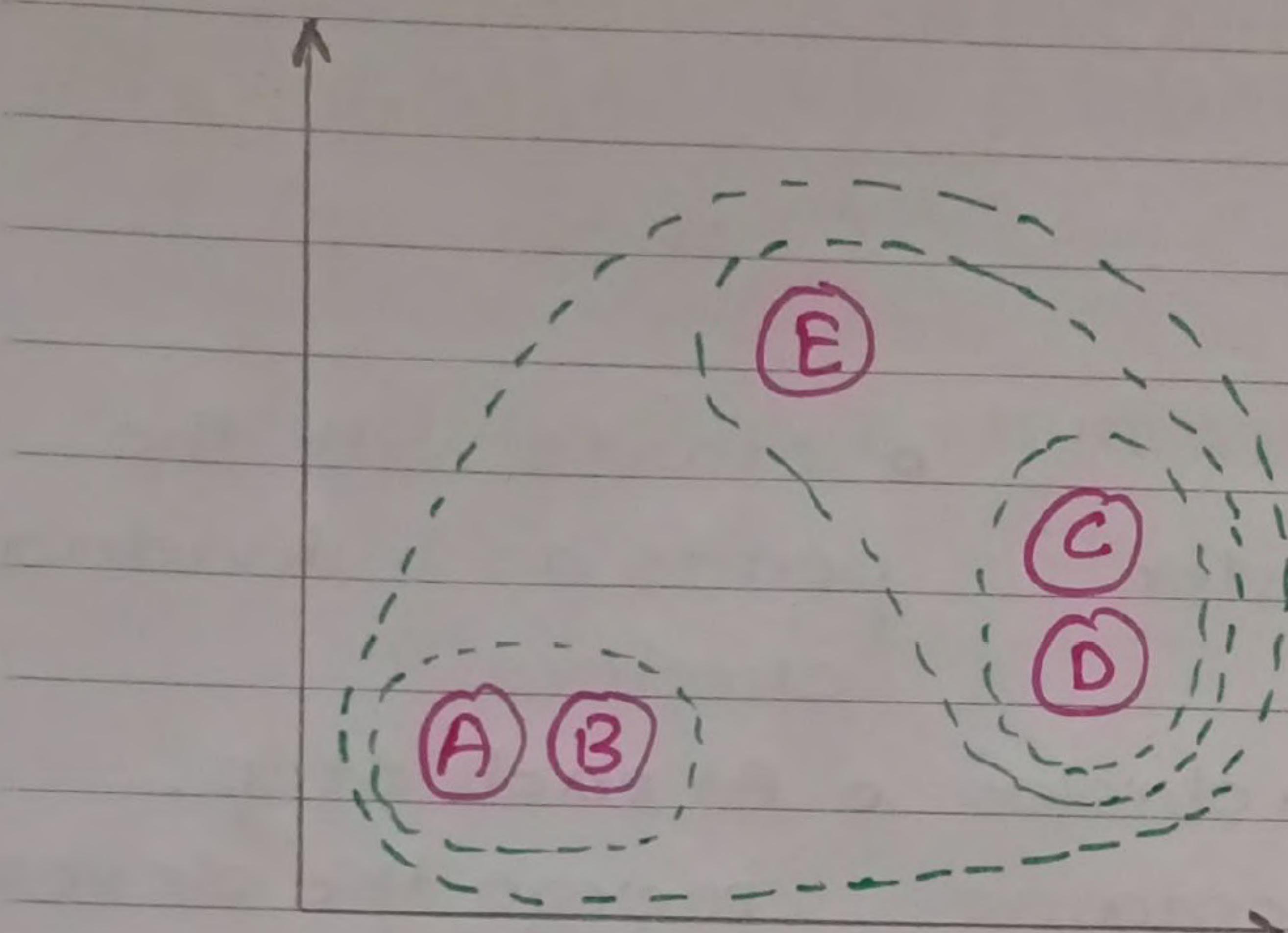


Hierarchical Clustering

Suppose we have 5 points

(A) (B) (C) (D) (E)



Hierarchical clustering is a popular method for grouping objects.

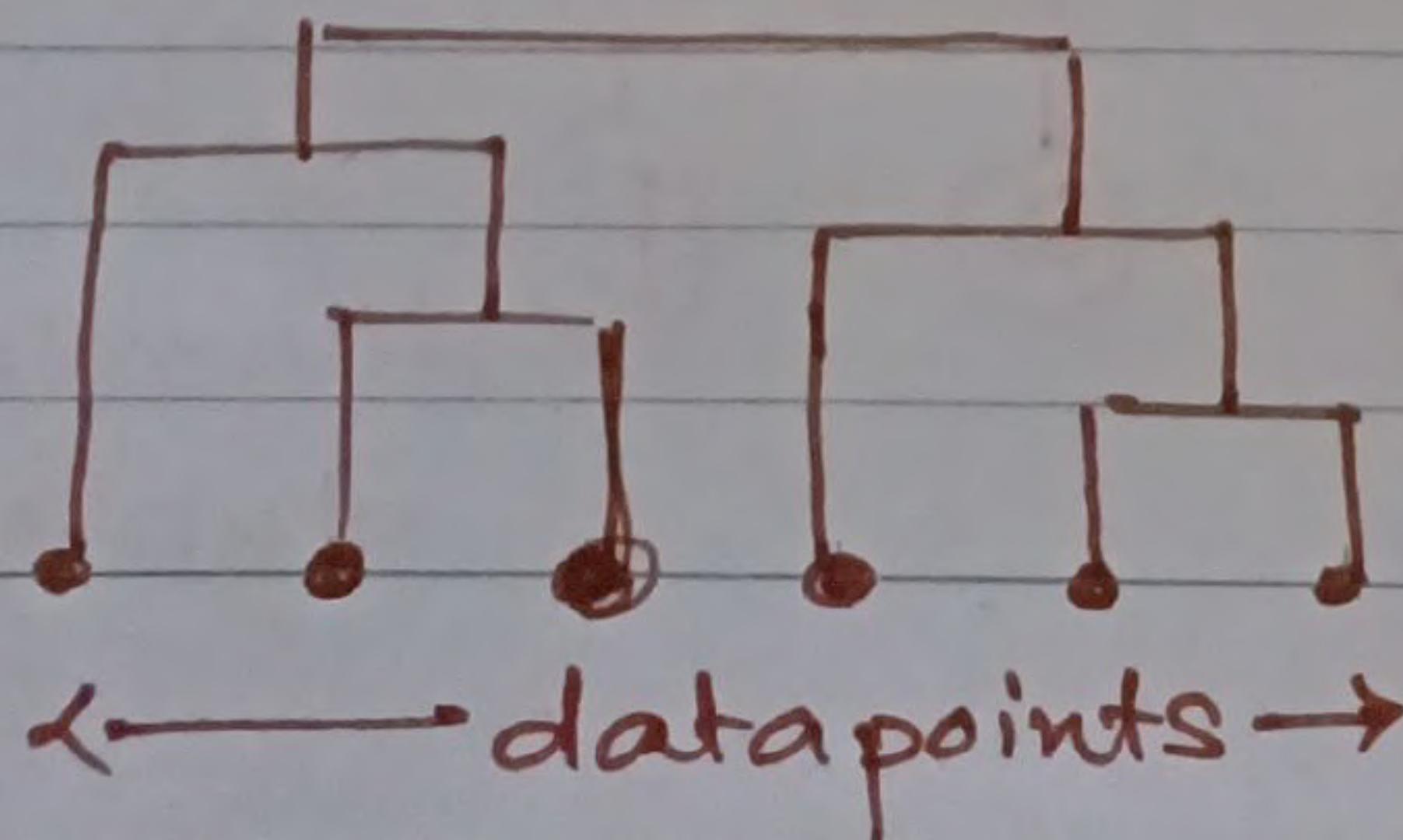
It creates groups so that objects within a group are similar to

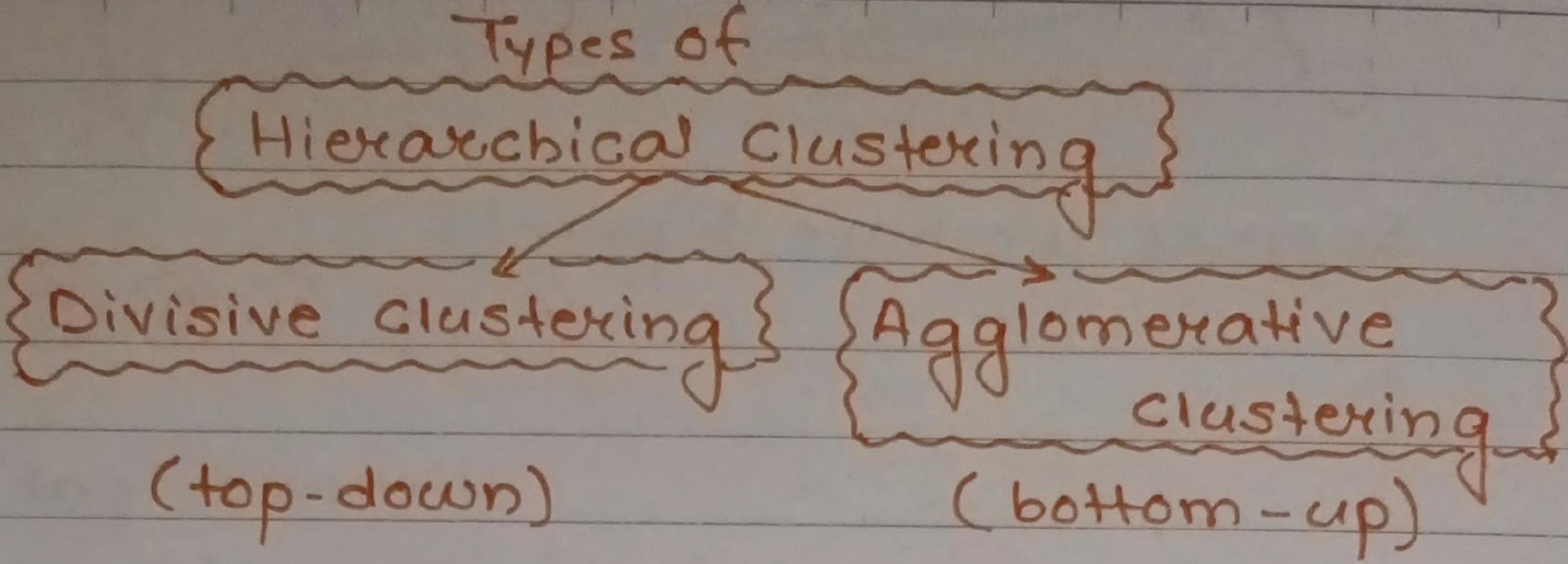
each other and different from objects in other groups.

