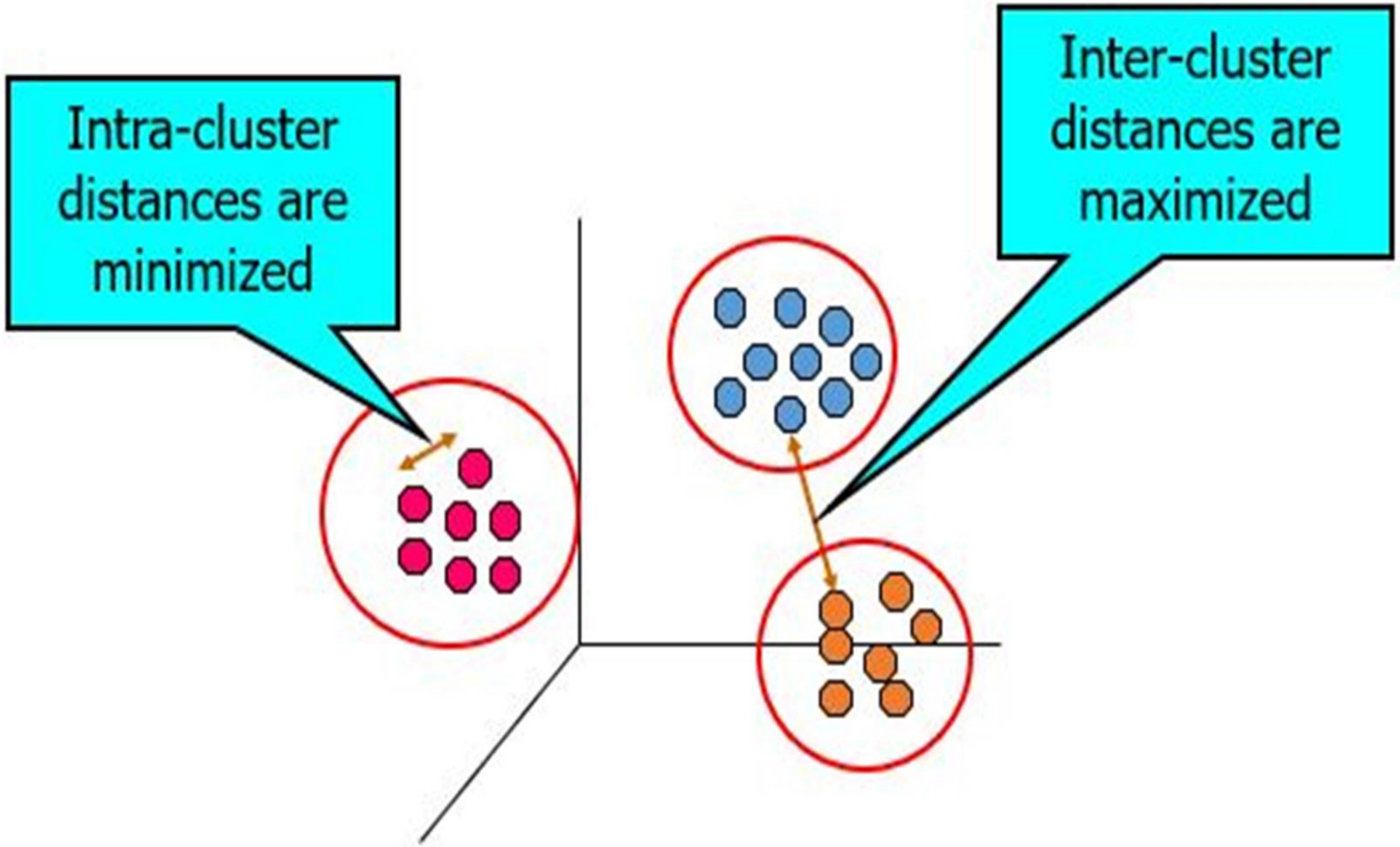
CLUSTERING

Partitioning methods

Cluster Analysis

CLUSTER ANALYSIS : FINDING GROUPS OF OBJECTS SUCH THAT THE OBJECTS IN A GROUP WILL BE SIMILAR (OR RELATED) TO ONE ANOTHER AN DIFFERENT FROM (OR UNRELATED TO) THE OBJECTS IN OTHER GROUPS.


