```
% Diego Alba - Quant. Bio - HW#4

%------------------------- Problem 1 -------------------------%
% Model of network A

% S always 1

% S -> x,   rate =k*S,  x = x+1
% S -> y,   rate =k*S,  y = y+1
% S -> z,   rate =k*S,  z = z+1
% x -> 0,   rate = gamma*x,  x = x-1
% y -> 0,   rate = gamma*y,  y = y-1
% z -> 0,   rate = gamma*z,  z = z-1
% x -> A,   rate = k*x, A = A+1 don't loose x, y or z
% y -> A,   rate = k*y, A = A+1
% z -> A,   rate = k*z, A = A+1

% stoichiometry table
stoich = [...
    1 0 0 0;...
    0 1 0 0;...
    0 0 1 0;...
    -1 0 0 0;...
    0 -1 0 0;...
    0 0 -1 0;...
    0 0 0 1;...
    0 0 0 1;...
    0 0 0 1;...
    ];

% probabilities
a1 = @(k,S,x,y,z,gamma,i,j) [k*S,k*S,k*S,...
            gamma*x(i,j),gamma*y(i,j),gamma*z(i,j),...
            k*x(i,j),k*y(i,j),k*z(i,j)];
n = 1000;

[~,~,~,tA1,~] = sim(n,stoich,a1); % custom function defined below

tfinalA = max(tA1,[],2);
avgA = mean(tfinalA) % avg. of network A
stdA = std(tfinalA) % std of network A
ratioA = stdA/avgA % ratio std/avg of network A

% Model of network B

% S always 1

% S -> x,   rate =k*S,  x = x+1
% S -> y,   rate =k*S,  y = y+1
% S -> z,   rate =k*S,  z = z+1
% x -> 0,   rate = gamma*x,  x = x-1
% y -> 0,   rate = gamma*y,  y = y-1
% z -> 0,   rate = gamma*z,  z = z-1
% x,y,z -> A,   rate = k*(x+y+z), A = A+1 don't loose x, y or z

% stoichiometry table
stoich = [...
    1 0 0 0;...
    0 1 0 0;...
    0 0 1 0;...
    -1 0 0 0;...
    0 -1 0 0;...
    0 0 -1 0;...
    0 0 0 1;...
    ];
% probabilities
a2 = @(k,S,x,y,z,gamma,i,j) [...
            k*S,k*S,k*S,...
            gamma*x(i,j),gamma*y(i,j),gamma*z(i,j),...
            k*(x(i,j)*y(i,j)*z(i,j))];
n = 1000;

[~,~,~,tB1,~] = sim(n,stoich,a2); % custom fuction defined below

tfinalB = max(tB1,[],2);
avgB = mean(tfinalB) % avg. of network B
stdB = std(tfinalB) % std of network B
ratioB = stdB/avgB % ratio std/avg of network B


%------------------------- Problem 2 -------------------------%
n = 1000;

% define the stoichiometry table and probabilities for the 4 networks
stoichA = [0 0 0 1];
probsA = @(k,S,x,y,z,gamma,i,j) k*S;

stoichB = [1 0 0 0;...
           -1 0 0 0; ...
           0 0 0  1];
probsB = @(k,S,x,y,z,gamma,i,j) [k*S,gamma*x(i,j),k*x(i,j)];

stoichC = [1 0 0 0;...
           0 1 0 0;...
           0 0 0 1;...
           -1 0 0 0;...
           0 -1 0 0];
probsC = @(k,S,x,y,z,gamma,i,j) [k*S,k*x(i,j),k*y(i,j),...
                                gamma*x(i,j),gamma*y(i,j)];

stoichD = [1 0 0 0;...
           0 1 0 0;...
           0 0 1 0;...
           0 0 0 1;...
           -1 0 0 0;...
           0 -1 0 0;...
           0 0 -1 0;];
probsD = @(k,S,x,y,z,gamma,i,j) [k*S,k*x(i,j),k*y(i,j),k*z(i,j),...

 gamma*x(i,j),gamma*y(i,j),gamma*z(i,j)];

% find the times for the 4 networks
[~,~,~,tA,~] = sim(n,stoichA,probsA);
[~,~,~,tB,~] = sim(n,stoichB,probsB);
[~,~,~,tC,~] = sim(n,stoichC,probsC);
[~,~,~,tD,~] = sim(n,stoichD,probsD);

tfinal = [max(tA,[],2),...
          max(tB,[],2),...
          max(tC,[],2),...
          max(tD,[],2)];

avg2 = mean(tfinal) % avg. of the 4 networks, 1-4
std2 = std(tfinal) % std of the 4 networks, 1-4
relative_variance = std2./avg2 % ratio of the 4 networks, 1-4

figure % plot histogram for each network
for i = 1:4
    subplot(2,2,i)
    histogram(tfinal(:,i),20)
    xlabel('Time')
    ylabel('Occurences')
    title(['Pathway ',num2str(i)])
end

%------------------------- FUNCTION -------------------------%
function [x,y,z,t,A] = sim(n,stoich,probs)
% initial values
S = 1;
x = zeros(n,1);
y = zeros(n,1);
z = zeros(n,1);
A = zeros(n,1);
t = zeros(n,1);
k = 1;
gamma = 1;

Atarget = 1;

for i = 1:n % run simulation n times
    j = 1;
    while A(i,j) < Atarget && j < 1000
        % probability of different reactions
        a = probs(k,S,x,y,z,gamma,i,j);
        a0 = sum(a);
        r = rand(1,2);

        % step length
        tau = -log(r(1))/a0;
        %reaction choice
        mu = find((cumsum(a) >= r(2)*a0), 1,'first');

        % Update time and reactants
        t(i,j+1) = t(i,j) + tau;
        x(i,j+1) = x(i,j) + stoich(mu,1);
        y(i,j+1) = y(i,j) + stoich(mu,2);
        z(i,j+1) = z(i,j) + stoich(mu,3);
        A(i,j+1) = A(i,j) + stoich(mu,4);

        j = j+1;
    end
end

% remove extra zeros from data
t(t == 0) = NaN;
t(:,1) = 0;
ck = isnan(t);
x(ck) = NaN;
y(ck) = NaN;
z(ck) = NaN;
A(ck) = NaN;
end

%------------------------- RESULTS -------------------------%
```