Clusters are visually represented in a hierarchical tree called **dendrogram**.

The sole concept of Hierarchical clustering lies in just the construction and analysis of a dendrogram.

A dendrogram is a tree-like structure that explains the relationship between all the data points in the system.

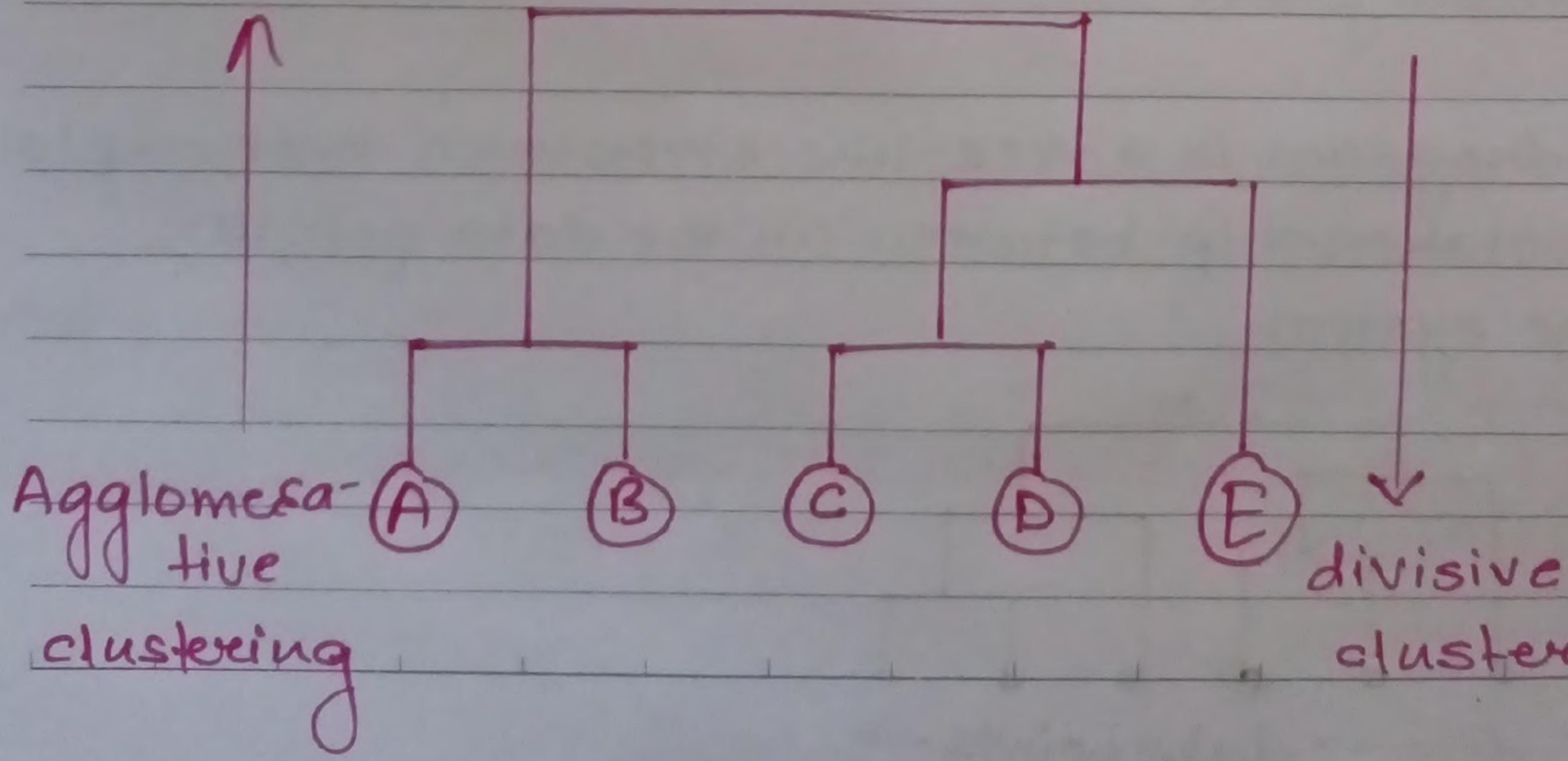




- Start with one, all-inclusive cluster
- At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Starts with the points as individual clusters.
- At each step, merge the closest pair of clusters until only one cluster left.

Traditional Hierarchical algorithms use a similarity or distance matrix.

→ merge or split one cluster at a time.



Strengths of Hierarchical Clustering.

- o Do not have to assume any particular number of clusters
 - Any desired number of clusters can be obtained by "cutting" the dendrogram at the proper level.
- o They may correspond to meaningful taxonomies
 - Example in biological sciences

Agglomerative Clustering

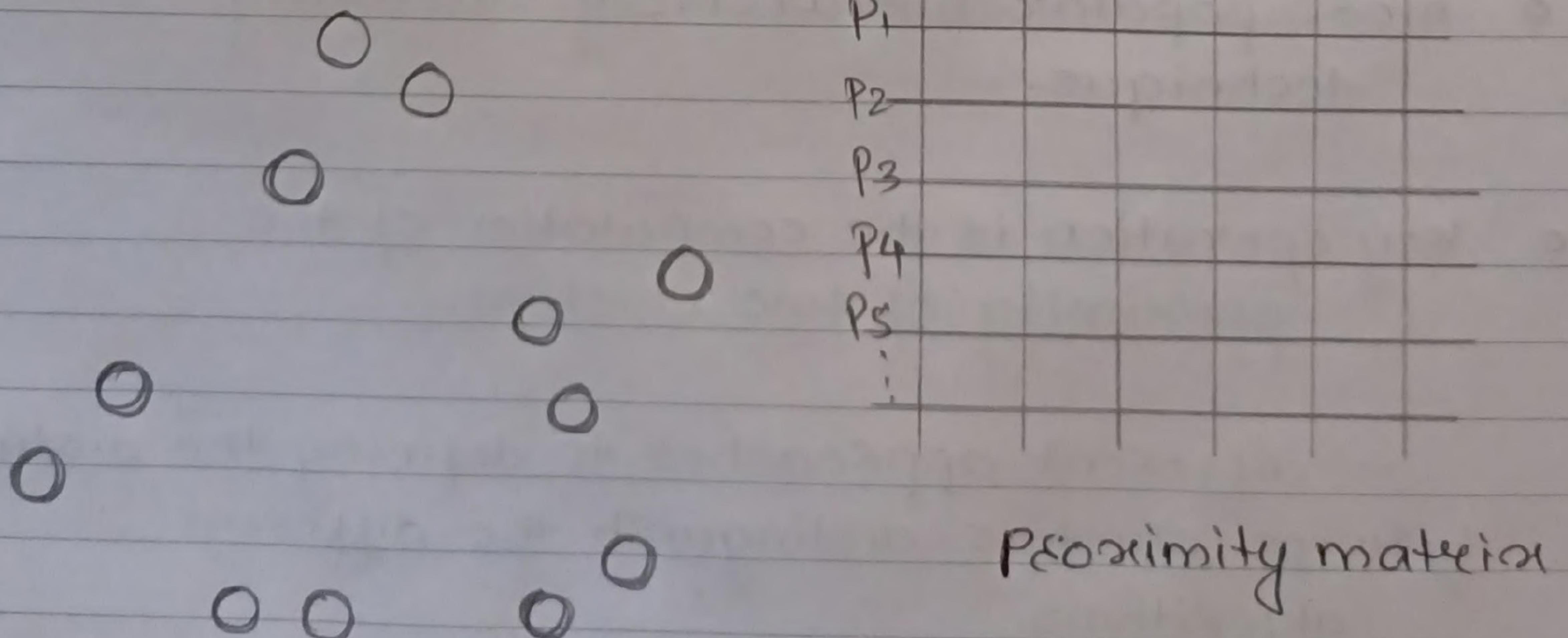
- o most popular hierarchical clustering technique.
- o key operation is the computation of the proximity of two clusters.
 - different approaches to defining the distance between clusters distinguish the different algorithms.