Intra-cluster distances are minimized
Inter-cluster distances are maximized

PARTITIONING METHODS

- The general criterion of a good partitioning is that objects in the same cluster are “close” or related to each other, whereas objects of different clusters are “far apart” or very different.

PARTITIONING METHODS

- Popular heuristic methods, such as

(1) the ***k-means algorithm***, where each cluster is represented by the mean value of the objects in the cluster, and

(2) the ***k-medoids algorithm***, *where each cluster is represented by one of the objects* located near the center of the cluster

CENTROID-BASED TECHNIQUE: THE *K-MEANS* METHOD

- The *k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intra cluster similarity is high but the inter cluster similarity* is low.

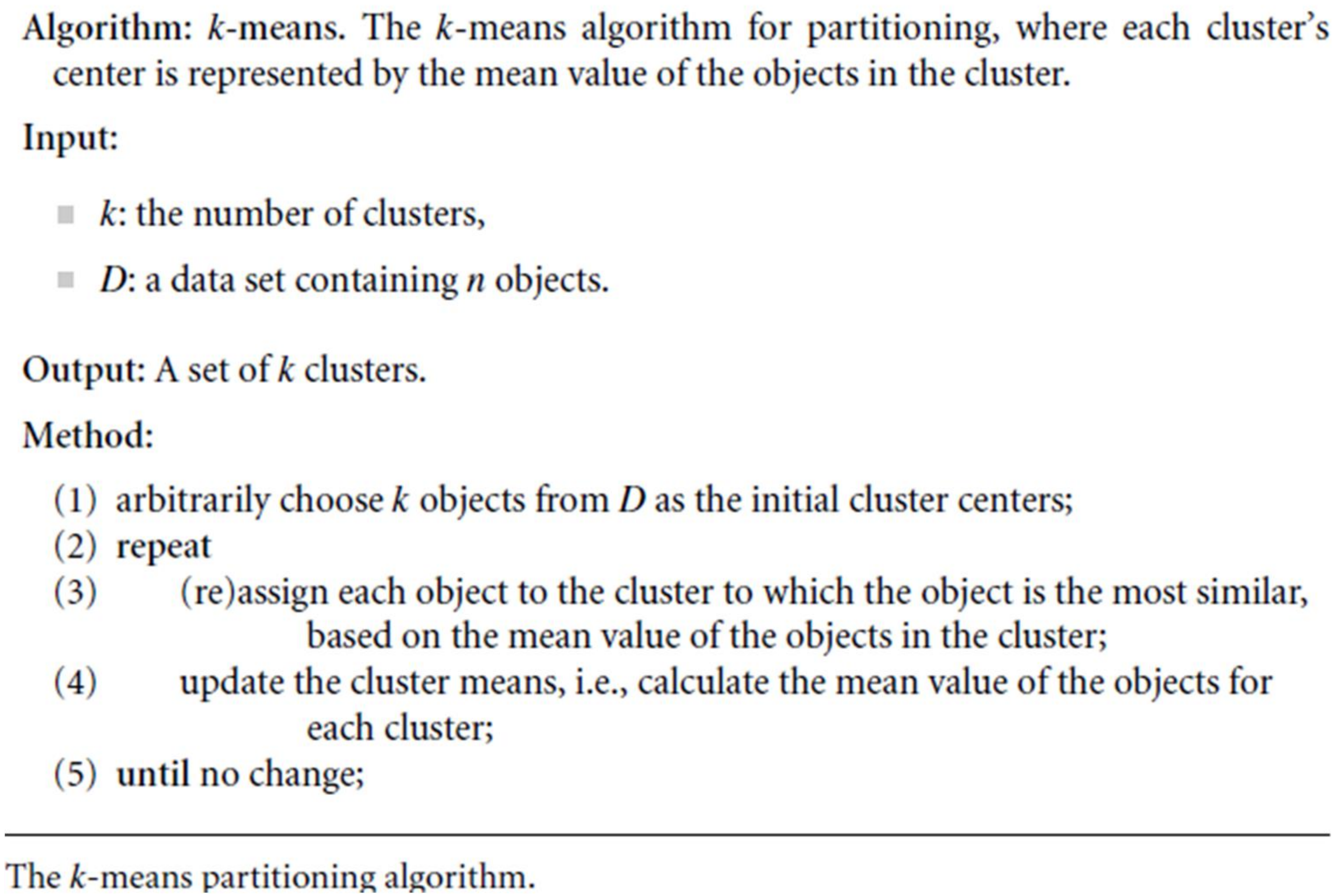
- Cluster similarity is measured in regard to the *mean value of the objects in a cluster*

