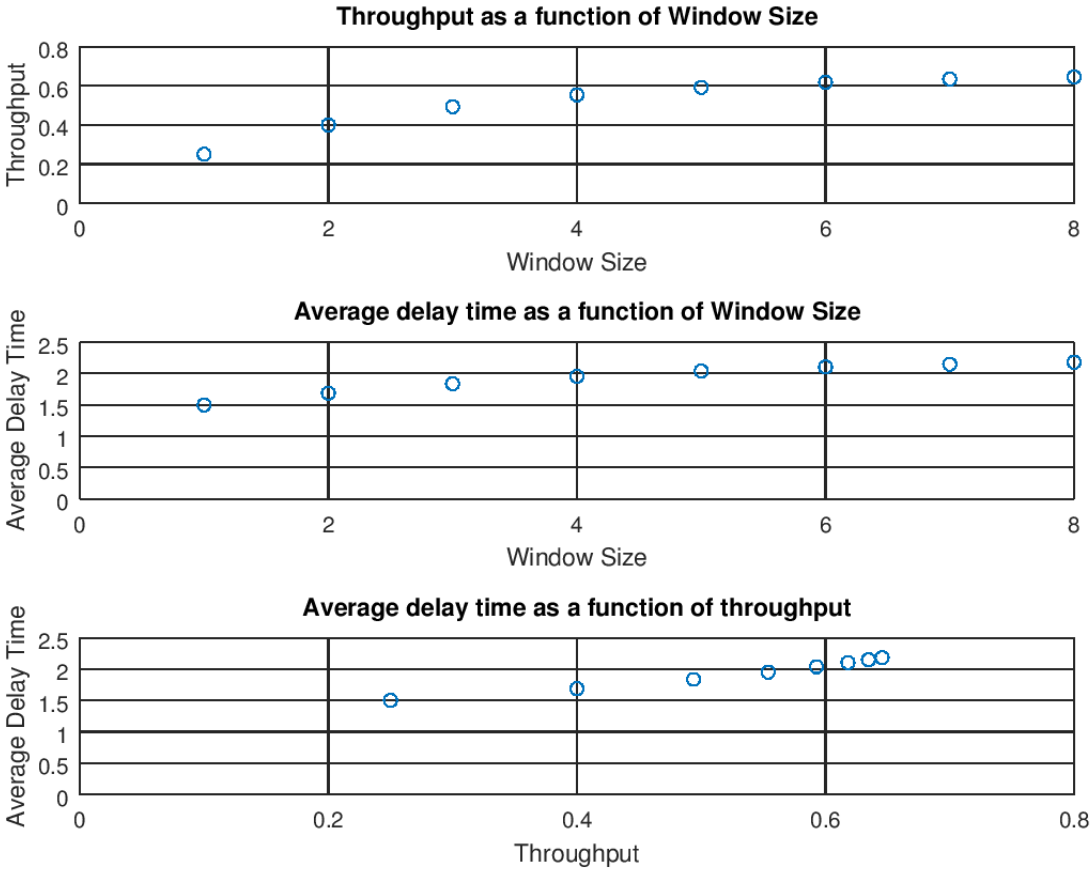
Συστήματα Αναμονής

6η Εργαστηριακή Άσκηση

Όνομα: Σταύρος Σταύρου
ΑΜ: 03115701
Εξάμηνο: 6ο-ΣΗΜΜΥ

Μηχανισμός ελέγχου ροής παραθύρου:

(1) Η Octave δίνει τα εξής διαγράμματα για τα μεγέθη που ζητούνται:



(2) Η Octave δίνει τα εξής (θεώρησα σταθερό μέγεθος παραθύρου W = 8):

```
usage =

   0.64549   0.32275   0.32275   0.32275   0.96824
   0.64549   0.32275   0.32275   0.32275   0.96824
   0.64549   0.32275   0.32275   0.32275   0.96824
   0.64549   0.32275   0.32275   0.32275   0.96824
   0.64549   0.32275   0.32275   0.32275   0.96824

waiting_time =

   2.48341   0.72745   0.72745   0.72745   7.72785
   1.24170   0.36372   0.36372   0.36372   3.86392
   0.82780   0.24248   0.24248   0.24248   2.57595
   0.62085   0.18186   0.18186   0.18186   1.93196
   0.49668   0.14549   0.14549   0.14549   1.54557

avg_clients =

   1.60303   0.46956   0.46956   0.46956   4.98828
   1.60303   0.46956   0.46956   0.46956   4.98828
   1.60303   0.46956   0.46956   0.46956   4.98828
   1.60303   0.46956   0.46956   0.46956   4.98828
   1.60303   0.46956   0.46956   0.46956   4.98828

gamma =

   0.64549   0.64549   0.64549   0.64549   0.64549
   1.29099   1.29099   1.29099   1.29099   1.29099
   1.93648   1.93648   1.93648   1.93648   1.93648
   2.58198   2.58198   2.58198   2.58198   2.58198
   3.22747   3.22747   3.22747   3.22747   3.22747
```