- Basic algorithm is straightforward

1. compute the proximity matrix
2. let each data point be a cluster
3. Repeat
4. merge the two closest clusters
5. update the proximity matrix
6. until only a single cluster remains

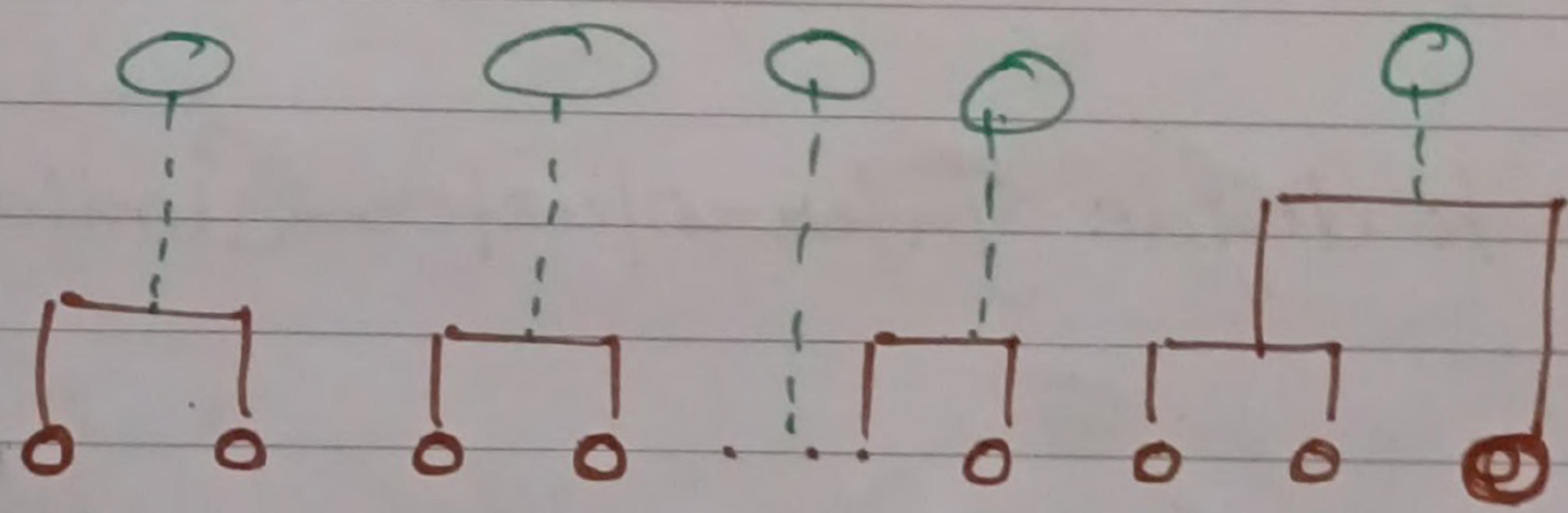
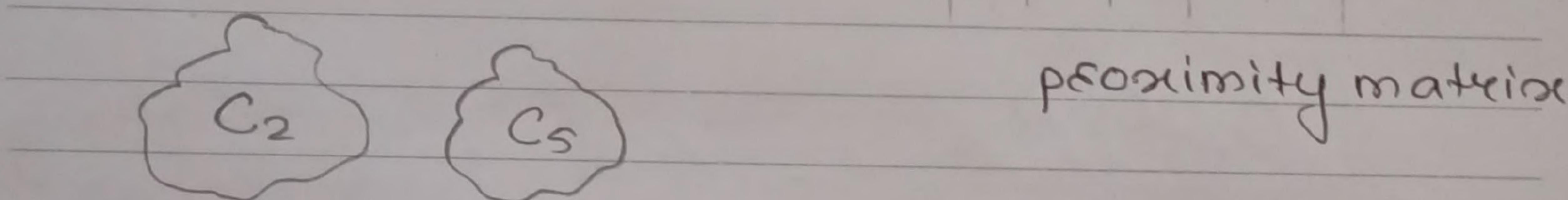
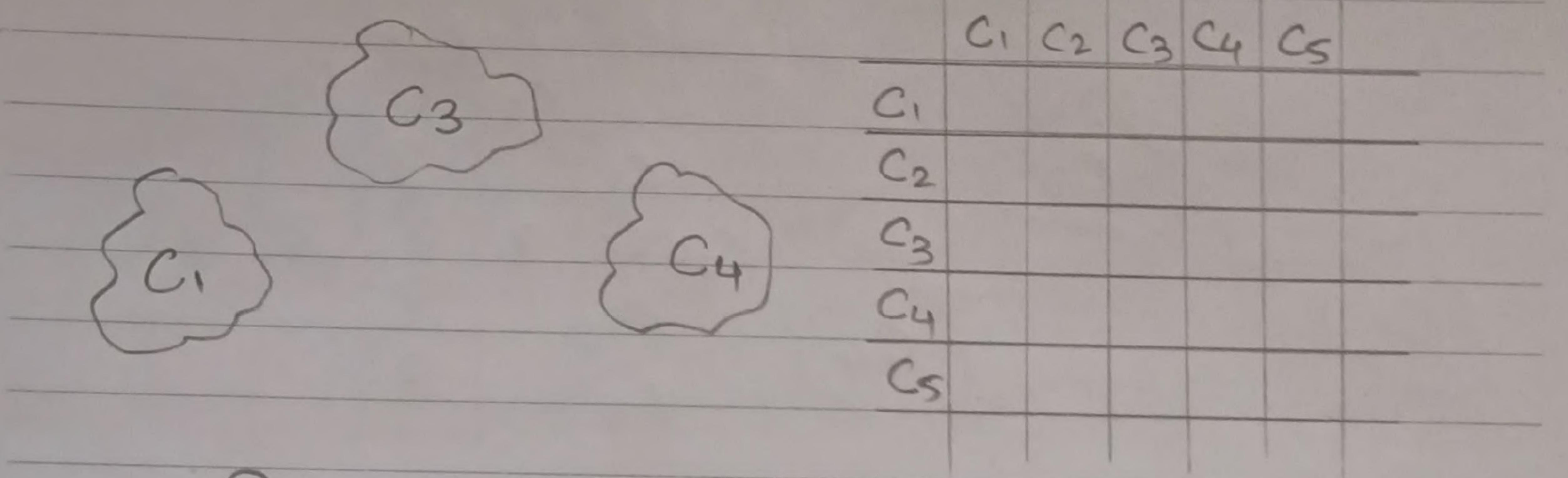
Starting Situation:

- o Start with clusters of individual points and a proximity matrix.

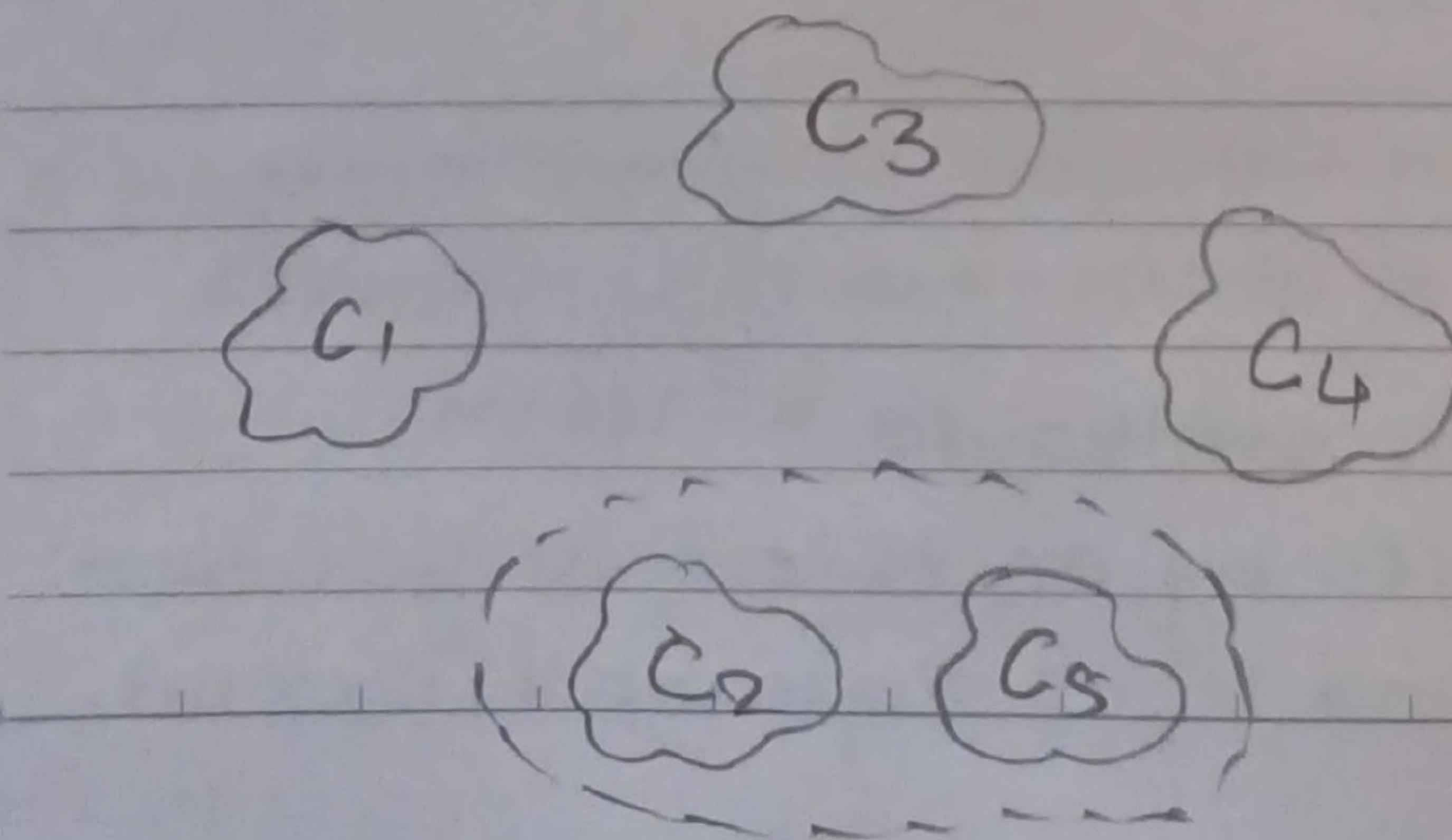


Intermediate Situation:

- after some merging steps, we have some clusters.