CENTROID-BASED TECHNIQUE: THE *K*-*MEANS* *METHOD*

- First, it randomly selects *k of the objects, each of which initially represents a cluster mean* or center.

- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.

CENTROID-BASED TECHNIQUE: THE *K*-*MEANS METHOD*



Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

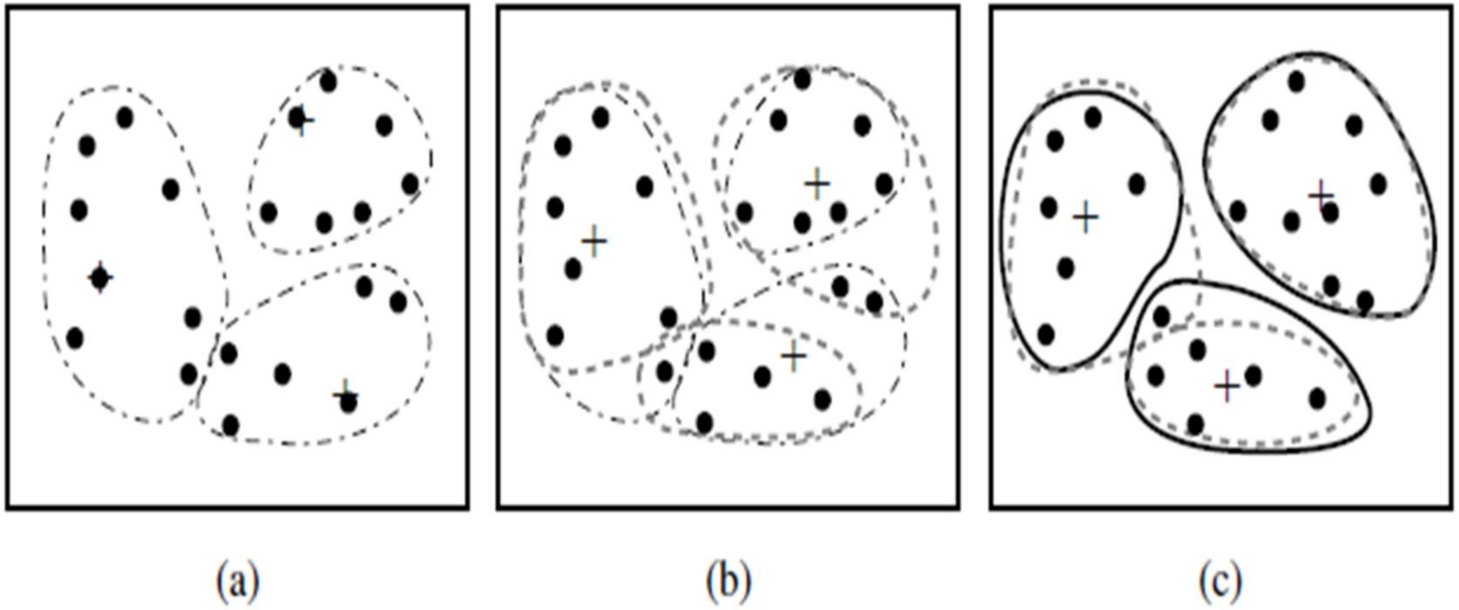
Output: A set of *k* clusters.