Παρατηρούμε πως ο μέσος αριθμός πελατών και χρησιμοποίησης του κάθε υποσυστήματος παραμένει σταθερός συναρτήσει των διαφόρων k. Αυτό φαίνεται λογικό καθώς οι εξισώσεις μετάβασης κατάστασης που παίρνουμε για τα διάφορα k είναι γραμμικά εξαρτημένες με την περίπτωση k=1 και ως εκ τούτου παίρνουμε τα ίδια αποτελέσματα.
Επίσης, παρατηρούμε πως η ρυθμαπόδοση είναι ίση μεταξύ των υποσυστημάτων (λογικό αφού το σύστημα έχει κοινή ρυθμαπόδοση) και όπως είναι λογικό αυξάνεται με τη μείωση των χρόνων εξυπηρέτησης, αφού πλέον στον ίδιο χρόνο εξυπηρετούμε περισσότερα πακέτα.
Τέλος, επίσης λόγω της μείωσης των χρόνων εξυπηρέτησης κάθε υποσυστήματος, βλέπουμε και μείωση στο χρόνο καθυστέρησης κάθε υποσυστήματος, αποτέλεσμα αναμενόμενο.

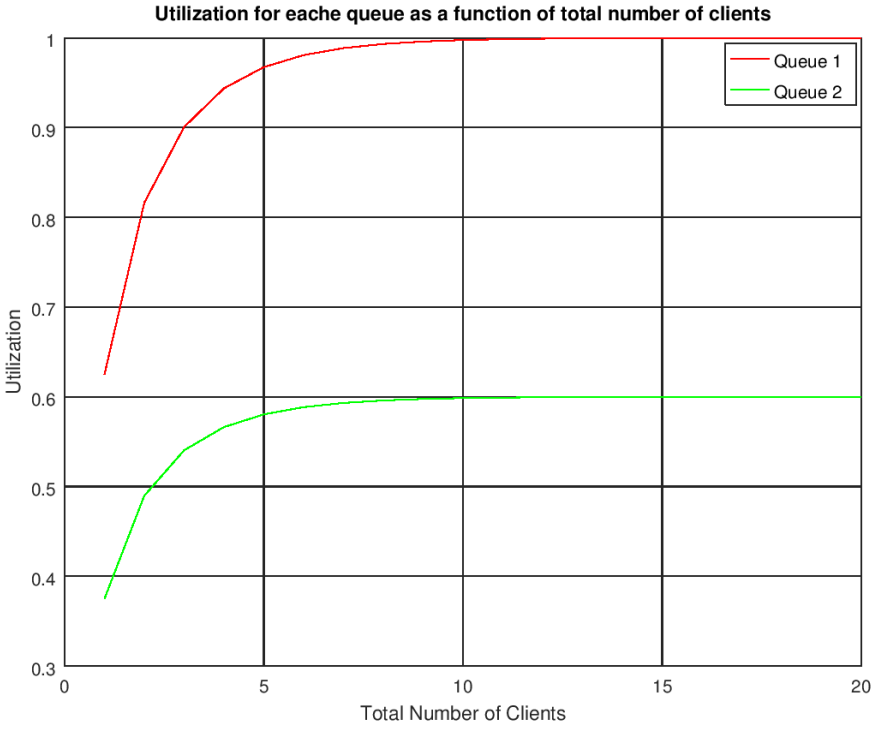