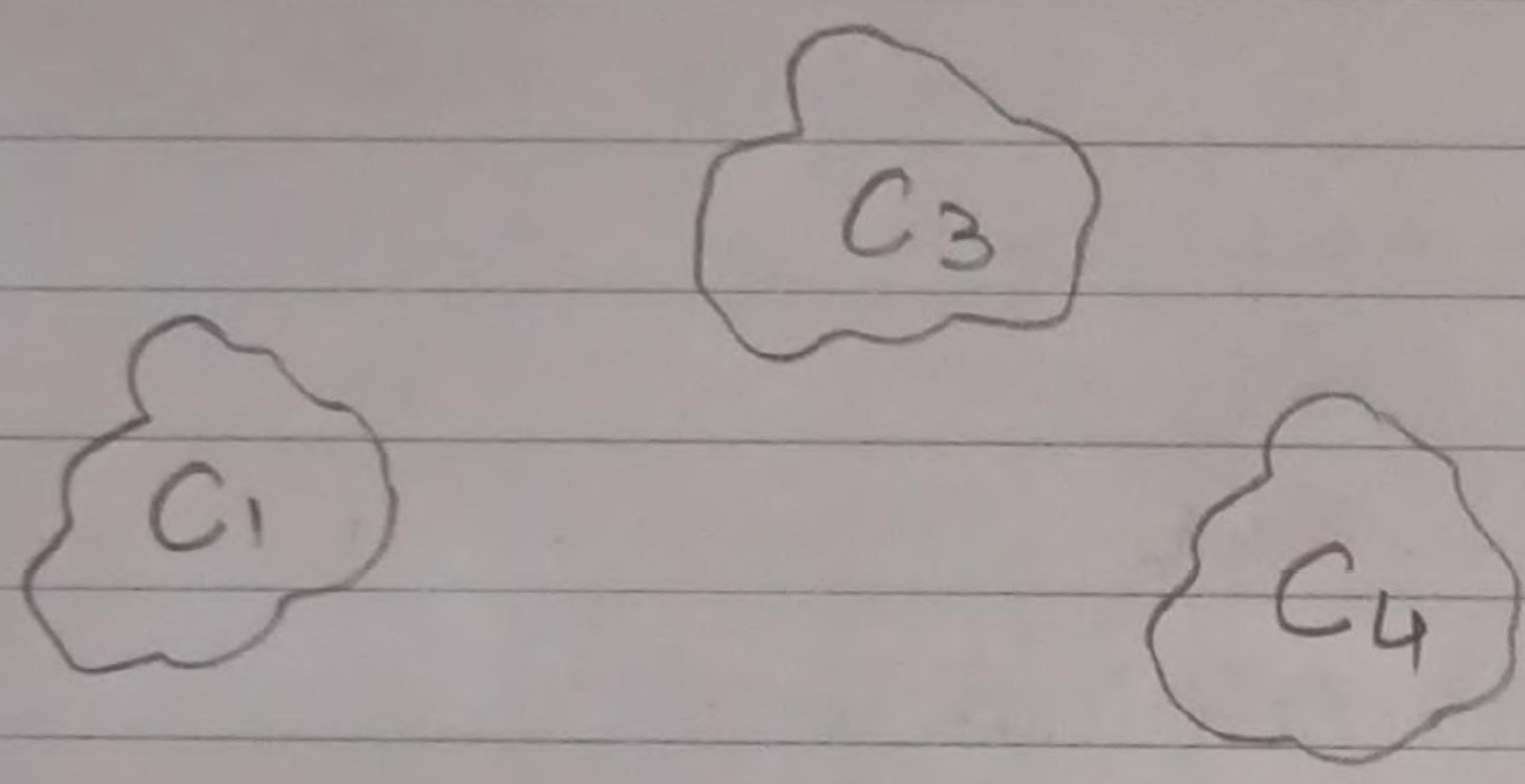


- we want to merge the two closest clusters (C₂ and C₅) and update the proximity matrix.



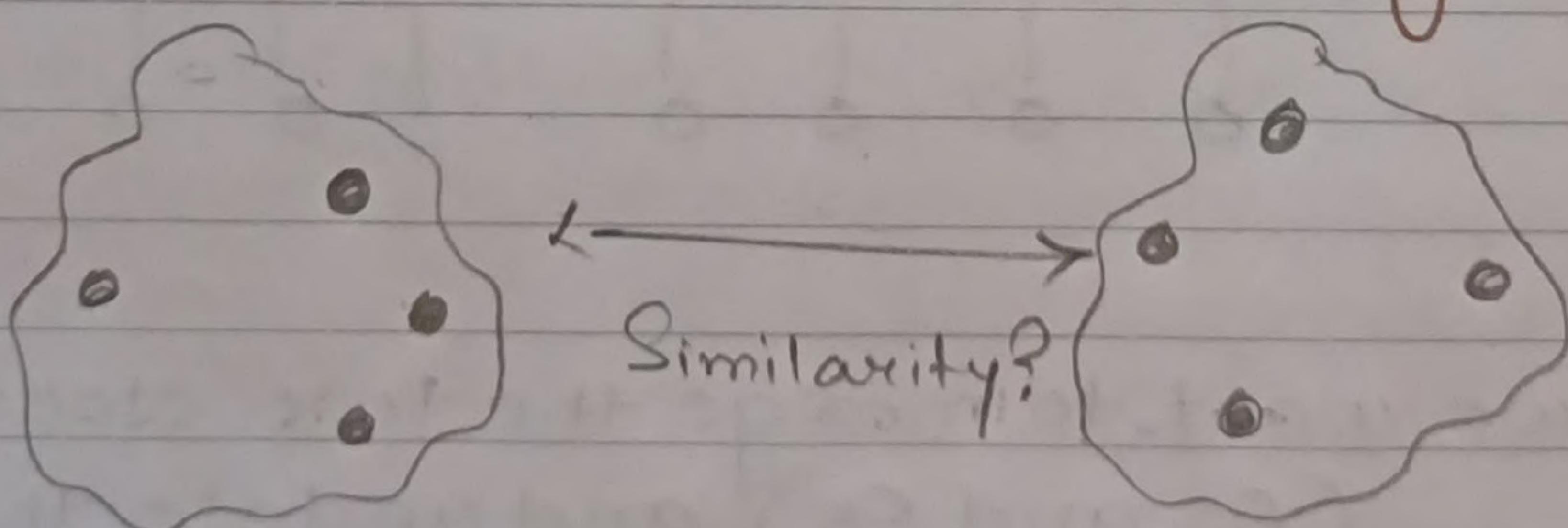
After merging :

- the question is "How do we update the proximity matrix?"



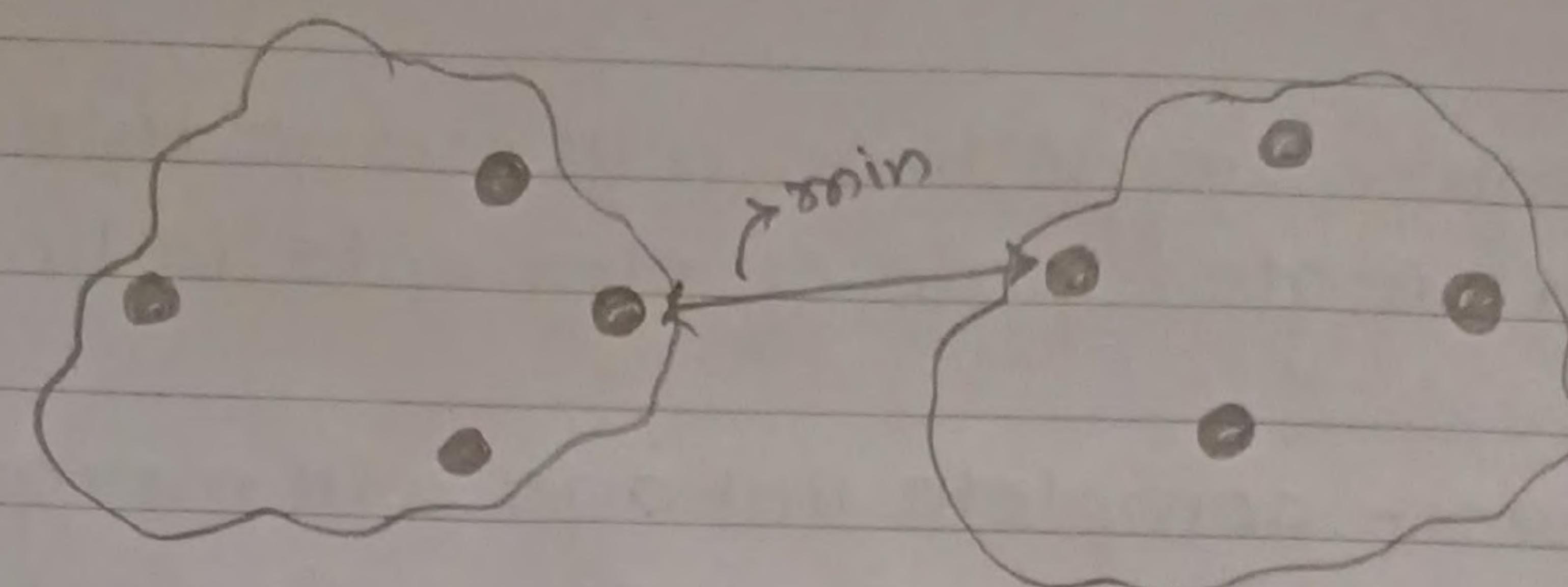
	C ₁	C ₂ ∪ C ₅	C ₃	C ₄
C ₁	?			
C ₂ ∪ C ₅	?	?	?	?
C ₃		?		
C ₄		?		

How to define Inter-cluster Similarity?