Method:

```
(1) arbitrarily choose $k$ objects from $D$ as the initial cluster centers;
(2) repeat
(3)       (re)assign each object to the cluster to which the object is the most similar,
                based on the mean value of the objects in the cluster;
(4)       update the cluster means, i.e., calculate the mean value of the objects for
                each cluster;
(5) until no change;
```

The *k*-means partitioning algorithm.

CENTROID-BASED TECHNIQUE: THE *K-MEANS* METHOD



Clustering of a set of objects based on the *k*-means method. (The mean of each cluster is marked by a "+".)

REPRESENTATIVE OBJECT-BASED TECHNIQUE: THE *K-MEDOIDS METHOD*

- K-Medoids (also called as Partitioning Around Medoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw.

- A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.

- The *k-means algorithm is sensitive to outliers because an object with an extremely large* value may substantially distort the distribution of data.

- Instead of taking the mean value of the objects in a cluster as a reference point, we can pick actual objects to represent the clusters, using one representative object per cluster.
- Each remaining object is clustered with the representative object to which it is the most similar.
- The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point

- That is, an absolute-error criterion is used, defined as

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|,$$

where *E is the sum of the absolute error for all objects in the data set;*

- ***p is the point in*** space representing a given object in cluster *Cj;*
- ***oj is the representative object of Cj.***

- ***In*** general, the algorithm iterates until, eventually, each representative object is actually the medoid, or most centrally located object, of its cluster.

- Case 1:

  p currently belongs to representative object, oj. If oj is replaced by o random as a representative object and ***p is closest to one of the other representative objects, oi,*** *i not equal j, then* ***p is reassigned to oi.***

- Case 2:

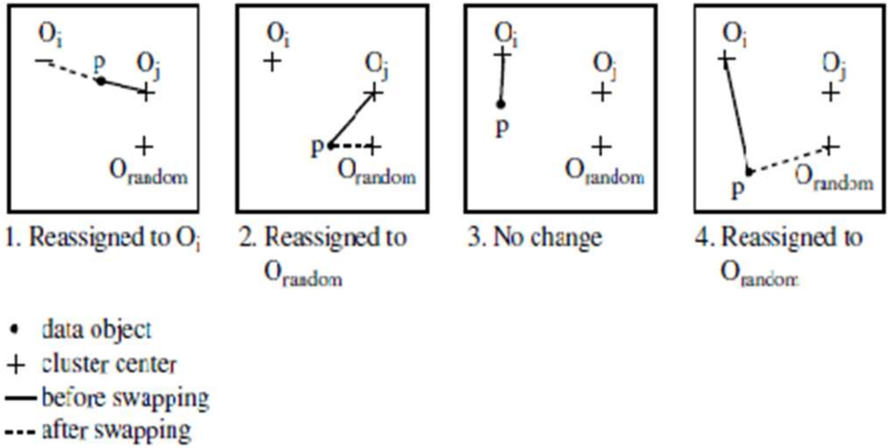
  p currently belongs to representative object, oj. If oj is replaced by o random as a representative object and ***p is closest to o random, then p is reassigned to o random.***

- Case 3:

  p currently belongs to representative object, oi, i not equal j. If oj is replaced by o random as a representative object and ***p is still closest to oi, then the assignment does not*** change.

- Case 4:

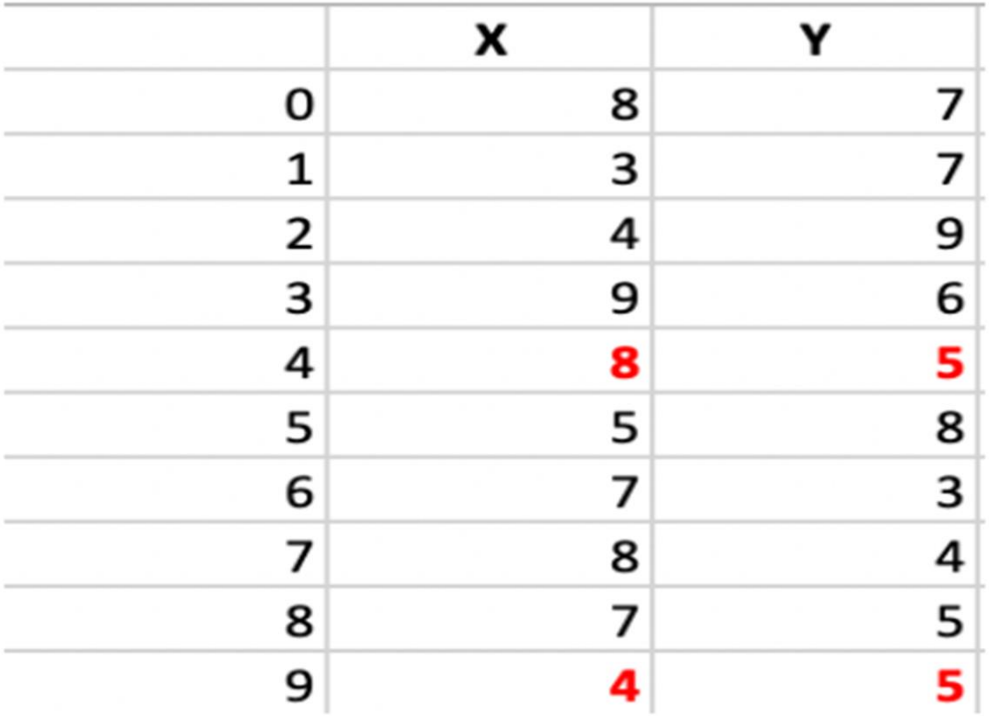
  p currently belongs to representative object, oi, i not equal j. If oj is replaced by o random as a representative object and p is closest to o random, then p is reassigned to ***o random.***



