Data Science and Artificial Intelligence

Machine Learning

Unsupervised learning

Lecture No.5

By- SIDDHARTH SABHARWAL SIR

Recap of Previous Lecture

Topic — Agglomerative clustering

Topic — Divisive Clustering → algo.

Topic

Topic

Topic

Turn on Slide map

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

MST

Divisive & agglomerative

Question - NPTEL


NEVER
Give Up
ON YOUR
Dreams

Basics of Machine Learning

Agglomerative Clustering

- $O(N^3)$ (naive)
- $O(N^2 + N^2 \log N)$
- advantage/disadvantage

Hierarchical vs flat

REVISION

Agglomerative Clustering

Clustering

→ Iterative way of applying K-means.

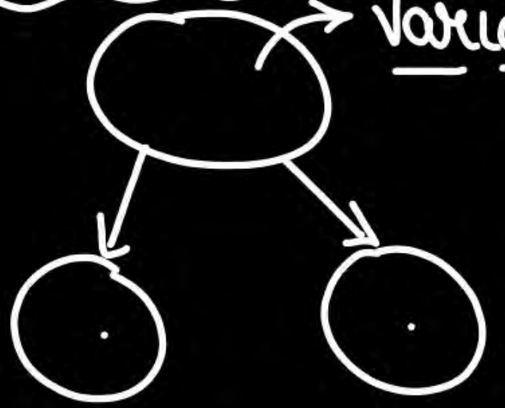
Divisive Clustering

Divisive clustering, also known as "top-down" clustering, is a type of hierarchical clustering that starts with all data points in a single cluster and iteratively splits them into smaller clusters until each data point is its own cluster or until a stopping criterion is met. This approach is the opposite of agglomerative clustering, which starts with each data point as its own cluster and then merges them.

Clustering

Divisive Clustering

- **Initial State**: All data points are grouped into a single cluster.
- **Process**: Iteratively splits the clusters into smaller clusters.
- **Stopping Criteria**: The process continues until each data point is in its own cluster or a predefined number of clusters is reached.