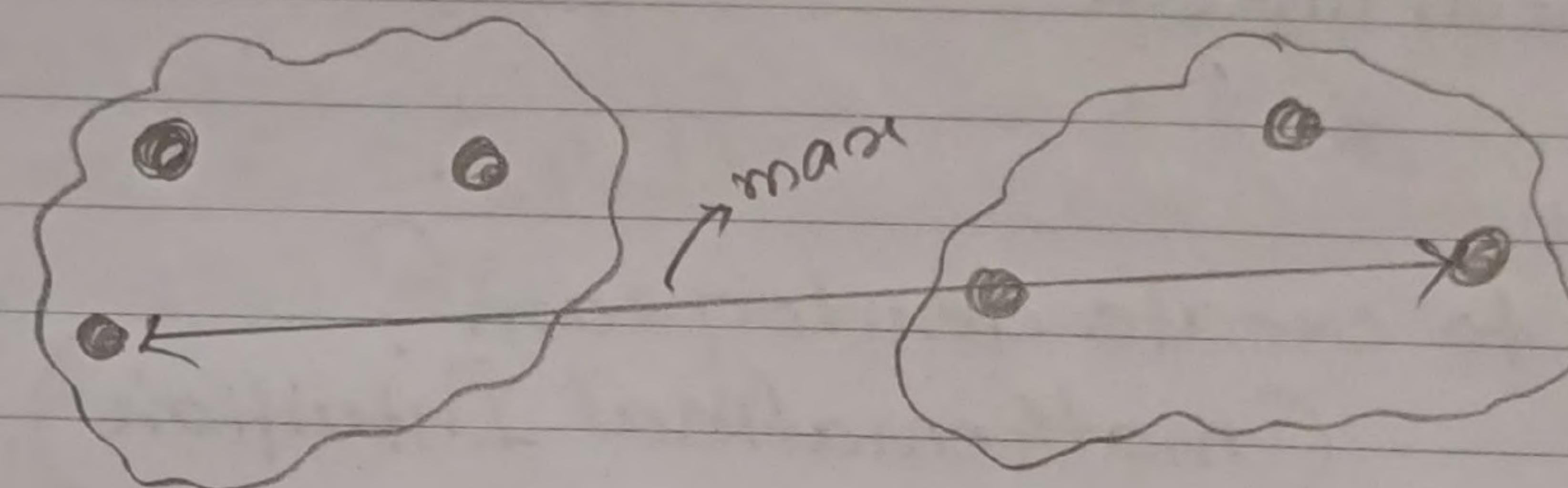


- o min
- o max
- o group average
- o distance between centroids
- o other methods driven by an objective function
 - ward's method uses squared errors.

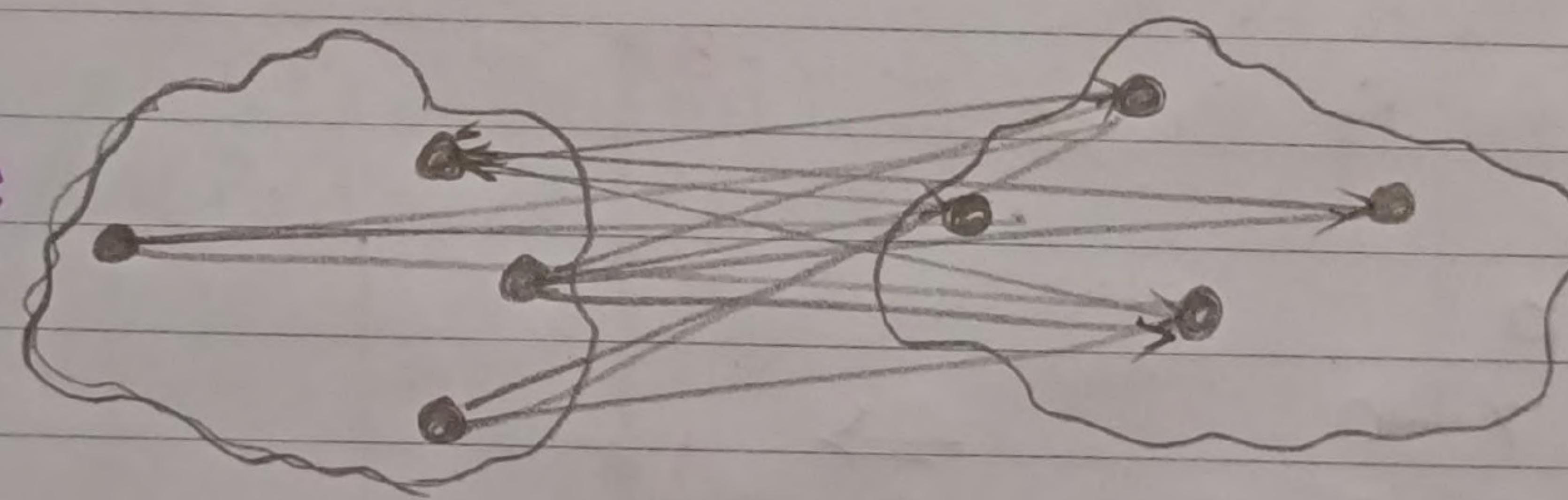
min



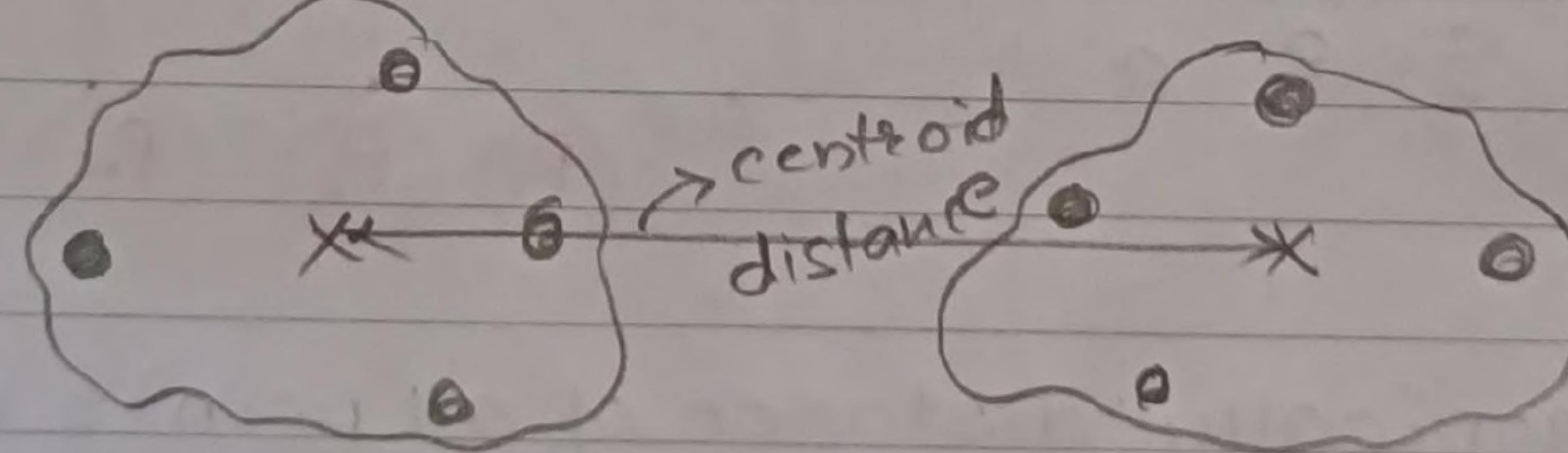
max



group
average



distance
between
centroids



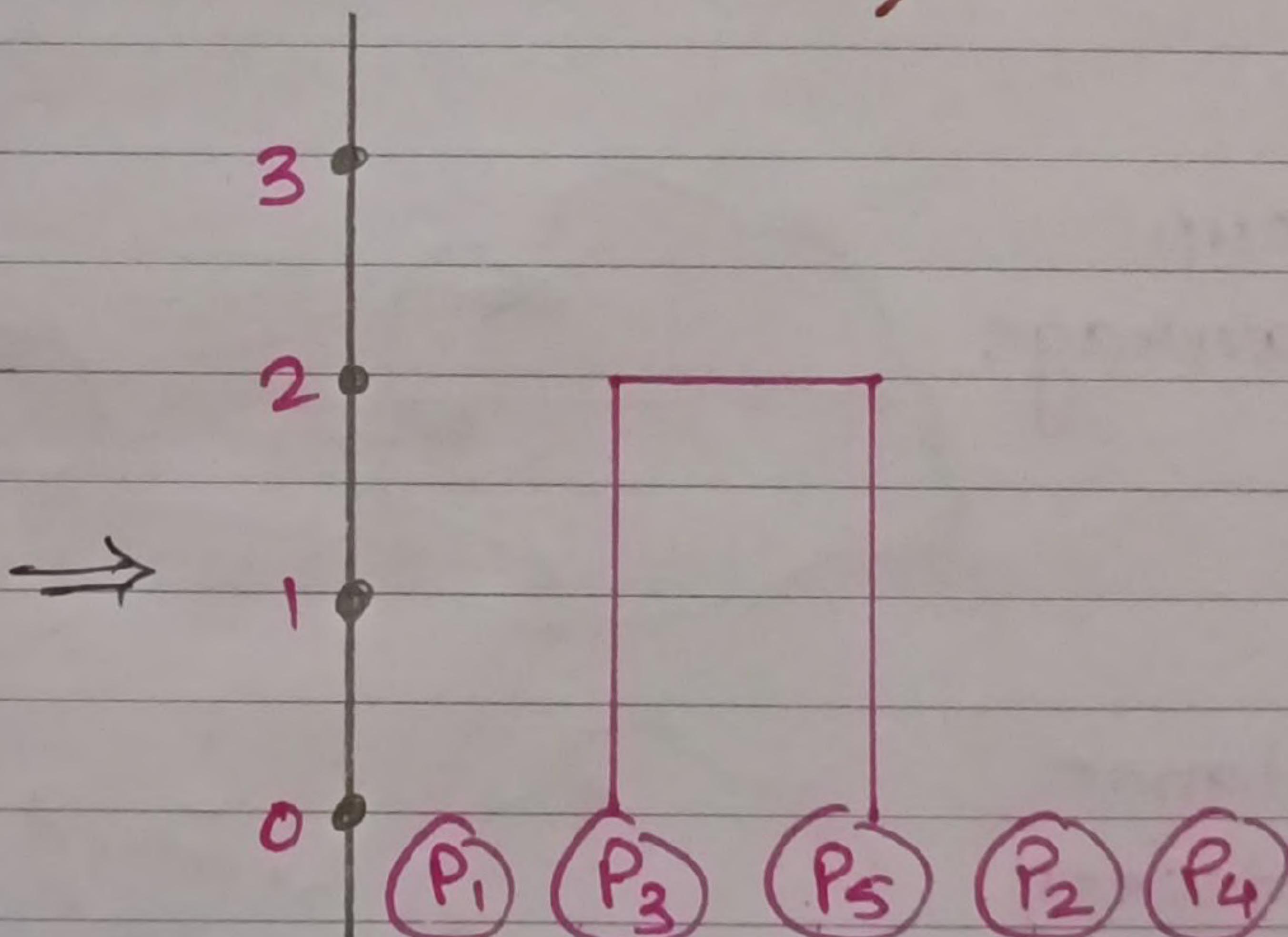
Single Link- complete link:

- Another way to view the processing of the hierarchical algorithm is that we create links between their elements in order of increasing distance.