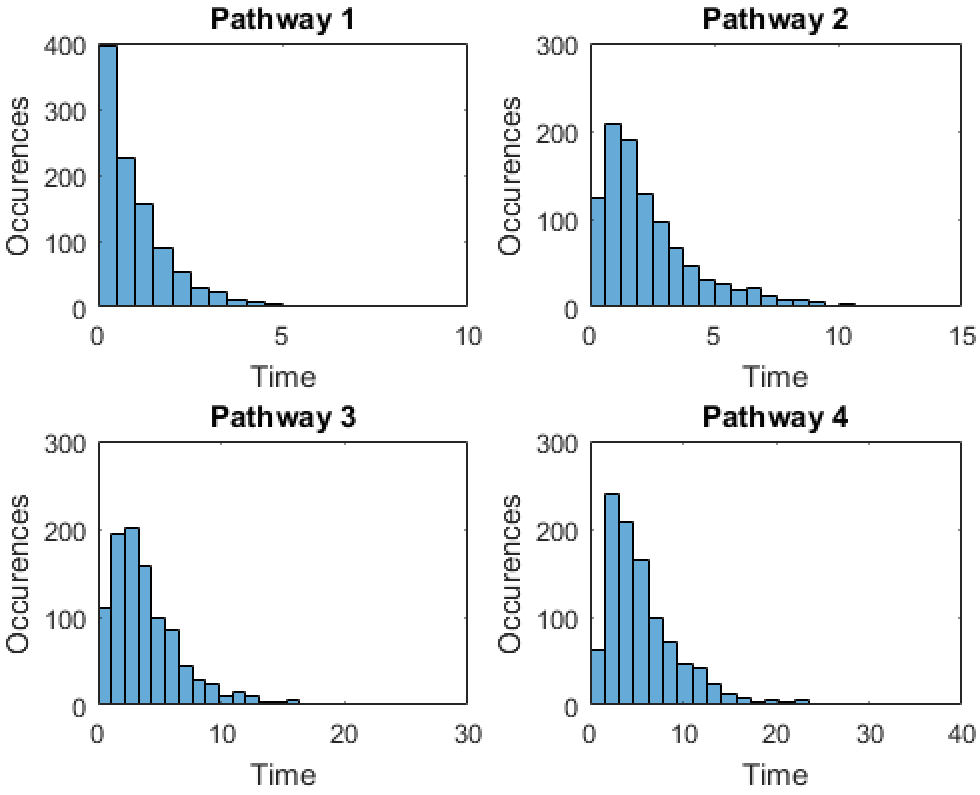
```
avgA =
    1.0327
stdA =
    0.7336
ratioA =
    0.7103
avgB =
    4.7949
stdB =
    3.6933
ratioB =
    0.7702
avg2 =
    1.0084    2.4034    3.9992    5.7068
std2 =
    1.0131    1.9810    3.0374    4.0850
relative_variance =
    1.0047    0.8243    0.7595    0.7158
```

Problem 1 B)

Network A was faster because it's OR.
The relative variance was very similar.

Problem 2 B)

Network A was the fastest, but D had the smallest relative variance



Published with MATLAB® R2017a

Problem 2 C)

The more elements the network has, the slower it becomes. The mean increases with each element and the relative variance decreases with each element. As the mean increases, the distribution shifts to the right, but a Gaussian distribution is maintained.