4 Four cases of the cost function for *k*-medoids clustering.

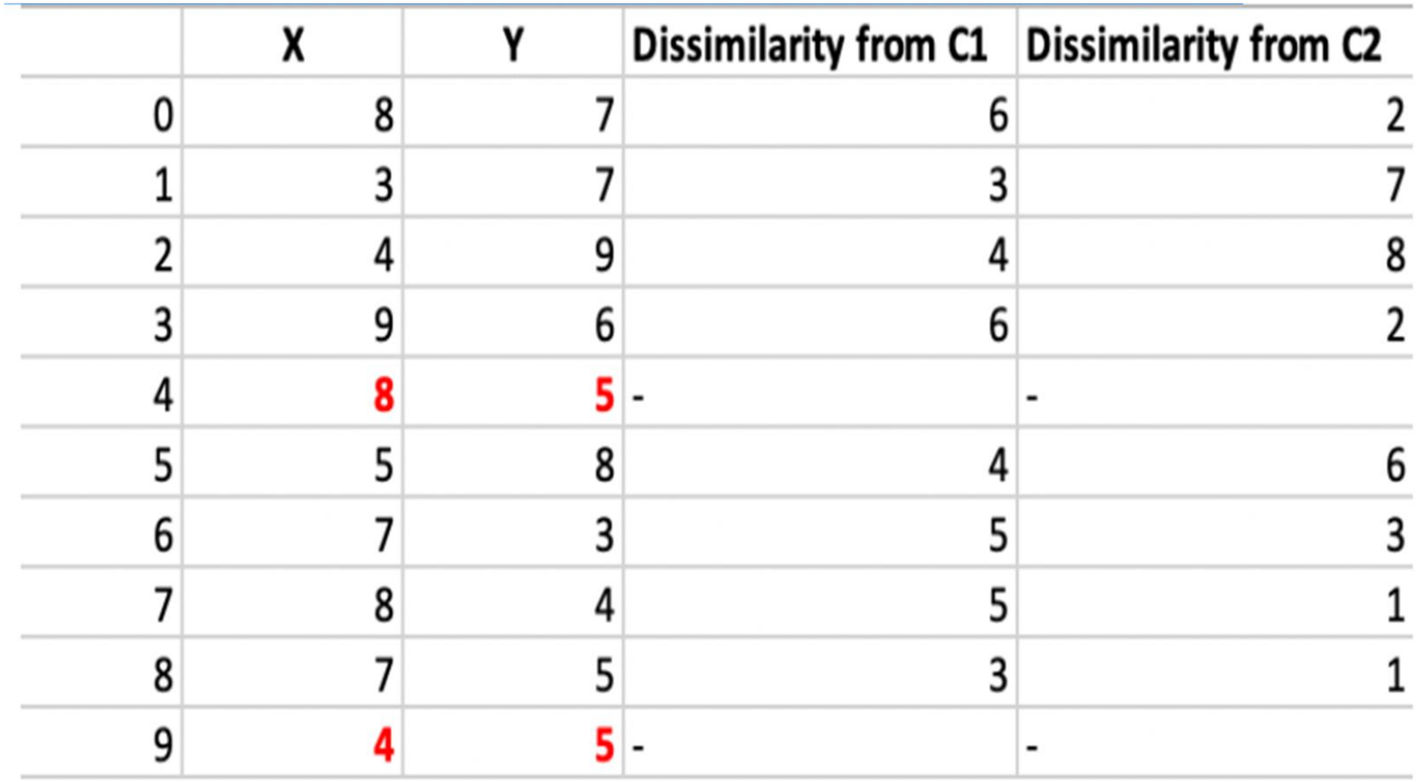
K-MEDOIDS CLUSTERING WITH SOLVED EXAMPLE

Let’s consider the following example:



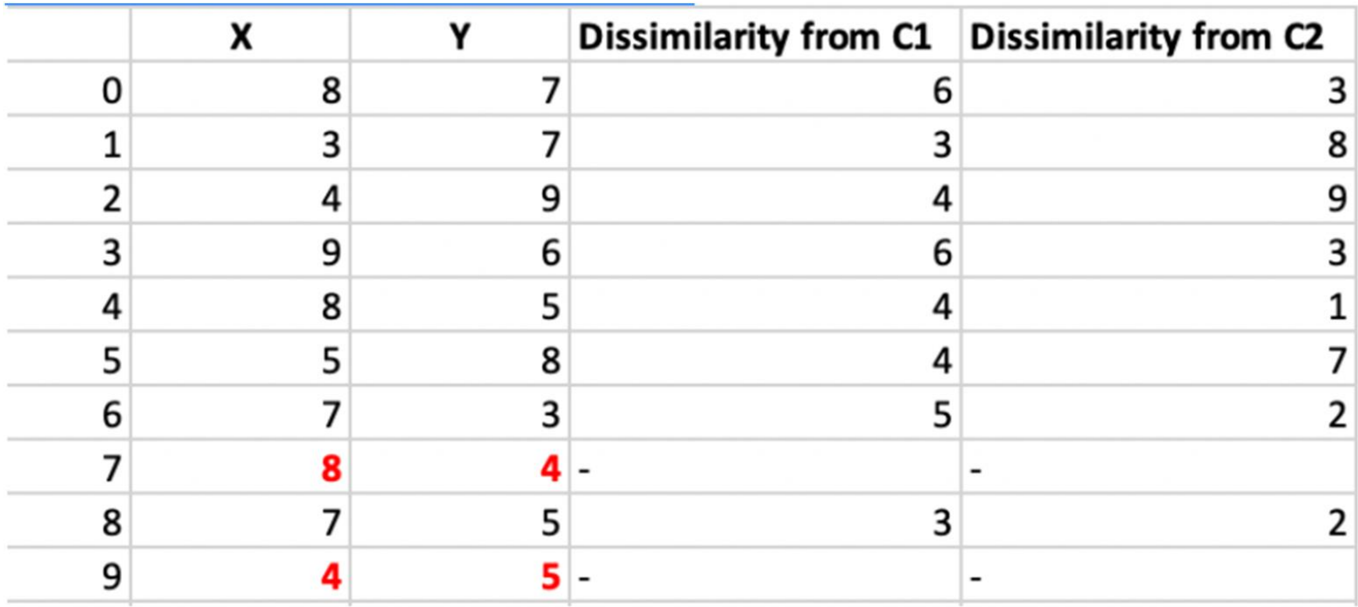
	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5