$$\text{Variance} \Rightarrow \frac{\sum \|x_i - \mu_i\|^2}{N}$$

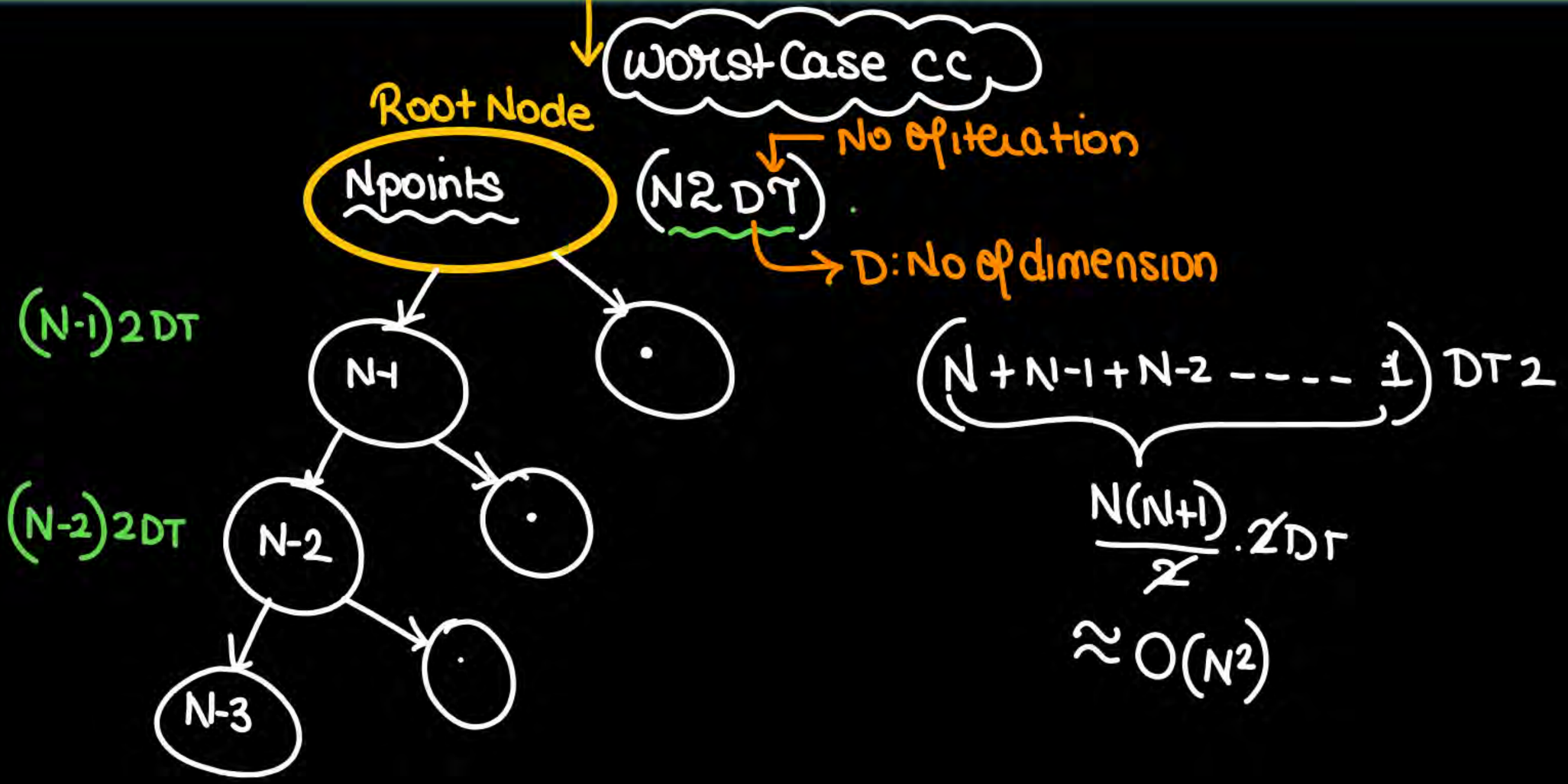
Clustering

Divisive Clustering

- **Steps in Divisive Clustering**
- **Start with a Single Cluster**:
  - Begin with all data points in one large cluster.
- **Choose a Cluster to Split**:
  - Select the cluster that needs to be split. This could be based on various criteria such as the largest cluster or the cluster with the highest variance.
- **Split the Cluster**:
  - Use a clustering algorithm (such as K-means) to divide the chosen cluster into two smaller clusters. This is the core step where a decision on how to split the data is made.
- **Repeat**:
  - Continue the process of choosing and splitting clusters until the stopping criterion is met.