<u>Ο Αλγόριθμος του Buzen:</u>

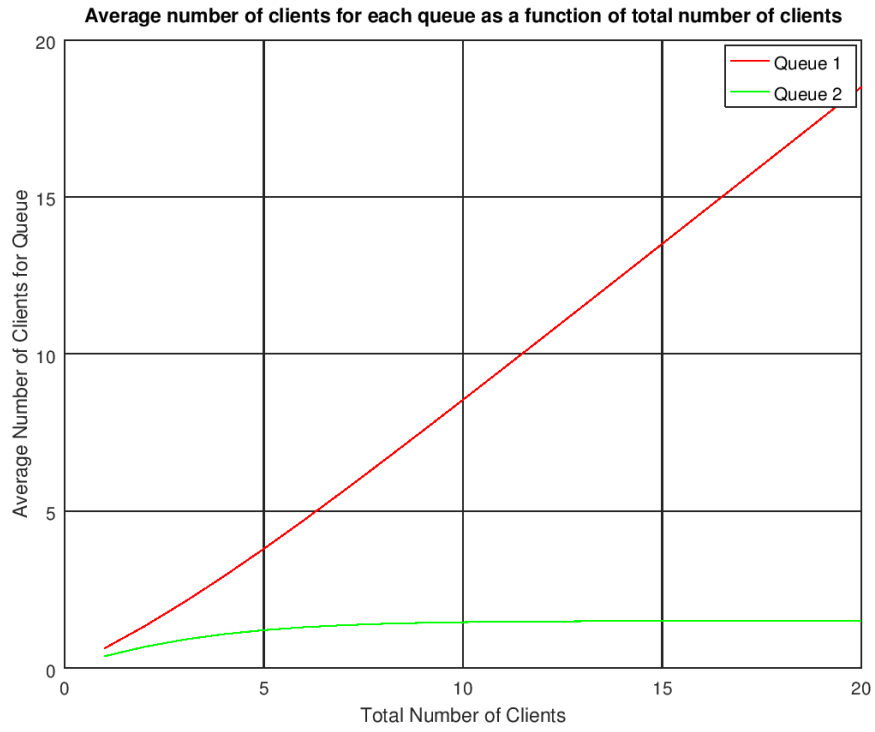
(1) Έχουμε $\mu_1 = 2,\ \mu_2 = 1$, $p_{11} = 1 - p = 0.7, p_{12} = p = 0.3, p_{21} = 1, p_{22} = 0$. Επίσης :
- $\mu_1 X_1 = \mu_1 X_1 p_{11} + \mu_2 X_2 p_{21} \rightarrow 2X_1 = 1.4X_1 + X_2 \rightarrow X_2 = 0.6X_1$
- $\mu_2 X_2 = \mu_1 X_1 p_{12} + \mu_2 X_2 p_{22} \rightarrow X_2 = 0.6X_1$

Βλέπουμε πως οι 2 εξισώσεις δίνουν την ίδια σχέση. Συνεπώς θέτουμε «αναγκαστικά» $X_1 = 1$, από το οποίο λαμβάνουμε $X_2 = 0.6$.

(2) Ο κώδικας στο παράρτημα. Η συνάρτηση επιστρέφει όλη την τελευταία στήλη του πίνακα ώστε οι τιμές αυτές να χρησιμοποιηθούν για το ερώτημα 3.

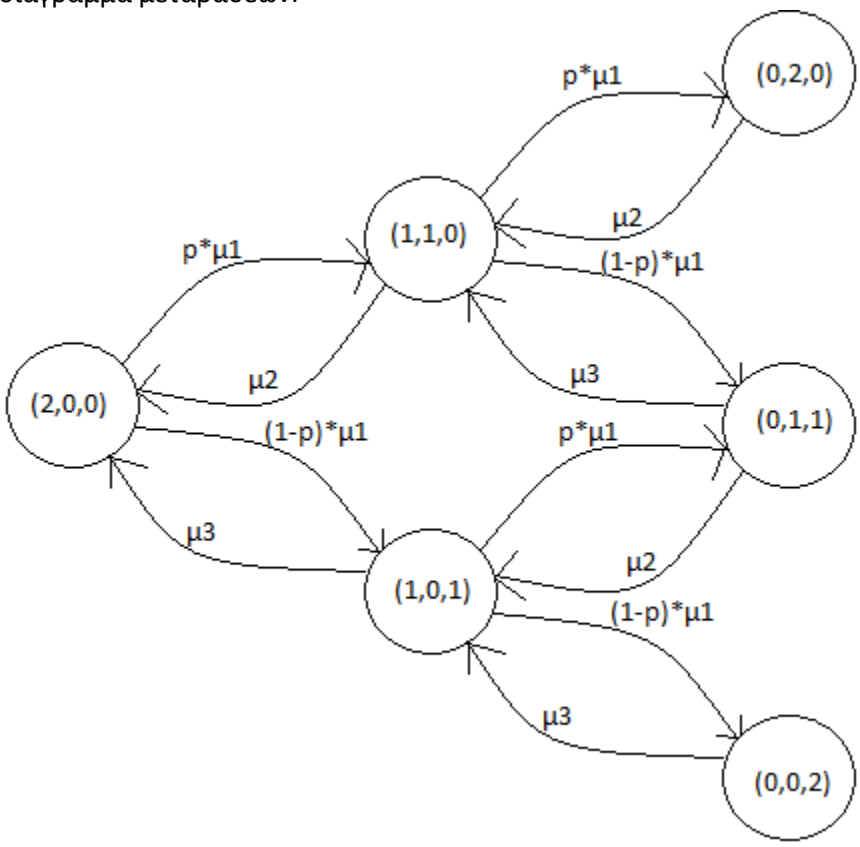
(3) Η Octave δίνει τις εξής γραφικές παραστάσεις με τη συνάρτηση που γράψαμε:





Προσομοίωση σε κλειστό δίκτυο εκθετικών ουρών αναμονής:

Έχουμε το εξής διάγραμμα μεταβάσεων:

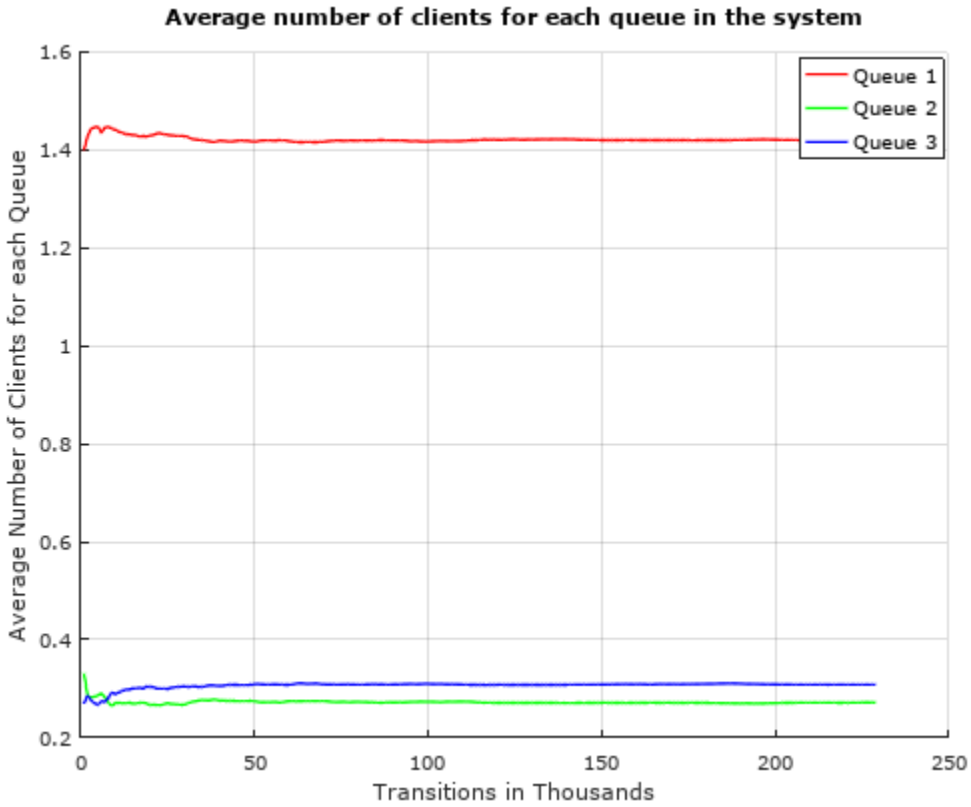


$$\mu_1 = 2, \mu_2 = 3, \mu_3 = 4, p = 0.4$$

(1) Παίρνουμε τα εξής από την Octave:

```
The probability for state (2,0,0) is 0.55347
The probability for state (1,1,0) is 0.14741
The probability for state (1,0,1) is 0.16573
The probability for state (0,2,0) is 0.040335
The probability for state (0,1,1) is 0.043745
The probability for state (0,0,2) is 0.049316
```

(2) Παίρνουμε την εξής γραφική παράσταση από την Octave:



Παρατηρούμε, πως το άθροισμα των μέσων όρων είναι ίσο με 2, γιατί τόσους πελάτες έχουμε στο σύστημα. Το άθροισμα πρέπει πάντα να είναι ακριβώς 2.