- **Step 1:**
  Let the randomly selected 2 medoids, so select k = 2 and let **C1 -(4, 5)** and **C2 -(8, 5)** are the two medoids.
- **Step 2: Calculating cost.**
  The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:



	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	8	5	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	4	5	-	-

- Each point is assigned to the cluster of that medoid whose dissimilarity is less.
  The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.
  The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20
- **Step 3: randomly select one non-medoid point and recalculate the cost.**
  Let the randomly selected point be (8, 4). The dissimilarity of each non-medoid point with the medoids – C1 (4, 5) and C2 (8, 4) is calculated and tabulated.
-

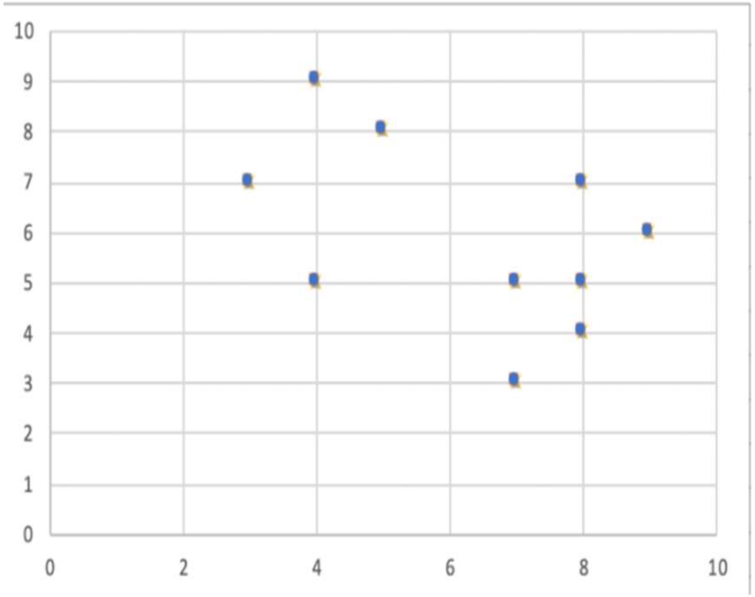


	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	8	4	-	-
8	7	5	3	2
9	4	5	-	-

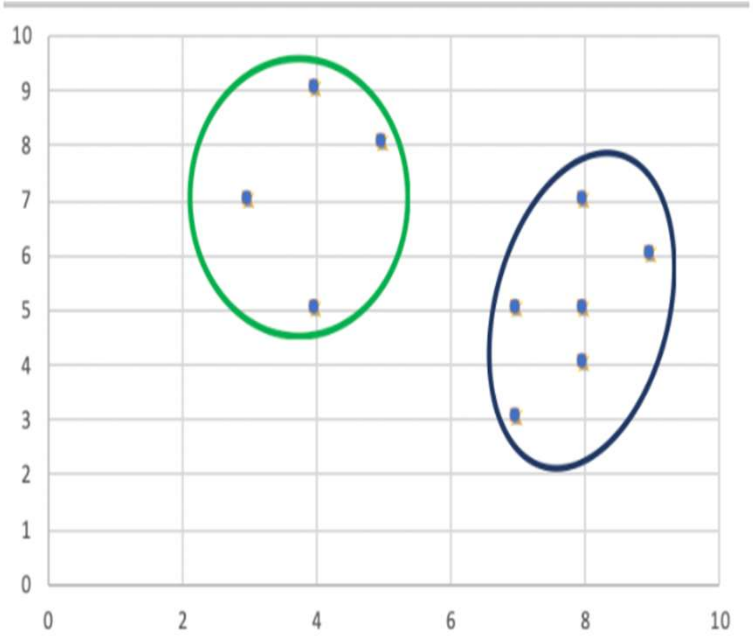
- Each point is assigned to that cluster whose dissimilarity is less. So, the points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.
  The New cost = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22
  Swap Cost = New Cost – Previous Cost = 22 – 20 and **2 >0**
- As the swap cost is not less than zero, we undo the swap. Hence (3, 4) and (7, 4) are the final medoids.

If a graph is drawn using the given data points, we obtain the following:

The clustering would be in the following way



10
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10



10
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10