Clustering

Computational Complexity



Clustering

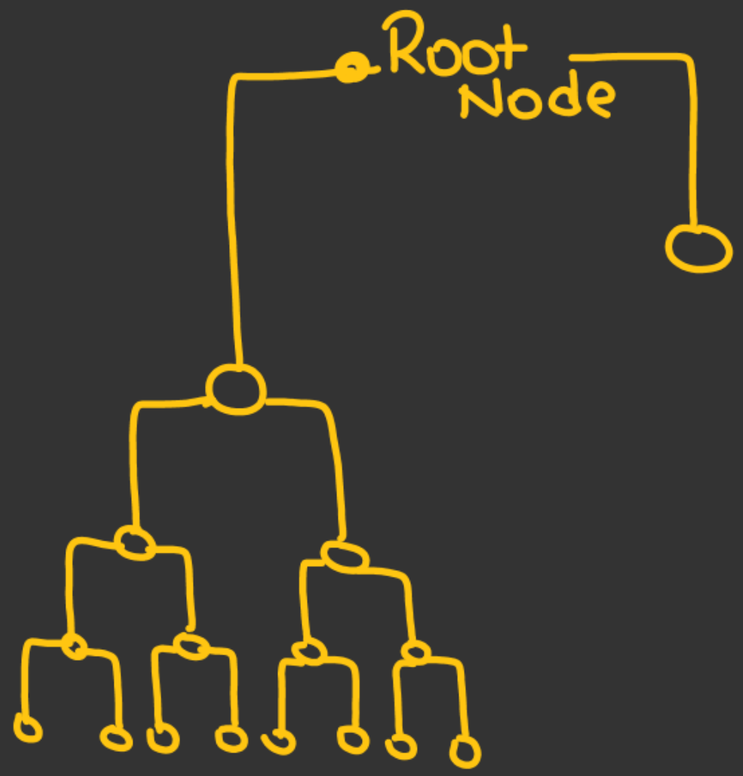
Comparison

Feature	Agglomerative Clustering	Divisive Clustering
Approach	Bottom-up	Top-down
Initial State	Each data point is its own cluster	All data points are in a single cluster
Process	Merges the closest pairs of clusters iteratively	Splits the clusters iteratively
Termination Condition	Until all points are merged into one cluster or a specified number of clusters is achieved	Until each point is its own cluster or a specified number of clusters is reached
Complexity	Typically more computationally expensive for large datasets due to repeated merging steps	Can be more efficient for large datasets as it avoids repeated merging
Example Algorithms	Single linkage, complete linkage, average linkage, Ward's method	Recursive application of clustering algorithms like K-means or spectral clustering
Dendrogram	Built from the bottom up, starting with individual points	Built from the top down, starting with all points
Usage	Commonly used due to simplicity and easy implementation	Less commonly used due to complexity in deciding optimal splits

Provide Better Picture of data.

$(N^2+N^2\log N)$

$O(N^2DT)$


Root
Node

Clustering

Comparison

(Linkages)

Flexibility	Generally more flexible and easier to implement with different linkage criteria	Requires an effective strategy for splitting clusters
Sensitivity to Noise	Less sensitive to noise, as noise points are merged into clusters gradually	More sensitive to noise, as initial splits can be affected by outliers
Example Use Cases	Hierarchical document clustering, gene expression data analysis, image segmentation	Rarely used, but can be applied in specific scenarios needing top-down clustering

→ More interpretable algo.

⇓

bcoz we directly get info abt the similarity / dissimilarity of clusters (using distance)

- But K means only bifurcate and create clusters.
- for similarity / dissimilarity b/w clusters we need to find

distances b/w clusters.

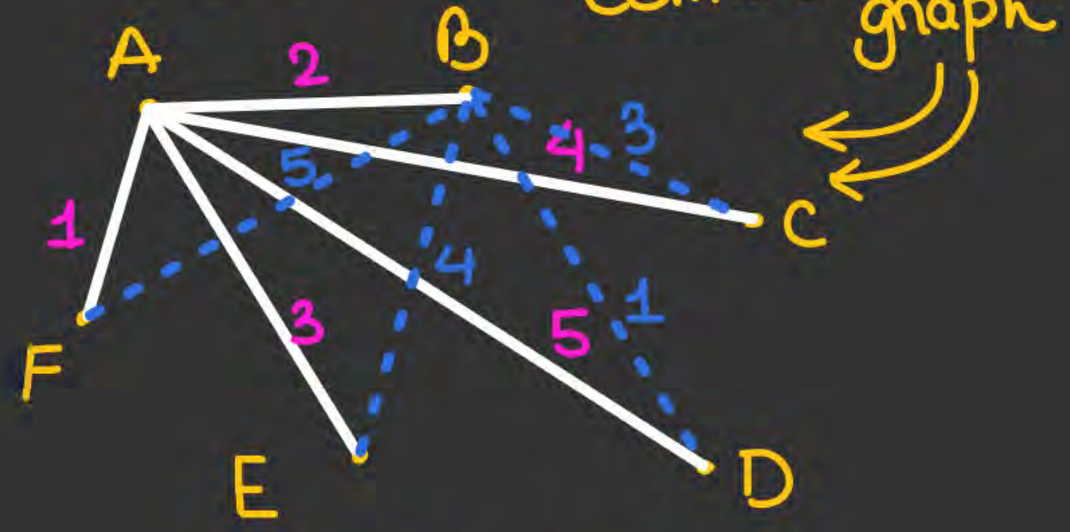
Clustering

Divisive Clustering using MST

A Minimum Spanning Tree (MST) is a subset of the edges of a connected, undirected graph that connects all the vertices together, without any cycles, and with the minimum possible total edge weight. In other words, it is a tree that spans all the vertices in the graph and has the smallest sum of edge weights among all possible spanning trees.