Όλοι οι κώδικες που χρησιμοποιήθηκαν για την άσκηση ακολουθούν στο παράρτημα.

ΠΑΡΑΡΤΗΜΑ

windows.m:

```
clc;
clear all;
close all;

Prob = [0 1 0 0 0; 0 0 1 0 0; 0 0 0 1 0; 0 0 0 0 1; 1 0 0 0 0];

V = qncsvisits(Prob);
S = [1 0.5 0.5 0.5 3/2];
m = [1 1 1 1 1];

gamma = zeros(1,8);
tDS = zeros(1,8);

for n = 1:1:8
  [U R Q X] = qnclosed(n, S, V, m);
  gamma(n) = mean(X);
  tDS(n) = (Q(2) + Q(3) + Q(4)) / gamma(n);
endfor

w = 1:1:8;

figure(1);
subplot(3, 1, 1);
scatter(w, gamma);
grid on;
title("Throughput as a function of Window Size");
xlabel("Window Size");
ylabel("Throughput");
xlim([0 8]);
ylim([0 0.8]);

subplot(3, 1, 2);
scatter(w, tDS);
grid on;
title("Average delay time as a function of Window Size");
xlabel("Window Size");
ylabel("Average Delay Time");
xlim([0 8]);
ylim([0 2.5]);

subplot(3, 1, 3);
scatter(gamma, tDS);
grid on;
title("Average delay time as a function of throughput");
xlabel("Throughput");
ylabel("Average Delay Time");
```

```
xlim([0 0.8]);
ylim([0 2.5]);
```

windows_diff_times.m:

```
clc;
clear all;
close all;

Prob = [0 1 0 0 0; 0 0 1 0 0; 0 0 0 1 0; 0 0 0 0 1; 1 0 0 0 0];

V = qncsvisits(Prob);
S = [1 0.5 0.5 0.5 3/2];
m = [1 1 1 1 1];

usage = zeros(5,5);
waiting_time = zeros(5,5);
avg_clients = zeros(5,5);
gamma = zeros(5,5);


for k = 1:1:5
  myS = (1 / k) * S;
  [usage(k,:) waiting_time(k,:) avg_clients(k,:) gamma(k,:)] = qnclosed(8, myS, V, m);
endfor

display(usage);
display(waiting_time);
display(avg_clients);
display(gamma);
```

buzen.m:

```
function [G Gconst] = buzen(N, M, X )

  if ( nargin != 3)
    print_usage();
  endif

  gamma = zeros(N+1, M+1);
  gamma(1, :) = ones(1, M+1);
  gamma(:, 1) = 0:1:N;
  for i = 2:1:N+1
    gamma(i, 2) = X(1) ** (i-1);
  endfor

  for i=2:1:N+1
    for j=3:1:M+1
```

```
    gamma(i,j) = gamma(i,j-1) + X(j-1) * gamma(i-1, j);
   endfor
  endfor
  Gconst = gamma(N+1, M+1);
  G = gamma(:, M+1);
endfunction
```

buzen_example.m:

```
clc;
clear all;
close all;

X = [1 0.6];
M = 2;
util1 = zeros(1,20);
util2 = zeros(1,20);
clients1 = zeros(1,20);
clients2 = zeros(1,20);
for N = 1:1:20
  [A B] = buzen(N, M, X);
  util1(N) = X(1) * A(N) / A(N+1);
  util2(N) = X(2) * A(N) / A(N+1);
  for k = 1:1:N
    clients1(N) = clients1(N) + X(1) ** k * A(N+1-k) / A(N+1);
    clients2(N) = clients2(N) + X(2) ** k * A(N+1-k) / A(N+1);
  endfor
endfor

N = 1:1:20;

figure(1);
hold on;
plot(N, util1, 'r');
plot(N, util2, 'g');
hold off;
grid on;
title("Utilization for eache queue as a function of total number of clients");
xlabel("Total Number of Clients");
ylabel("Utilization");
legend("Queue 1" , "Queue 2");
legend("show");

figure(2);
hold on;
plot(N, clients1, 'r');
plot(N, clients2, 'g');
hold off;
```

```
grid on;
title("Average number of clients for each queue as a function of total number of clients");
xlabel("Total Number of Clients");
ylabel("Average Number of Clients for Queue");
legend("Queue 1", "Queue 2");
legend("show");
```

