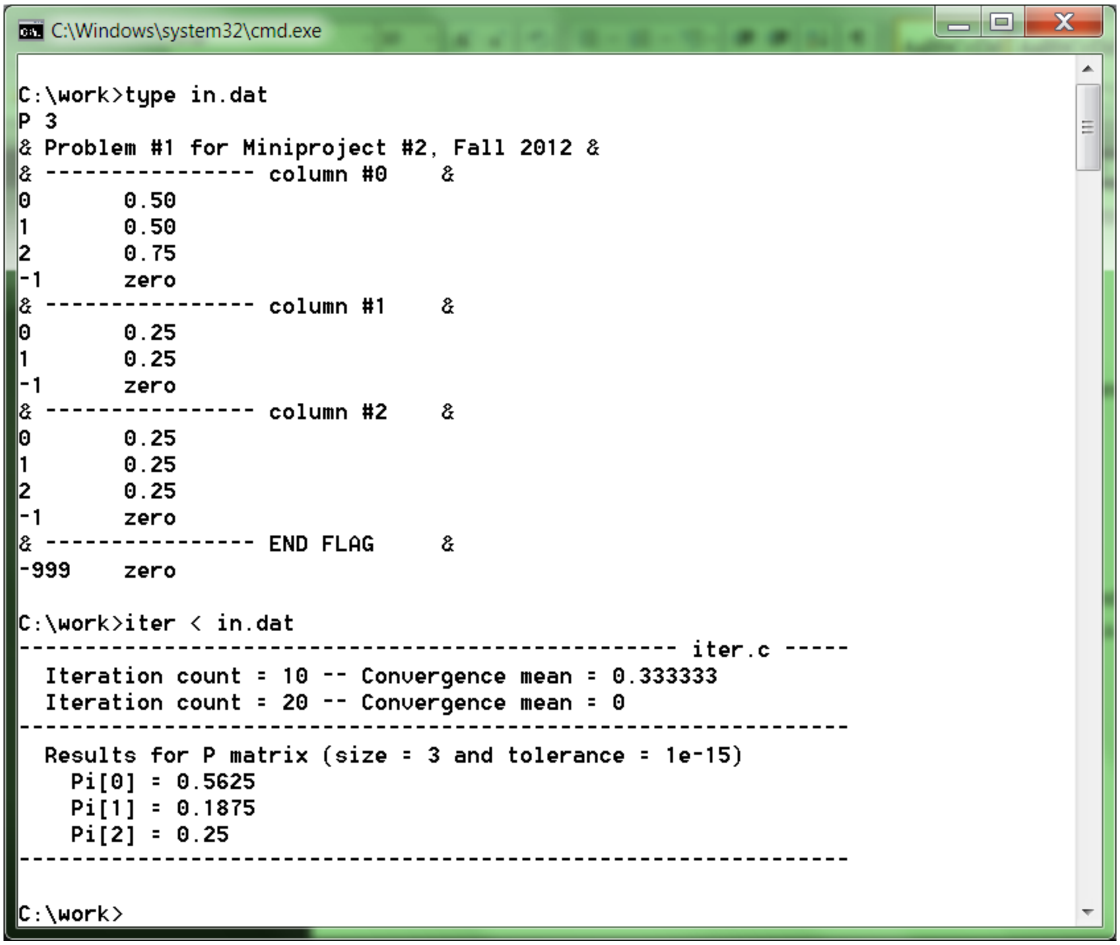
$$\begin{aligned} 0.50\pi_1 + 0.50\pi_2 + 0.75\pi_3 &= \pi_1 \\ 0.25\pi_1 + 0.25\pi_2 &= \pi_2 \\ 0.25\pi_1 + 0.25\pi_2 + 0.25\pi_3 &= \pi_3 \end{aligned}$$

These equations are dependent, so we break the dependence with the identify (sum of steady state probabilities must be 1). So,

$$
\begin{aligned}
0.50\pi_1 + 0.50\pi_2 + 0.75\pi_3 &= \pi_1 \\
0.25\pi_1 + 0.25\pi_2 &= \pi_2 \\
\pi_1 + \pi_2 + \pi_3 &= 1
\end{aligned}
$$

This we can solve and we get $\pi_1 = 9/16$, $\pi_2 = 3/16$, and $\pi_3 = 4/16$, So, the average power draw of the system or component is $(9/16)\cdot 1 + (3/16)\cdot 0.80 + (4/16)\cdot 0.1 = 0.738$ W.

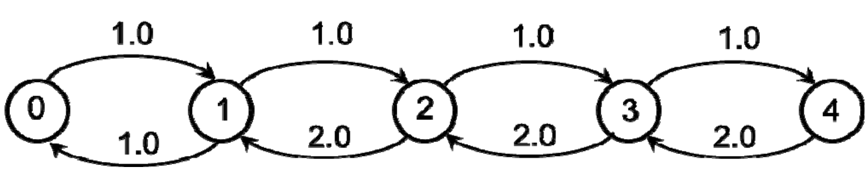
And, here is the P matrix solved iteratively using `iter.c` (note that index starts at 0 and not 1):



```
C:\Windows\system32\cmd.exe

C:\work>type in.dat
P 3
& Problem #1 for Miniproject #2, Fall 2012 &
& ---------------- column #0    &
0       0.50
1       0.50
2       0.75
-1      zero
& ---------------- column #1    &
0       0.25
1       0.25
-1      zero
& ---------------- column #2    &
0       0.25
1       0.25
2       0.25
-1      zero
& ---------------- END FLAG     &
-999    zero

C:\work>iter < in.dat
-------------------------------------------------- iter.c -----
  Iteration count = 10 -- Convergence mean = 0.333333
  Iteration count = 20 -- Convergence mean = 0
----------------------------------------------------------------
  Results for P matrix (size = 3 and tolerance = 1e-15)
    Pi[0] = 0.5625
    Pi[1] = 0.1875
    Pi[2] = 0.25
----------------------------------------------------------------

C:\work>
```

3) Consider an M/M/2/4 queue (a queue with two service centers and a buffer with capacity of two customers) with arrival rate 1.0 customers/second and service rate 1.0 customers/second for each service center. Answer the following questions (show your work!):

a) Sketch the continuous time Markov chain for this queue.



b) Solve the Markov chain for steady state probabilities and mean number in the system.