Root data points

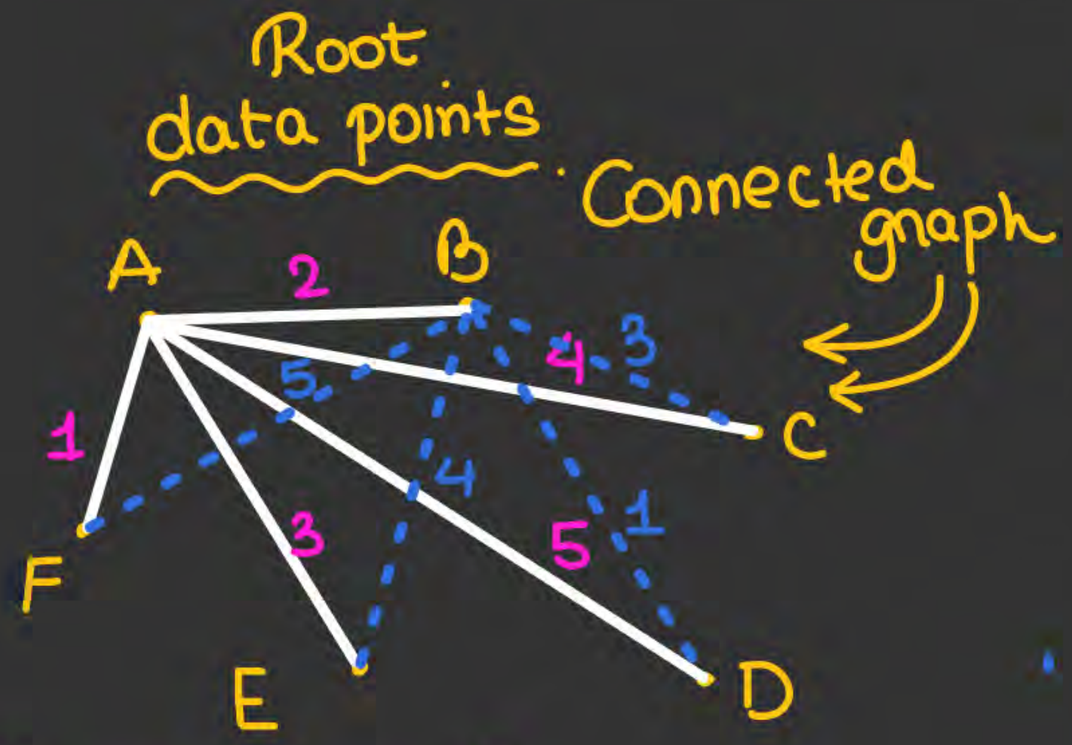
Connected graph



	A	B	C	D	E	F
A	0	2	4	5	3	1
B	2	0	3	1	4	5
C	4	3	0	2	4	1
D	5	1	2	0	5	3
E	3	4	4	5	0	2
F	1	5	1	3	2	0

- all points in dataset are connected to each other with a weight

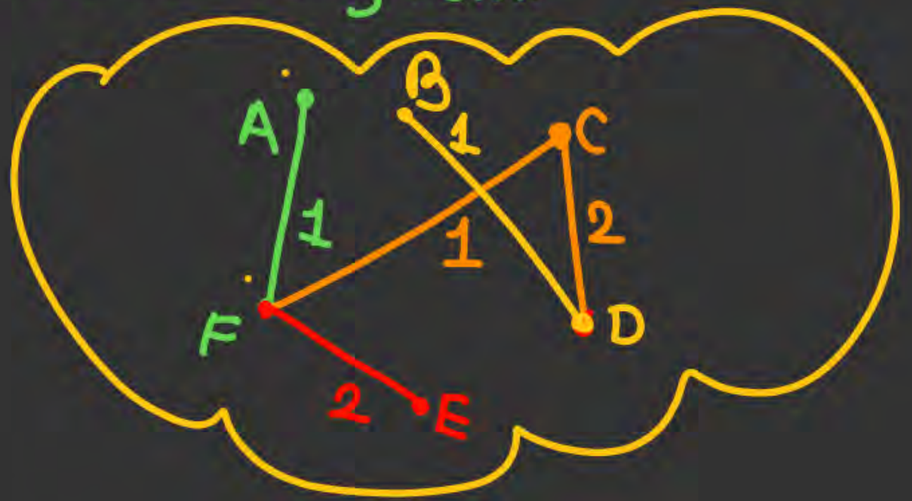
MST ⇒
(I) all points
(II) edges with min weight
(III) all vertices shd be connected
(IV) No cycles.