closed.m:

```
clc;
clear all;
close all;

mu1 = 2;
mu2 = 3;
mu3 = 4;
p = 0.4;

arrivals(200) = 0;
arrivals(110) = 0;
arrivals(101) = 0;
arrivals(20) = 0;
arrivals(11) = 0;
arrivals(2) = 0;
total_arrivals = 0;
times = 0;
steps = 0;
sigklisi = false;

threshold0 = p*mu1/ (mu1+ mu2 + mu3);
threshold1 = mu1 / (mu1 + mu2 + mu3);
threshold2 = (mu1 + mu2) / (mu1 + mu2 + mu3);

prev_mean1 = 0;
prev_mean2 = 0;
prev_mean3 = 0;

current_state = 200;

while !sigklisi && steps <= 300000
  steps = steps + 1;
  arrivals(current_state) = arrivals(current_state) + 1;
  total_arrivals = total_arrivals + 1;
  decision = rand(1);

  if (current_state == 200 && decision < threshold0)
    current_state = 110;
  elseif (current_state == 200 && decision > threshold0 && decision < threshold1)
```

```
  current_state = 101;
elseif (current_state == 110 && decision < threshold0)
  current_state = 20;
elseif (current_state == 110 && decision > threshold0 && decision < threshold1)
  current_state = 11;
elseif (current_state == 110 && decision < threshold2 && decision > threshold1)
  current_state = 200;
elseif (current_state == 101 && decision < threshold0)
  current_state = 11;
elseif (current_state == 101 && decision > threshold0 && decision < threshold1)
  current_state = 2;
elseif (current_state == 101 && decision > threshold2)
  current_state = 200;
elseif (current_state == 20)
  current_state = 110;
elseif (current_state == 11 && decision > threshold2)
  current_state = 110;
elseif (current_state == 11 && decision > threshold1 && decision < threshold2)
  current_state = 101;
elseif (current_state == 2)
  current_state = 101;
else
  arrivals(current_state) = arrivals(current_state) - 1;
  total_arrivals = total_arrivals - 1;
endif

if mod(steps, 1000) == 0
  times = times + 1;

  T200 = 1 / mu1 * arrivals(200);
  T110 = 1 / (mu1 + mu2) * arrivals(110);
  T101 = 1 / (mu1 + mu3) * arrivals(101);
  T20 = 1 / mu2 * arrivals(20);
  T11 = 1 / (mu2 + mu3) * arrivals(11);
  T2 = 1 / mu3 * arrivals(2);

  total_time = T200 + T110 + T101 + T20 + T11 + T2;
  P(2) = T2 / total_time;
  P(11) = T11 / total_time;
  P(20) = T20 / total_time;
  P(101) = T101 / total_time;
  P(110) = T110 / total_time;
  P(200) = T200 / total_time;

  current_mean1 = P(101) + P(110) + 2 * P(200);
  current_mean2 = P(11) + 2 * P(20) + P(110);
  current_mean3 = 2 * P(2) + P(11) + P(101);
```

```
   clients1(times) = current_mean1;
   clients2(times) = current_mean2;
   clients3(times) = current_mean3;

   if abs(current_mean1 - prev_mean1) < 0.00001 && abs(current_mean2 - prev_mean2) < 0.00001 &&
abs(current_mean3 - prev_mean3) < 0.00001
     sigklisi = true;
   endif

   prev_mean1 = current_mean1;
   prev_mean2 = current_mean2;
   prev_mean3 = current_mean3;
  endif
endwhile

total_steps = 1:1:times;

disp(cstrcat("The probability for state (2,0,0) is ", num2str(P(200))));
disp(cstrcat("The probability for state (1,1,0) is ", num2str(P(110))));
disp(cstrcat("The probability for state (1,0,1) is ", num2str(P(101))));
disp(cstrcat("The probability for state (0,2,0) is ", num2str(P(20))));
disp(cstrcat("The probability for state (0,1,1) is ", num2str(P(11))));
disp(cstrcat("The probability for state (0,0,2) is ", num2str(P(2))));


figure(1);
hold on;
plot(total_steps, clients1, 'r');
plot(total_steps, clients2, 'g');
plot(total_steps, clients3, 'b');
hold off;
title("Average number of clients for each queue in the system");
xlabel("Transitions in Thousands");
ylabel("Average Number of Clients for each Queue");
legend("Queue 1", "Queue 2", "Queue 3");
legend("show");
```