First we write the Q matrix directly from the ctmc.

$$Q = \begin{bmatrix} -1.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & -2.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 2.0 & -3.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 2.0 & -3.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 2.0 & -2.0 \end{bmatrix}$$

From this we can write five equations in five unknowns and solve it. Easier, however, is an iterative solution as shown here:

```
VC++ cmd
c:\work>type in.dat
Q 5
& Assignment #2, problem #3     &
& ---------------- column #0    &
0       -1.00
1        1.00
-1       zero
& ---------------- column #1    &
0        1.00
1       -2.00
2        2.00
-1       zero
& ---------------- column #2    &
1        1.00
2       -3.00
3        2.00
-1       zero
& ---------------- column #3    &
2        1.00
3       -3.00
4        2.00
-1       zero
& ---------------- column #4    &
3        1.00
4       -2.00
-1       zero
& ---------------- END FLAG     &
-999     zero

c:\work>
```

```
c:\work>iter < in.dat
-------------------------------------------------- iter.c -----
  Iteration count = 10 -- Convergence mean = 0.2
  Iteration count = 20 -- Convergence mean = 0.00717278
  Iteration count = 30 -- Convergence mean = 0.000411738
  Iteration count = 40 -- Convergence mean = 2.37385e-05
  Iteration count = 50 -- Convergence mean = 1.37151e-06
  Iteration count = 60 -- Convergence mean = 7.93181e-08
  Iteration count = 70 -- Convergence mean = 4.58927e-09
  Iteration count = 80 -- Convergence mean = 2.65587e-10
  Iteration count = 90 -- Convergence mean = 1.53714e-11
  Iteration count = 100 -- Convergence mean = 8.89695e-13
  Iteration count = 110 -- Convergence mean = 5.14935e-14
  Iteration count = 120 -- Convergence mean = 2.97262e-15
  Iteration count = 130 -- Convergence mean = 1.73472e-16
---------------------------------------------------------------
  Results for Q matrix (size = 5 and tolerance = 1e-15)
    Pi[0] = 0.347826
    Pi[1] = 0.347826
    Pi[2] = 0.173913
    Pi[3] = 0.0869565
    Pi[4] = 0.0434783
---------------------------------------------------------------

c:\work>_
```

4) You have invented a new method of routing packets end-to-end to minimize packet delay. You think that your method is better (that is, mean packet delay is less) than existing methods. To evaluate your method you have conducted experiments on a network test bed sending packets from multiple sources to a given destination. You have measurement results for the existing routing method and your new routing method (which you inserted into the network test bed routers). Your measurement results are:

Experiment	**Old method Delay (ms)**	**New method Delay (ms)**
1	2.11	1.95
2	2.10	2.94
3	2.30	1.98
4	3.01	2.10
5	2.05	2.22
6	1.95	1.99
7	2.05	2.00

You believe that your experiments are representative of the real Internet. Is your method better? Should it be deployed in the Internet? Below is a T-score table that may be helpful to your argument.

T-scores. Selected values of $t_{\alpha/2;N-1}$

	$\alpha/2 = 0.05$	$\alpha/2 = 0.025$
***N* – 1**	***t***	***t***
4	2.13	2.78
5	2.02	2.57
6	1.94	2.45
7	1.90	2.37
8	1.86	2.31
9	1.83	2.26
10	1.81	2.23
11	1.80	2.20
12	1.78	2.18

The mean packet delay (for the seven experiments) for the old method is 2.23 ms and for the new method it is 2.16 ms. This is about a 3% lower delay. First we should ask if an approximately 3% improvement is significant in any way. In most real systems such a small difference in response time would be insignificant compared to many other possible design parameters (such as final cost of the product in terms of development time, material, operation, and so on) to influence the decision whether to choose design A or B. But, since we were asked the question we will answer it. The approach to this problem is to use hypothesis testing (using the Student's t-test since we have less than 30 samples) for paired experiments. Subtracting the measured delays we get D = 0.16, -0.84, 0.32, 0.91, -0.17, -0.04, and 0.05 ms (the packet delay of the new system subtracted from the packet delay of the existing system). We compute the mean difference and sample standard deviation:

$\overline{D} = \frac{1}{7}\sum_{i=1}^{7} D_i = 0.056$ and $S = \sqrt{\frac{1}{7-1}\sum_{i=1}^{7}\left(D_i - \overline{D}\right)^2} = 0.528$. For 90% confidence and 6 degrees of freedom the t score is 1.94, for 95% confidence it is 2.45. Thus, the half-width for 90% CI is $1.94 \cdot \left(0.528/\sqrt{7}\right) = 0.387$ and for 95% CI it is $2.45 \cdot \left(0.528/\sqrt{7}\right) = 0.489$. The mean difference (0.056 ms) plus/minus the 90% and 95% CIs both cross zero. <u>Thus, with 90% and 95% confidence the new system is not better (less delay) than the old system</u>. But, even if this were not the case we still likely would "not care" since a 3% improvement is generally not of much interest.