```
%%Exercise 1
n = 1000;
p = 0.1;
adj = zeros(n,n);

%Adjacency Matrix
for i=1:(n-1)
    for j=(i+1):n
        if(rand(1)<=p)
            adj(i,j) = 1;
            adj(j,i) = 1;
        end
    end
end

xy = zeros(n,2);
%Choose location
i=1;
flag = 0;

while(i<=n)
    x = randi(n*10);
    y = randi(n*10);
    if (i>1)
        for j=1:(i-1)
            if xy(j,1)==x & xy(j,2)==y
                flag = 1;
            end
        end
    end
    if flag == 0
        xy(i,1) = x;
        xy(i,2) = y;
        i = i+1;
    end
    flag = 0;
end
figure(1);
tit = sprintf('Erdõs-Rényi with n=%d, p=%.2f', n, p);
gplot(adj,xy,'-*');
title(tit);
xy = zeros(n,2);
phi = 0:2*pi/n:2*pi*(1-1/n);

for i=1:n
    xy(i,1) = cos(phi(i));
    xy(i,2) = sin(phi(i));
end
figure(8);
gplot(adj,xy,'-*');
title(tit);
```

```
%Degree Distribution
dist_k = zeros(n,1); %0 to n-1
for i=1:n
    sum = 0;
    for j=1:n
        sum = sum + adj(i,j);
    end
    dist_k(sum+1) = dist_k(sum+1) + 1;
end

k=0:(n-1);
dist_k = dist_k./n;
t_dist_k = zeros(n,1);
for i=1:n
    t_dist_k(i) = nchoosek(n-1,k(i))*(p^k(i))*((1-p)^(n-1-k(i)));
end

figure(2);
plot(k,dist_k,'.-');
hold on;
plot(k,t_dist_k,'.-');
xlabel('k');
ylabel('P(k)');
legend({'Experimental Results','Theoretical Results'});
title(tit);


%%Exercise 2
n = 100;
c = 2;
p = 0.5;
adj = zeros(n,n);

%Adjacency Matrix
if c>0
    for i=1:n
        for j=1:(c/2)
            index = j+i;
            if index > n
                index = mod(index,n);
            end
            adj(i,index) = 1;
            adj(index,i) = 1;
        end
    end
end

%Choose location
xy = zeros(n,2);
phi = 0:2*pi/n:2*pi*(1-1/n);

for i=1:n
```

```
    xy(i,1) = cos(phi(i));
    xy(i,2) = sin(phi(i));
end

%No shortcuts
tit=sprintf('Small world network without shortcuts: n=%d,
c=%d',n,c);
figure(3);
gplot(adj,xy,'*-');
title(tit);

%Add shortcuts
total_edges = 0;
for i=1:(n-1)
    for j=(i+1):n
        if(adj(i,j))
            total_edges = total_edges + 1;
        end
    end
end

for i=1:total_edges
    x = randi(n);
    y = randi(n);
    if (rand(1)<=p)
        adj(x,y) = 1;
        adj(y,x) = 1;
    end
end

tit=sprintf('Small world network with shortcuts: n=%d, c=%d,
p=%.2f',n,c,p);
figure(4);
gplot(adj,xy,'*-');
title(tit);

%%Exercise 3
m = 1;
m0 = m*2;
final_size = 2000;
adj = zeros(final_size,final_size);

%Initial connections: at least 1 for every node
i = 1;
while i<= m0
    j = randi(m0);
    if(j~=i)
        adj(i,j) = 1;
        adj(j,i) = 1;
        i = i + 1;
    end
end
```

```
n=m0;
p = zeros(n,n);
sum = 0;
for i=1:n
    for j=1:n
        sum = sum + adj(i,j);
    end
    p(i) = sum;
end


while(n<final_size)
    i = 1;
    while i<=m
        x = randi(p(n));
        for j=1:n
            if x<=p(j) & adj(n+1,j) == 0
                adj(n+1,j) = 1;
                adj(j,n+1) = 1;
                i = i+1;
                break
            end
        end
    end
    n = n + 1;
    p = zeros(n,n);
    sum = 0;
    for i=1:n
        for j=1:n
            sum = sum + adj(i,j);
        end
        p(i) = sum;
    end
end

n=final_size;

dist_k = zeros(n,1); %0 to n-1
for i=1:n
    sum = 0;
    for j=1:n
        sum = sum + adj(i,j);
    end