- The min- single link, will merge two clusters when a single pair of elements is linked.
- The max- complete linkage will merge two clusters when all pairs of elements have been linked.

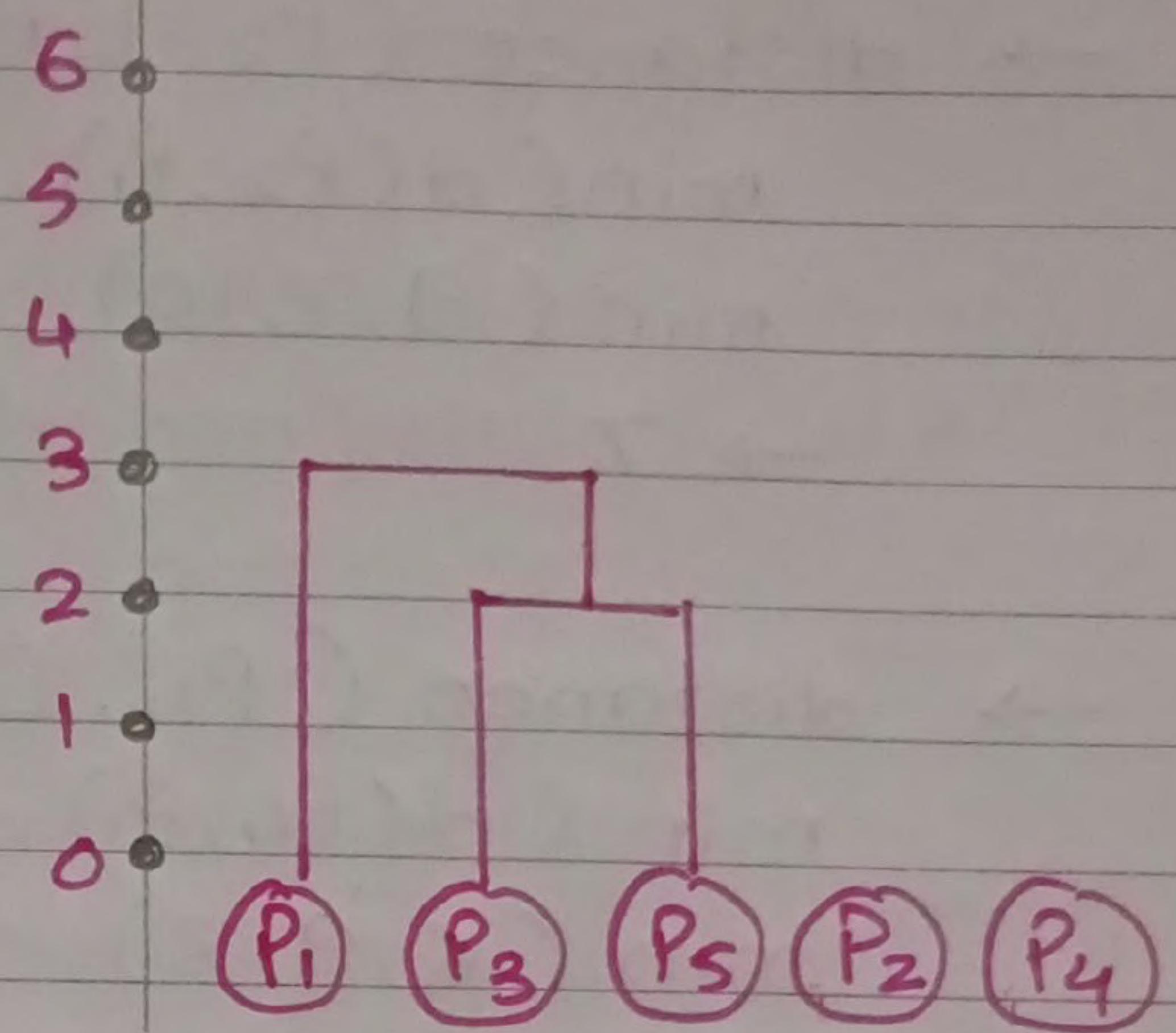
How to create dendrogram? (mathematical Intuition)

	P ₁	P ₂	P ₃	P ₄	P ₅
P ₁	0				
P ₂	9	0			
P ₃	3	7	0		
P ₄	6	5	9	0	
P ₅	11	10	2	8	0



Diagonally distance of all points will be zero.

	P_1	P_2	$P_3 \cup P_5$	P_4
P_1	0			
P_2	9	0		
$P_3 \cup P_5$	3	7	0	
P_4	6	5	8	0

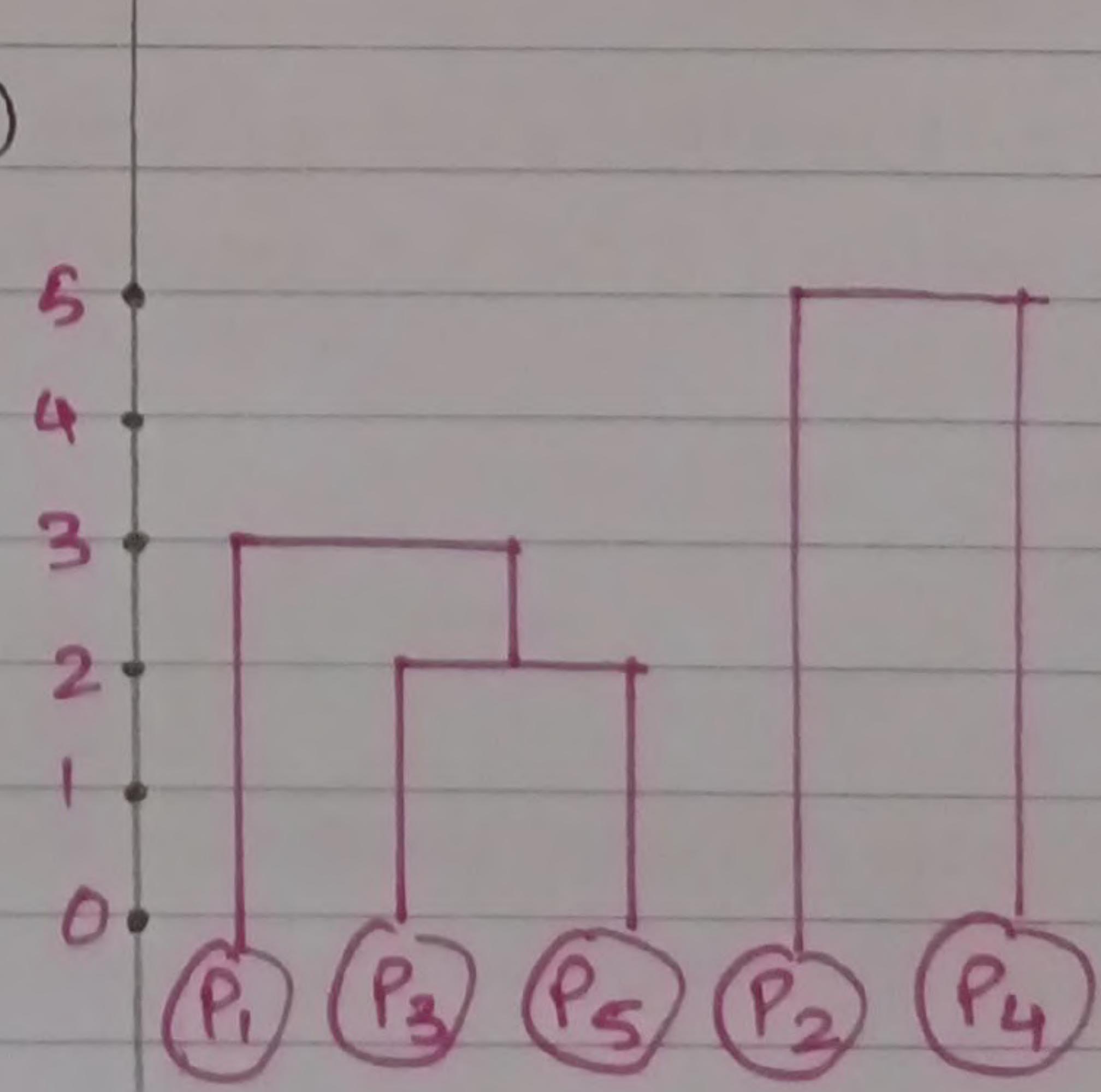


$\rightarrow \text{distance}(P_1, [P_3, P_5])$
 $\min(d(P_1, P_3), d(P_1, P_5))$
 $\min(3, 11)$
 $\rightarrow 3.$

$\rightarrow \text{distance}(P_2, [P_3, P_5])$
 $\min(d(P_2, P_3), d(P_2, P_5))$
 $\min(7, 10)$
 $\rightarrow 7$

$\rightarrow \text{distance}([P_3, P_5], P_4)$
 $\min(d(P_3, P_4), d(P_5, P_4))$
 $\min(9, 8)$
 $\rightarrow 8.$