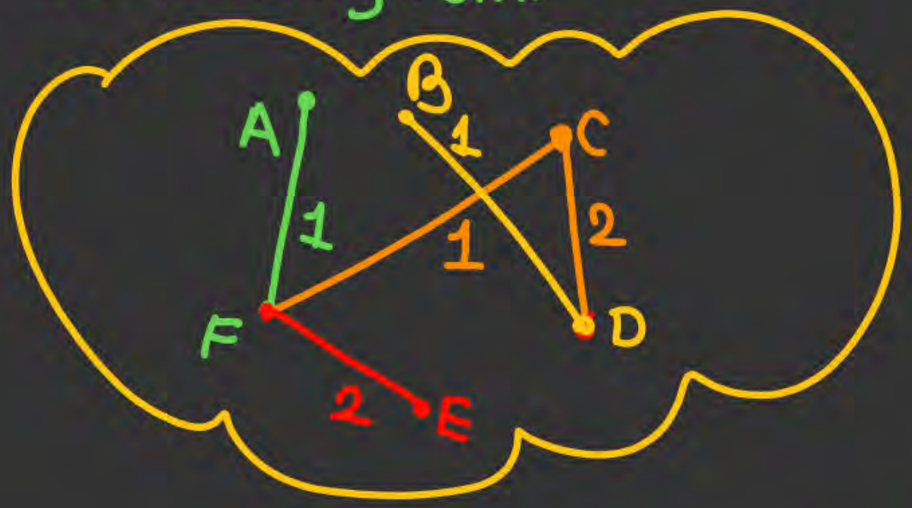
	A	B	C	D	E	F
A	0	2	4	5	3	1
B	2	0	3	1	4	5
C	4	3	0	2	4	1
D	5	1	2	0	5	3
E	3	4	4	5	0	2
F	1	5	1	3	2	0

- all points in dataset are connected to each other with a weight

Start with any Point



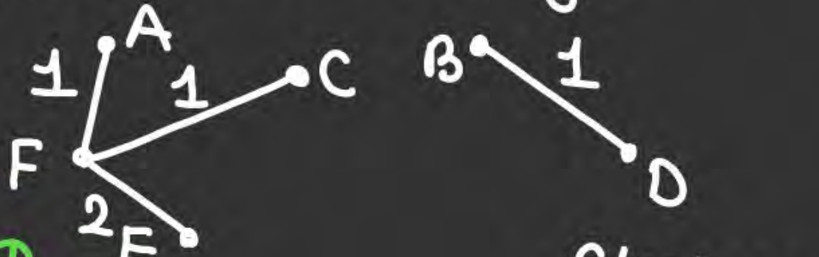
Start with any Point



If we have MST ⇒ we have Root Node (all points in MST)