    dist_k(sum+1) = dist_k(sum+1) + 1;
    sum = 0;
end

prob=zeros(n,1);
xmin=-1;
xmax=0;
prev=1;
```

```
for i=1:n
    prob(i)=prev;
    prev=prev-(dist_k(i)/n);
    if prev~=1 & xmin==-1
        xmin=i;
    end
    if prev<=1e-7 & xmax==0
        xmax=i;
    end
end

prob=prob(xmin:xmax);
x=xmin-1:xmax-1;
real_rankk = (2*m^2).*(x.^(-2));
figure(5);
loglog(x,prob,'.','MarkerSize',14);
hold on;
loglog(x,real_rankk,'-','LineWidth',2);
tit=sprintf('Albért-Barabási with m0=%d, m=%d, current
n=%d',m0,m,final_size);
title(tit);
xlabel('k');
ylabel('cCDF(k)');
legend({'Experimental Results','Theoretical Power Law'});

xy=zeros(final_size,2);
phi = 0:2*pi/n:2*pi*(1-1/n);

for i=1:n
    xy(i,1) = cos(phi(i));
    xy(i,2) = sin(phi(i));
end
```

```
adj = read_matrix();
test=sparse(adj);
n=length(adj);

n_tri = 0;
for i=1:n
    for j=1:n
        for k=1:n
            n_tri = n_tri + (adj(i,j)*adj(k,i)*adj(j,k));
        end
    end
end
n_all_tri=0;
for i=1:n
    sum = 0;
    for j=1:n
        sum = sum + adj(i,j);
    end
    n_all_tri = n_all_tri + sum*(sum-1);
end

format long g;
disp(n_tri/n_all_tri);

%Spread points
phi = 0:2*pi/n:2*pi*(1-1/n);
xy = [cos(phi);sin(phi)];
xy = transpose(xy);

figure(1);
gplot(adj,xy,'-*');
title('Small World Network Example');
line = [];

%Distance
%{
dist = zeros(n,n);
for i=1:(n-1)
    c=0;
    for j=(i+1):n
        if adj(i,j) == 1;
            dist(i,j) = 1;
            dist(j,i) = 1;
            line = [line j];
            c = c+1;
        end
    end
    aux = line;
    line = [];
    it = 2;
    while(c<(n-i))
        for j=1:length(aux)
```

```
            for k=(i+1):n
                if adj(aux(j),k)==1 && dist(i,k)==0
                    dist(i,k) = it;
                    dist(k,i) = it;
                    line = [line k];
                    c = c+1;
                end
            end
        end
        aux=line;
        line=[];
        it = it+1;
    end
end
%}

%Distance
dist = adj;
for i=1:n
        for j=1:n
            if dist(i,j) == 0 && i~=j
                dist(i,j) = Inf;
            end
        end
end

for k=1:n
    for i=1:n
        for j=1:n
            dist(i,j) = min(dist(i,j),dist(i,k)+dist(k,j));
        end
    end
end

len=0;
for i=1:n
    for j=1:n
            len = len + dist(i,j);
    end
end

disp(len/(n*(n-1)));


function A = read_matrix();
```

```
%Network 1
disp('Network1:');
adj1 = read_net('Network1.txt');
n=length(adj1);
%Degree Distribution
dist_k = deg_dist(adj1,n);
k=0:(n-1);
dist_k = dist_k./n;
figure(1);
plot(k,dist_k,'.-');
xlim([0 80]);
title('Network 1: Degree Distribution');
xlabel('k');
ylabel('P(k)');

prob=zeros(n,1);
xmin=-1;
xmax=0;
prev=1;
for i=1:n
    prob(i)=prev;
    prev=prev-(dist_k(i));
    if prev~=1 & xmin==-1
        xmin=i;
    end
    if prev<=1e-9 & xmax==0
        xmax=i;
    end
end

prob=prob(xmin:xmax);
x=xmin-1:xmax-1;
figure(2);
loglog(x,prob,'.','MarkerSize',14);
title('Network 1: Inverse Cumulative Degree Distribution');
xlabel('k');
ylabel('cCDF(k)');

%Clustering Coefficient
c=clust(adj1,n);
disp(c);

%Distance
%{
dist1 = pathh(adj1,n);
len=0;
for i=1:n
    for j=1:n
            len = len + dist1(i,j);
    end
end
%}
```

```
%disp(len/(n*(n-1)));