	$P_1, P_3, P_5]$	P_2	P_4
$[P_1, P_3, P_5]$	0	.	
P_2	7	0	
P_4	6	5	0



→ distance ($P_2, [P_1, P_3, P_5]$)

$$\min(d(P_2, P_1), d(P_2, P_3), d(P_2, P_5))$$

$$\min(9, 7, 10)$$

$$\rightarrow 7$$

→ distance ($P_4, [P_1, P_3, P_5]$)

$$\min(d(P_4, P_1), d(P_4, P_3), d(P_4, P_5))$$

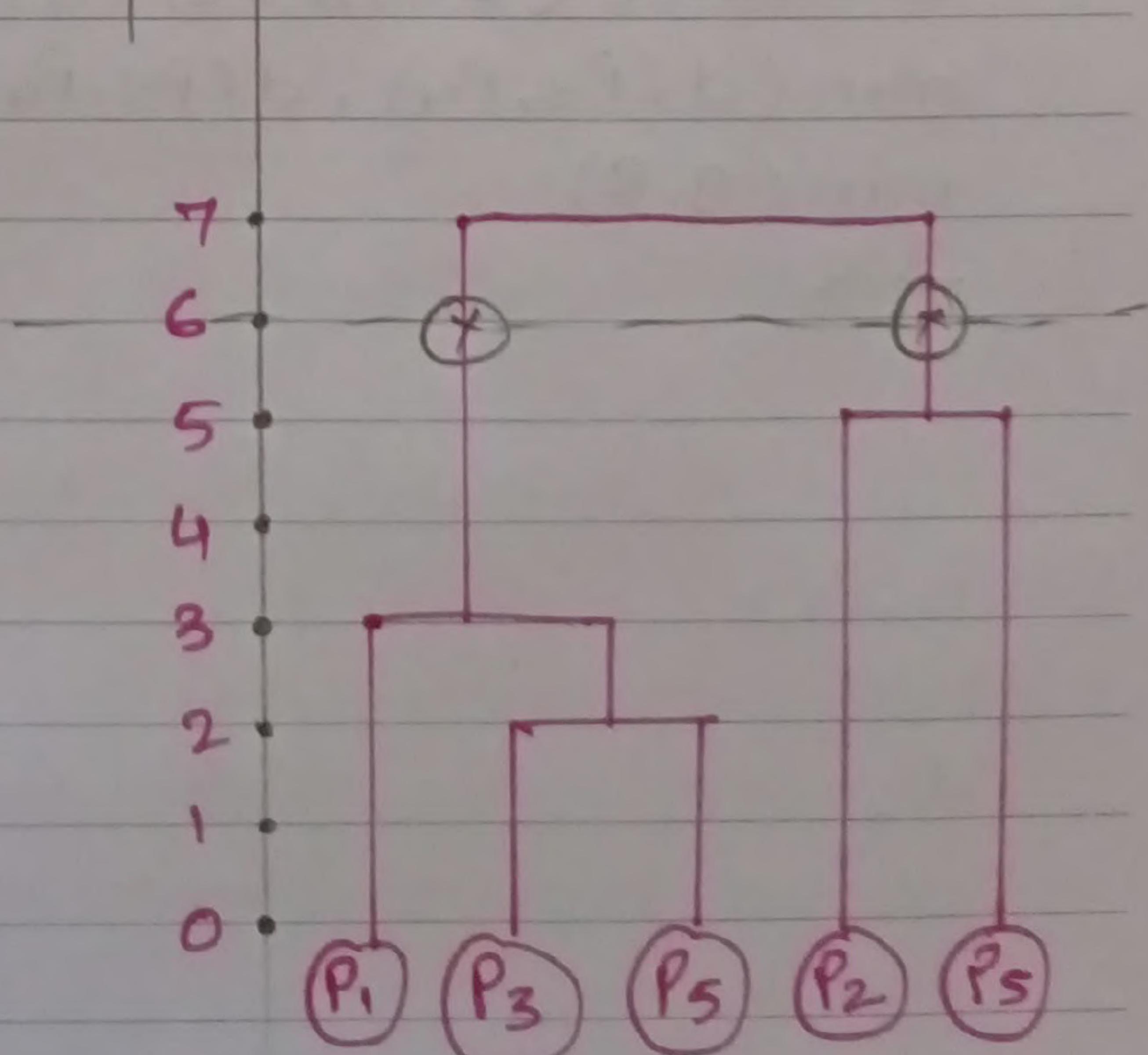
$$\min(6, 9, 8)$$

$$\rightarrow 6.$$

Now the new cluster will be

$[P_1, P_3, P_5]$ and $[P_2, P_4]$

	$[P_1, P_3, P_5]$	$[P_2, P_4]$
$[P_1, P_3, P_5]$	0	
$[P_2, P_4]$	6	0



Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$\text{proximity}(\text{cluster}_i, \text{cluster}_j) = \frac{\sum_{p_i \in \text{cluster}_i, p_j \in \text{cluster}_j} \text{proximity}(p_i, p_j)}{|\text{cluster}_i| \times |\text{cluster}_j|}$$

$p_i \in \text{cluster}_i$
 $p_j \in \text{cluster}_j$

- Limitations: biased towards globular clusters.

Cluster Similarity: Ward's method

- Similarity of two clusters is based on the increase in squared error (SSE) when two clusters are merged.
- Biased towards globular clusters

Hierarchical clustering: time and space requirements

$O(N^2)$ space since it uses the proximity matrix.
- N is the number of points

$O(N^3)$ time in many cases.

- There are N steps and at each step the size, N^2 , proximity matrix must be updated and searched.
- complexity can be reduced to $O(N^2 \log(N))$ time for some approaches.

Density-Based Clustering

- Density-based spatial clustering of applications with noise (DBSCAN)

- o DBSCAN is a Density-Based Clustering algorithm.

The DBSCAN algorithm uses two parameters:

- o minPts: The minimum number of points (a threshold) clustered together for a region to be considered dense.
- o eps(ϵ): A distance measure that will be used to locate the points in the neighbourhood of any point.

These parameters can be understood if we explore two concepts called Density Reachability and Density Connectivity.

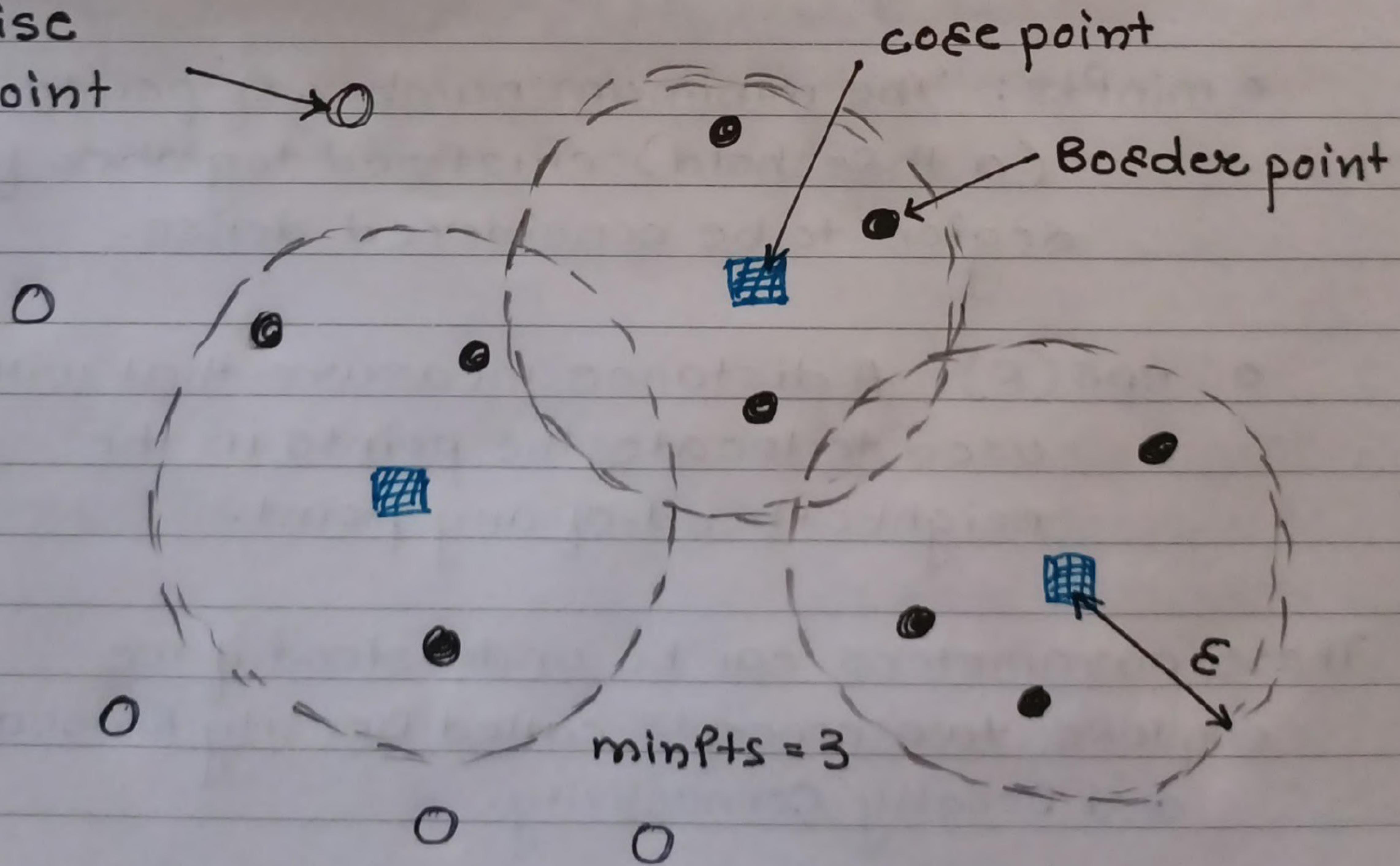
Reachability: in terms of density establishes a point to be reachable from another if it lies within a particular distance (ϵ) from it.

connectivity: involves a transitivity based chaining - approach to determine whether points are located in a particular cluster.

These are three types of points after the DBSCAN clustering is complete:

Noise

point



within a circle of radius Eps.
Centered at point B; number of points

core: this is a point that has at least m points within distance n from itself.

- a point is a core point if it has more than a specified number of points (minpts) within Eps

- These points belong in a dense region and are at the interior of a cluster.

Border: this is a point that has at least one core point at a distance 'n'.

- a border point has fewer than minpts within Eps, but is in the neighbourhood of a core point.

Noise: this is a point that is neither a core nor a border and it has less than m points within distance n from itself.

DBSCAN:

- Density at point P: number of points within a circle of radius Eps.
- Dense Region: A circle of radius Eps that contains at least minpts points.

a spe
Eps

- These points belong in a dense region and are at the interior of a cluster.

Border: this is a point that has at least one core point at a distance 'n'.