1) Cut edge / branch with max weight

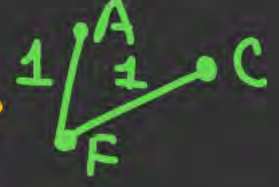
Cut C-D



Cluster 1

Cluster 2

2) Cut FE ⇒



in cluster 1

Cut BD in Cluster 2

B

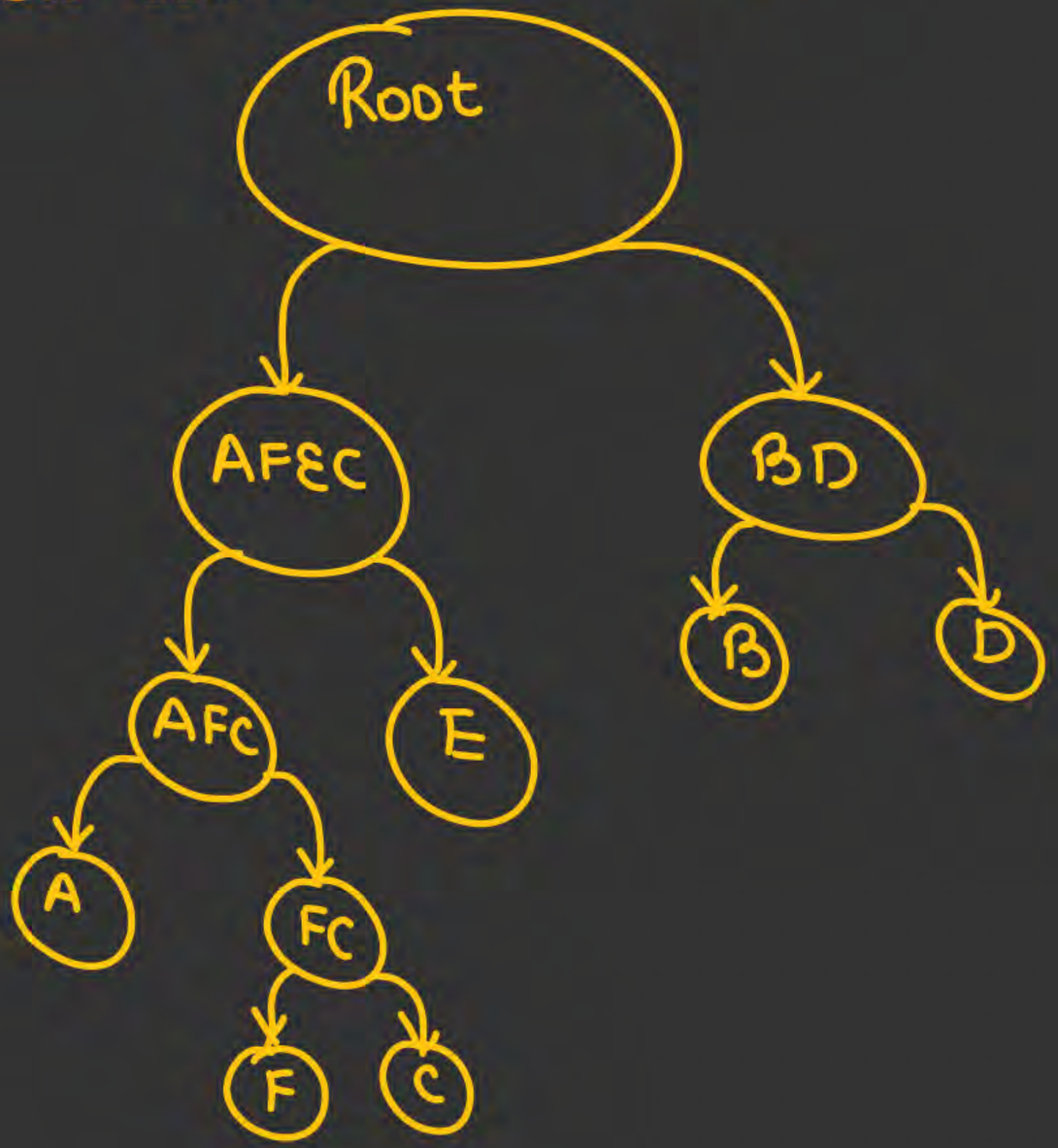
D

E

Cluster 1

Cluster 2

Now Cut AF and then FC



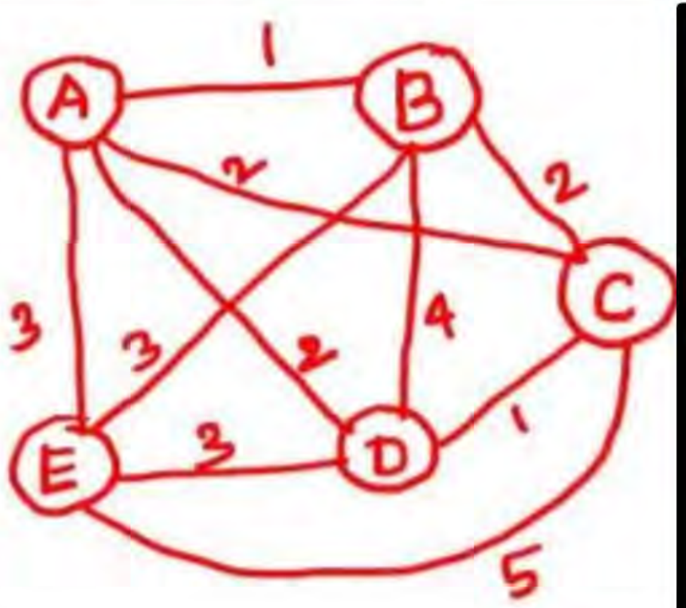
Clustering

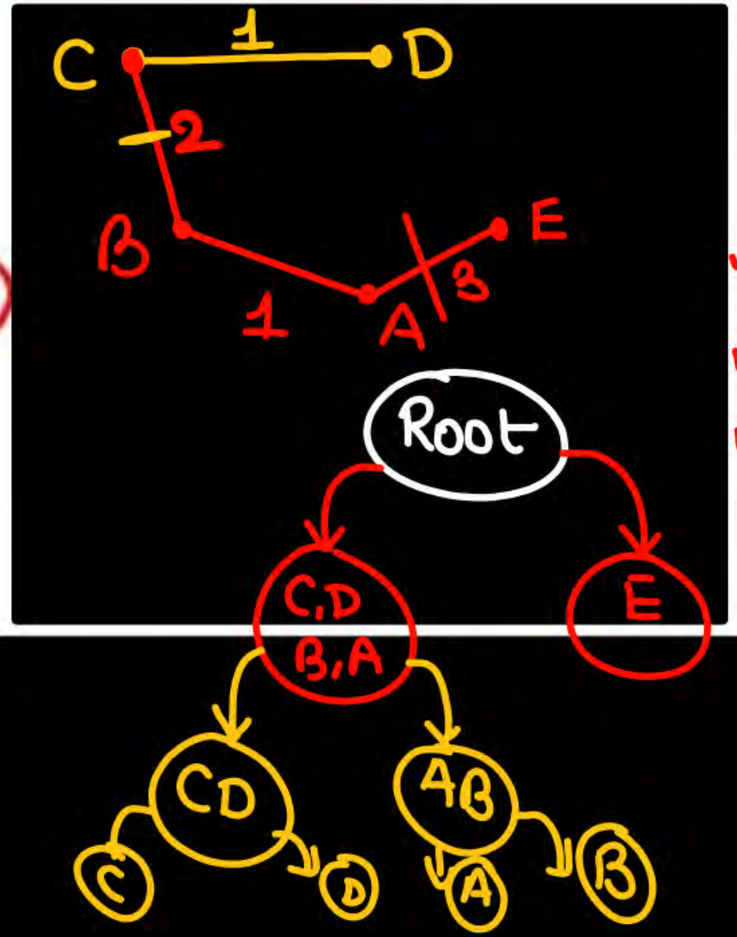
Divisive Clustering using MST

- MST starts with a tree that consists of a point p.
  - Then check for the closest pair of points (p, q) such that p is in the current tree but q is not in the tree.
  - With this closest pair of points (p, q), add q to the tree and create an edge between p and q.
- Remove the edges from MST graph from largest to smallest repeatedly.
  - All the items are in one cluster {A, B, C, D, E}
  - Largest edge is between D and E. so remove it, and make as 2 clusters- {E}, {A., B, C, D}
  - Next, remove the edge between B and C, which results in {E}, {A, B} {C, D}
  - Finally, remove the edges between A and B (also between C and D), that results {E}, {4}, {B}, {C} and {D}