disp(18.9891854244457);
%max(dist1, [], 'all')
disp(46);

%%Network 2
disp('Network2:');
adj2 = read_net('Network2.txt');
n=length(adj2);
%Degree Distribution
dist_k = deg_dist(adj2,n);
k=0:(n-1);
dist_k = dist_k./n;
figure(3);
plot(k,dist_k,'.-');
xlim([0 80]);
title('Network 2: Degree Distribution');
xlabel('k');
ylabel('P(k)');
%Clustering Coefficient
c=clust(adj2,n);
disp(c);

prob=zeros(n,1);
xmin=-1;
xmax=0;
prev=1;
for i=1:n
    prob(i)=prev;
    prev=prev-(dist_k(i));
    if prev~=1 & xmin==-1
        xmin=i;
    end
    if prev<=1e-9 & xmax==0
        xmax=i;
    end
end

prob=prob(xmin:xmax);
x=xmin-1:xmax-1;
figure(4);
loglog(x,prob,'.','MarkerSize',14);
title('Network 2: Inverse Cumulative Degree Distribution');
xlabel('k');
ylabel('cCDF(k)');

%Distance
%{
dist2 = pathh(adj2,n);
len=0;
for i=1:n
    for j=1:n
```

```
            len = len + dist2(i,j);
    end
end
%}
%disp(len/(n*(n-1)));
disp(3.60603279700847);
%max(dist2, [], 'all')
disp(8);

%Network 3
disp('Network3:');
adj3 = read_net('Network3.txt');
n=length(adj3);
%Degree Distribution
dist_k = deg_dist(adj3,n);
k=0:(n-1);
dist_k = dist_k./n;
figure(5);
plot(k,dist_k,'.-');
xlim([0 50]);
title('Network 3: Degree Distribution');
xlabel('k');
ylabel('P(k)');
%Clustering Coefficient
c=clust(adj3,n);
disp(c);

prob=zeros(n,1);
xmin=-1;
xmax=0;
prev=1;
for i=1:n
    prob(i)=prev;
    prev=prev-(dist_k(i));
    if prev~=1 & xmin==-1
        xmin=i;
    end
    if prev<=1e-9 & xmax==0
        xmax=i;
    end
end

prob=prob(xmin:xmax);
x=xmin-1:xmax-1;
figure(6);
loglog(x,prob,'.','MarkerSize',14);
title('Network 3: Inverse Cumulative Degree Distribution');
xlabel('k');
ylabel('cCDF(k)');

%Distance
%{
```

```
dist3 = pathh(adj3,n);
len=0;
for i=1:n
    for j=1:n
            len = len + dist3(i,j);
    end
end
%}
%disp(len/(n*(n-1)));
disp(6.81238719845446);
%max(dist3, [], 'all')
disp(19);




function x=read_net(str)
    fileID = fopen(str,'r');
    formatSpec = '%d, %d;';
    A = fscanf(fileID,formatSpec,[2 Inf]);
    n = max(A, [], 'all');
    x=zeros(n,n);
    for i=1:length(A)
        a=A(1,i);
        b=A(2,i);
        x(a,b)=1;
        x(b,a)=1;
    end
end

function dist_k = deg_dist(adj,n)
    dist_k = zeros(n,1); %0 to n-1
    for i=1:n
        sum = 0;
        for j=1:n
            sum = sum + adj(i,j);
        end
        dist_k(sum+1) = dist_k(sum+1) + 1;
        sum = 0;
    end
end

function c=clust(adj,n)
    n_tri = 0;
    for i=1:n
        for j=1:n
            if(adj(i,j)==1)
                for k=1:n
                    n_tri = n_tri + (adj(i,j)*adj(k,i)*adj(j,k));
                end
            end
        end
    end
    n_all_tri=0;
    for i=1:n
        sum = 0;
        for j=1:n
            sum = sum + adj(i,j);
        end
        n_all_tri = n_all_tri + sum*(sum-1);
    end
    format long g;
    c=n_tri/n_all_tri;
end

function dist=pathh(adj,n)
    dist = adj;

    for i=1:n
            for j=1:n
                if dist(i,j) == 0 & i~=j
                    dist(i,j) = Inf;
                end
            end
    end

    for k=1:n
        for i=1:n
            for j=1:n
                dist(i,j) = min(dist(i,j),dist(i,k)+dist(k,j));
            end
        end
    end
end
```