MATLAB CODE:

```
%% max_dist_K_mean_test.m
%  author: mrinmoy sarkar
%  email:msarkar@aggies.ncat.edu

clear;
close all;
% data
x = [0 0 5 5 4 1;
     0 1 4 5 5 0];

%% plot data
figure;
plot(x(1,:),x(2,:),'or');
xlabel('x1');
ylabel('x2');
axis([-1 6 -1 6])
title('visualize of data points')



%% Maximum-Distance algorithm
[z,L] = maximumDistanceAlgorithm(x);
figure
noOfClass = size(z,2);
classes = cell(1,noOfClass);
indicator={'g>','b<'};
for i=1:noOfClass
    index = find(L==i);
    data = x(:,index);
    classes{1,i} = data;
    %plot different cluster member with different color
    hold on;
    plot(data(1,:),data(2,:),indicator{i},'MarkerSize',10);
end

%indicate cluster centre
plot(z(1,:),z(2,:),'ko','MarkerSize',15);
axis([-1 6 -1 6])
xlabel('x1');
ylabel('x2');
legend('cluster 1','cluster 2','cluster center');
title('clustering using Maximum-Distance Algorithms')


%% K-Means Algorithm
k = 2;
[z, classes] = kmeanAlgorithm(x,k);
figure
for i=1:k
    data = classes{1,i};
    %plot different cluster member with different color
    hold on;
    plot(data(1,:),data(2,:),indicator{i},'MarkerSize',10);
```

```
end

%indicate cluster centre
plot(z(1,:),z(2,:),'k*','MarkerSize',15);
axis([-1 6 -1 6])
xlabel('x1');
ylabel('x2');
legend('cluster 1','cluster 2','cluster center');
title('clustering using K-Means Algorithms')
```

For 3.6:

```
function [z,L] = maximumDistanceAlgorithm(x)
z = x(:,1);
dist = 0;
xx=x;
x=x(:,2:end);
while 1
    n = size(x,2);
    distances = zeros(1,n);
    for i=1:n
        c = size(z,2);
        p = ones(size(z,1),c).*x(:,i);
        d = z - p;
        d = sum(d.^2);
        d = sqrt(d);
        distances(i) = min(d);
    end

    [dn, dni] = max(distances);
    if dist ~= 0 && dn > dist
        temp = ones(size(z)).*x(:,dni);
        dist =.5 * mean(sqrt(sum((z-temp).^2)));
        z = [z x(:,dni)];
        if size(x,2) == 1
            break;
        elseif dni == 1
            x=x(:,2:end);
        elseif dni == size(x,2)
            x=x(:,1:dni-1);
        else
            x=[x(:,1:dni-1) x(:,dni+1:end)];
        end
    elseif dist ~= 0 && dn < dist
        break;
    else
        z = [z x(:,dni)];
        dist = sqrt(sum((z(:,2) - z(:,1)).^2))/2;
        if size(x,2) == 1
            break;
        elseif dni == 1
            x=x(:,2:end);
        elseif dni == size(x,2)
            x=x(:,1:dni-1);
        else
```

```
            x=[x(:,1:dni-1) x(:,dni+1:end)];
        end
    end
end
L = zeros(1,size(xx,2));
for i= 1:size(xx,2)
    temp = ones(size(z)).*xx(:,i);
    [mi, L(i)] = min(sqrt(sum((z-temp).^2)));
end
end
```

For 3.7:

```
function [z,classes] = kmeanAlgorithm(x,k)
classes = cell(1,k);
z=zeros(size(x,1),k);
for i=1:k
    z(:,i) = x(:,i);
    classes{1,i}=[];
end

while 1
    for i=1:size(x,2)
        temp = ones(size(z)).*x(:,i);
        [m mi] = min(sqrt(sum((z-temp).^2)));
        classes{1,mi} = [classes{1,mi} x(:,i)];
    end
    zNew = zeros(size(z));
    for i=1:k
        temp = classes{1,i};
        zNew(:,i) = (1/size(temp,2))*sum(temp,2);
    end
    if sum(sum(z-zNew)) == 0
        break;
    else
        z=zNew;
    end
    for i=1:k
        classes{1,i}=[];
    end
end
end
```

PLOT:

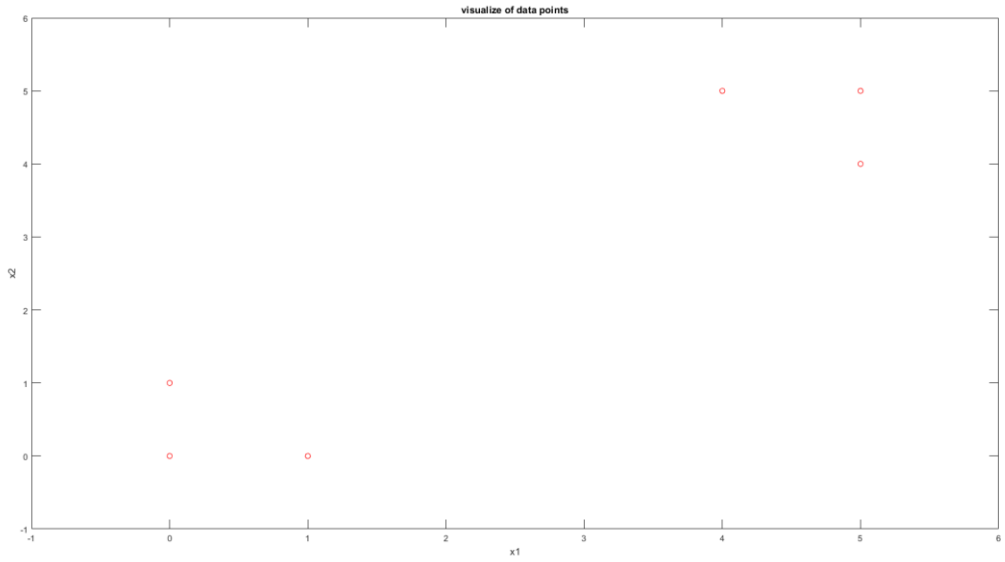

Figure 1: Data visualization

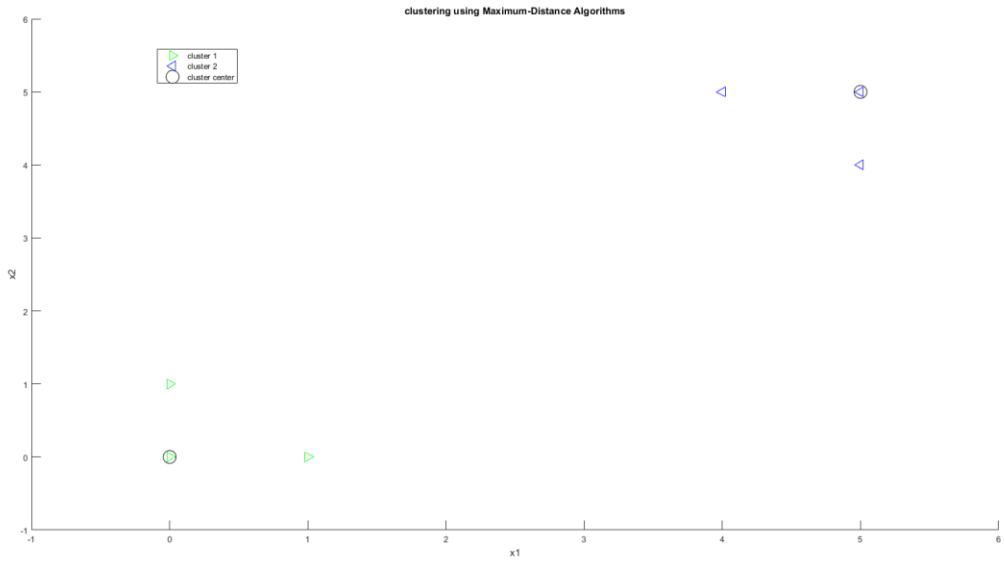

Figure 2: Clustering Using Maximum Distance Algorithms

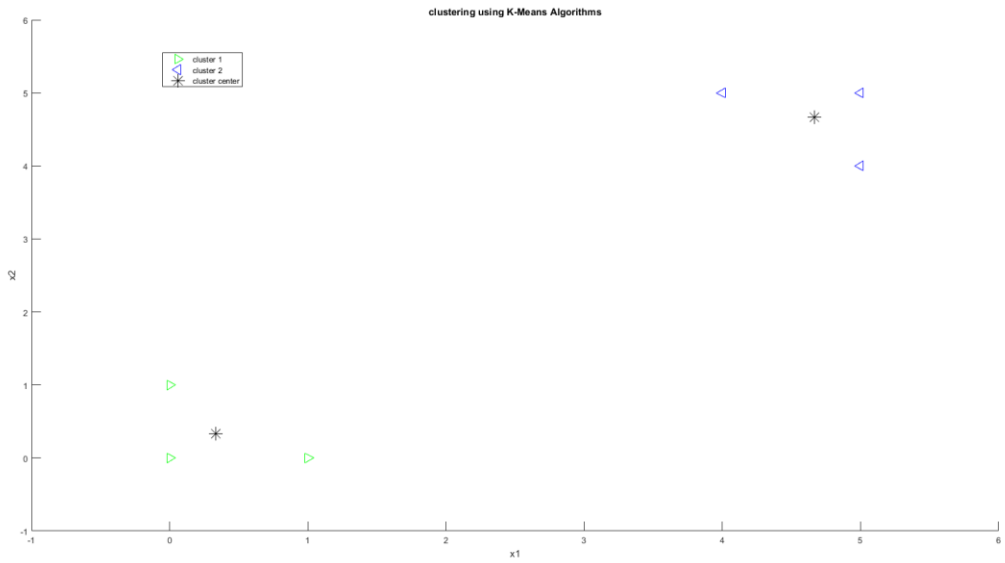

Figure 3: Clustering Using K-Means Algorithms