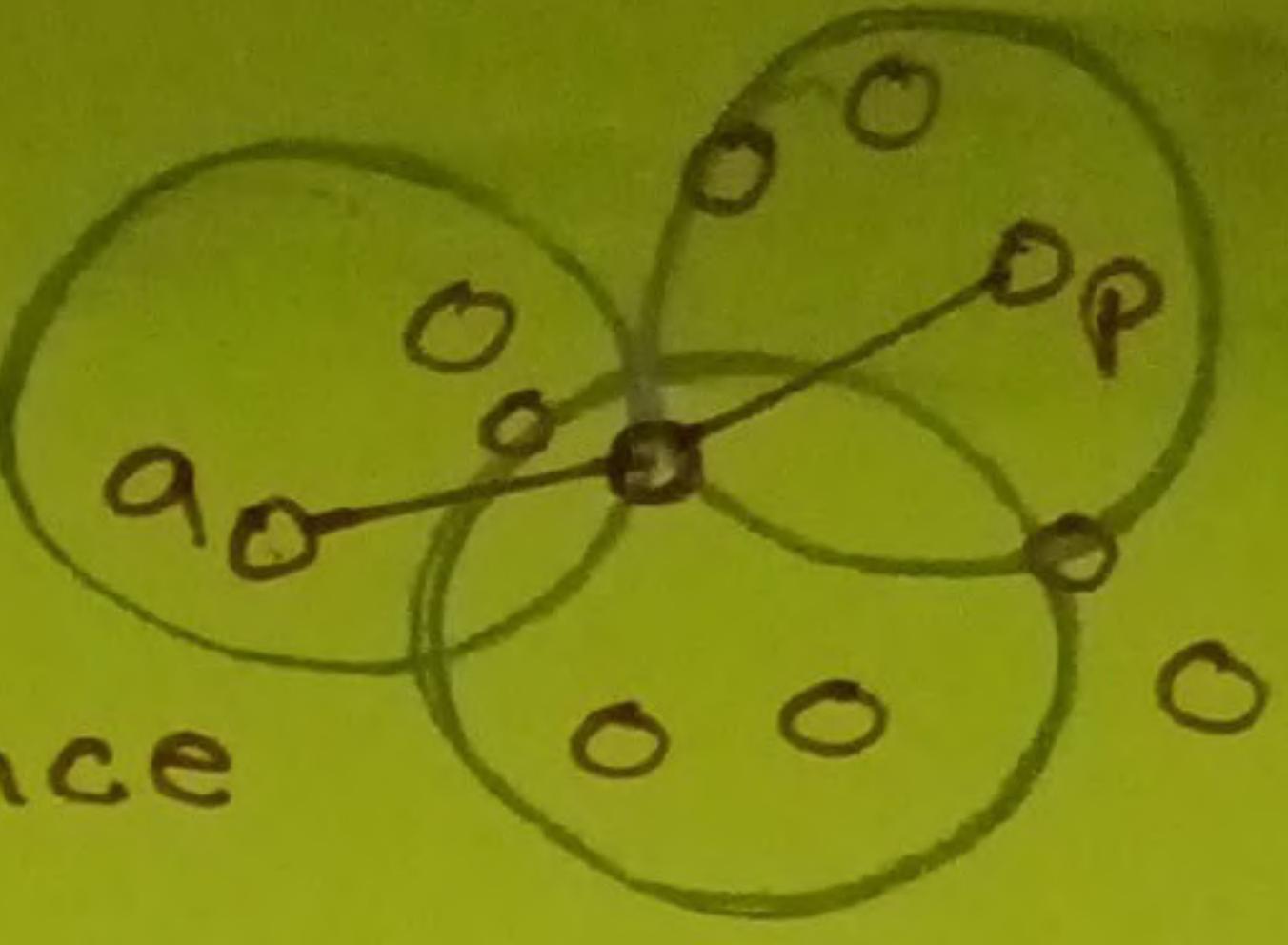
- a border point has fewer than minpts within Eps, but is in the neighbourhood of a core point.

Noise: this is a point that is neither a core nor a border and it has less than m points within distance n from itself.

Algo

Density edge:

- we place an edge between two core points q and p if they are within distance ϵ_{ps} .



Density-connected:

- o The pick - a point p is density-connected to a pt q if there is a path of edges from p to q .

- o If there are at least 'minPoints' points within a radius of ' ϵ ' to the point then we consider all these points to be part of the same cluster.
- o The clusters are then expanded by recursively repeating the neighbourhood calculation for each neighbouring point.

.....

- Label points as core, border and noise
- Eliminate noise points
- For every core point p that has not been assigned to a cluster
 - Create a new cluster with the point p and all the points that are density-connected to p .
 - Assign border points to the cluster of the closest core point.

Algorithmic steps \Rightarrow

- The algorithm proceeds by arbitrarily picking up a point in the dataset (until all points have been visited)
- If there are at least 'minPoints' points within a radius of ' ϵ ' to the point then we consider all these points to be part of the same cluster.
- The clusters are then expanded by recursively repeating the neighbourhood calculation for each neighbouring point.
.....
- Label points as core, border and noise
- Eliminate noise points
- For every core point p that has not been assigned to a cluster
 - Create a new cluster with the point p and all the points that are density-connected to p .
 - Assign border points to the cluster of the closest core point.

Parameter Estimation: - for DBSCAN, the parameters ϵ and minPts are needed.

- minPts : as a rule of thumb, a minimum minPts can be derived from the number of dimensions D in the dataset,

$$\text{minPts} \geq D+1$$

$\text{minPts} = 1 \rightarrow$ doesn't make any sense

$\text{minPts} \leq 2 \rightarrow$ the result will be the same as of hierarchical clustering with the single link metric, with the dendrogram cut at height ϵ .

∴ minPts must be chosen at least 3.

- ϵ : The value for ϵ can then be chosen by using a k-distance graph, plotting the distance to the $k = \text{minPts} - 1$ nearest neighbors ordered from the largest to the smallest value.

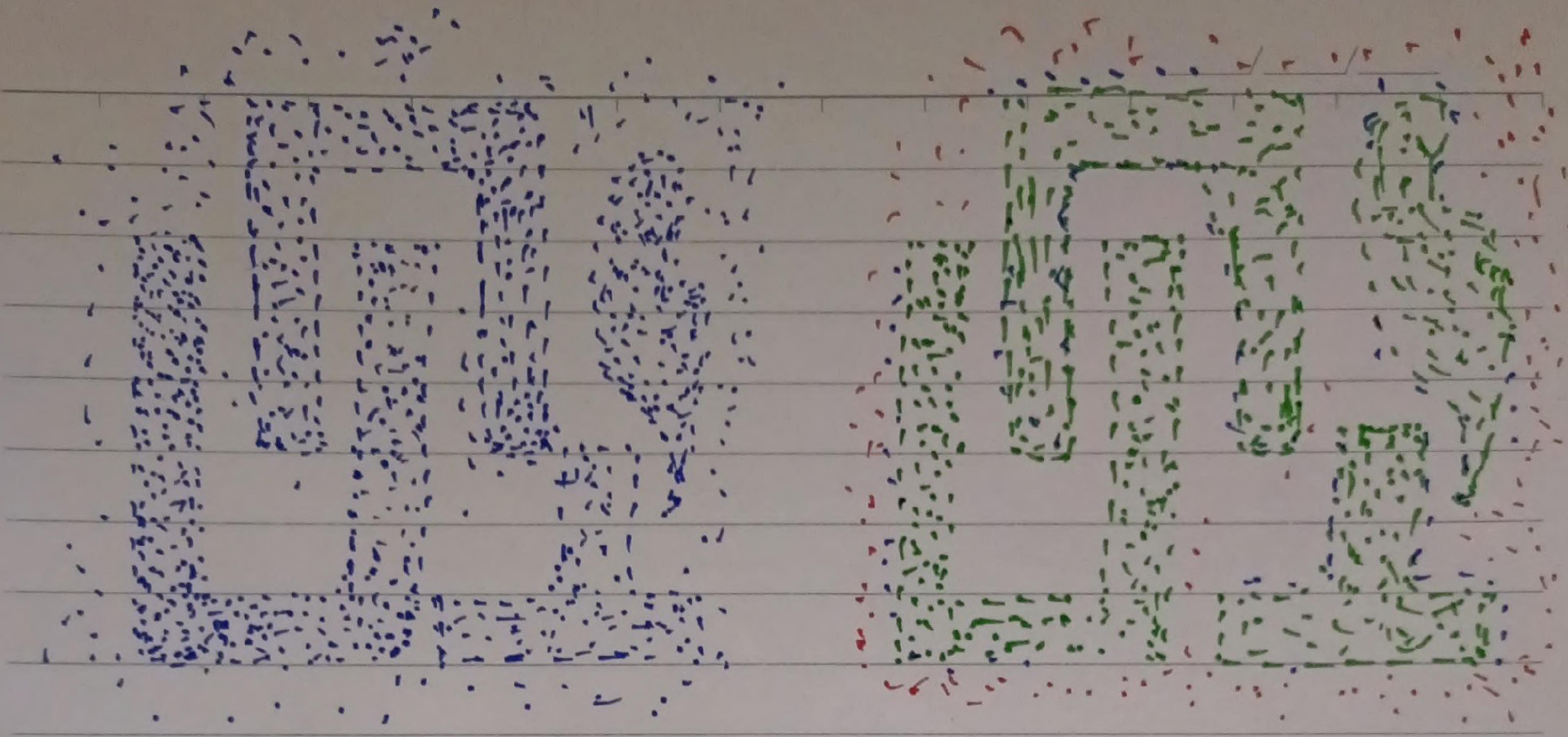
Complexity :- $O(n \log n) \rightarrow$ Best case
 $O(n^2)$ \rightarrow worst case

Advantages of DBSCAN:

- Is great at separating clusters of high density versus clusters of low density within a given dataset.
- Is great with handling outliers within the dataset.

Disadvantages of DBSCAN:

- does not work well when dealing with clusters of very different densities.
while DBSCAN is great at separating high density clusters from low density clusters, DBSCAN struggles with clusters of similar density.
- struggle with high dimensionality data.



original points.

points types:
core, border and
noise

Eps = 10, minPts = 4