Clustering

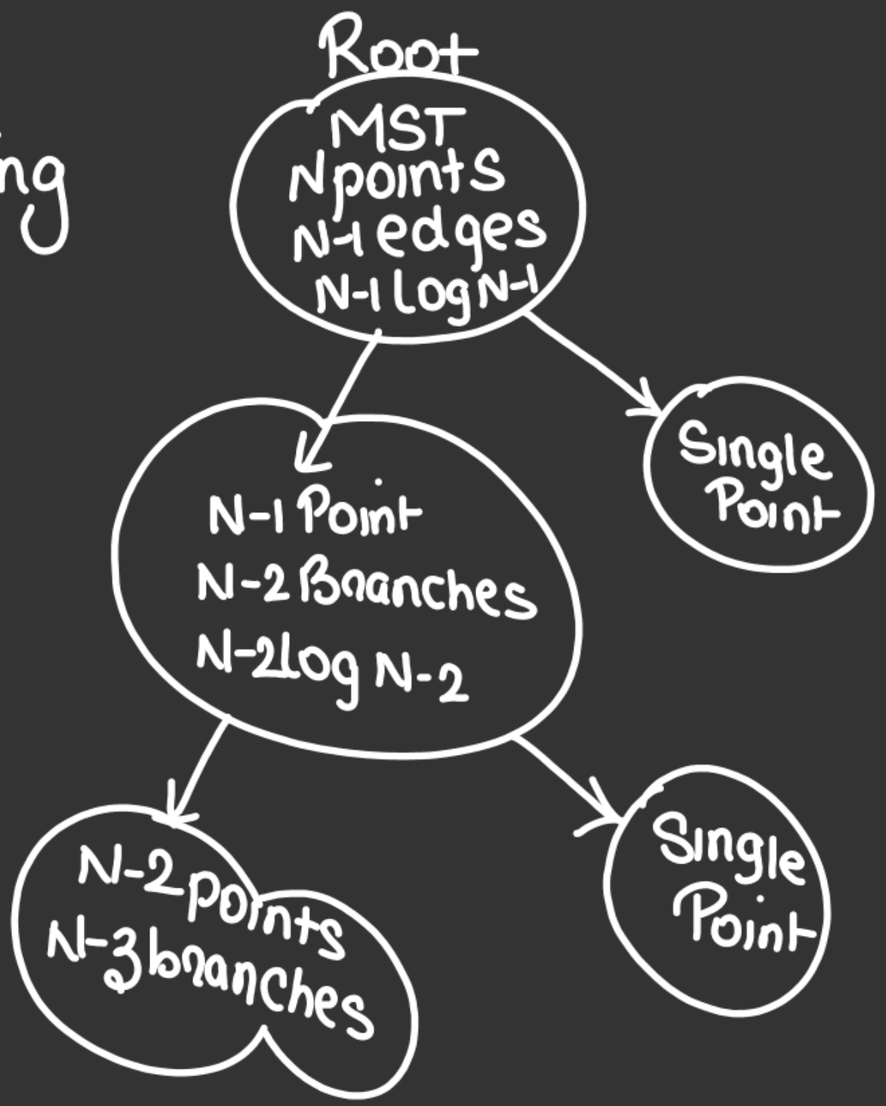
Divisive Clustering using MST





	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

Sorting


Root
MST
N points
N-1 edges
N-1 log N-1
Single Point
N-1 Point
N-2 Branches
N-2 log N-2
N-2 points
N-3 branches
Single Point

Clustering

Hierarchical Vs Flat

Aspect	Hierarchical Clustering	Flat Clustering
Methodology	Builds a hierarchy of clusters	Partitions data into a set number of clusters
Types	Agglomerative (bottom-up), Divisive (top-down)	K-means, K-medoids, etc.
Cluster Number	Does not require a predefined number of clusters	Requires a predefined number of clusters
Complexity	Typically more computationally intensive	Generally less computationally intensive
Flexibility	Can provide more flexible clustering	Less flexible due to predefined number of clusters
Visualization	Produces a dendrogram to visualize the clustering process	No hierarchical structure visualization; can use scatter plots
Merge/Split Criteria	Uses linkage criteria for merging/splitting clusters	Uses centroid or medoid to define cluster centers

Clustering

Hierarchical Vs Flat

Optimal Number of Clusters	Can determine optimal number of clusters using dendrogram	Requires methods like Elbow or Silhouette for optimal number
Data Size Suitability	Suitable for smaller datasets due to computational demands	Suitable for larger datasets
Handling Noise and Outliers	Can be sensitive to noise and outliers	Can be more robust depending on the algorithm used
Result Interpretability	Easier to interpret hierarchical relationships	Interpretation depends on cluster centroids/medoids
Examples	Agglomerative Clustering, Divisive Clustering	K-means, K-medoids, DBSCAN
Initialization Dependence	Not dependent on initial cluster centers	Can be sensitive to initial cluster center selection

1. **What is divisive clustering?**

- A) A bottom-up hierarchical clustering method
- B) A top-down hierarchical clustering method
- C) A method based on centroid calculation
- D) A density-based clustering method

2. **Which of the following is a key characteristic of divisive clustering?**

- A) It starts with individual points and merges them into clusters.
- B) It starts with a single cluster containing all points and splits it into smaller clusters.
- C) It relies on the density of data points to form clusters.
- D) It uses a fixed number of clusters from the beginning.

3. **In divisive clustering, what is the main criterion for splitting a cluster?**

- A) Minimizing the total within-cluster variance
- B) Maximizing the total within-cluster variance
- C) Minimizing the distance between data points
- D) Maximizing the number of clusters

6. **Divisive clustering can be computationally expensive because:**

- A) It requires merging clusters repeatedly.
- B) It needs to compute distances between all pairs of points.
- C) It must split clusters recursively.
- D) It uses complex statistical models.

7. **What is a potential drawback of divisive clustering?**

- A) It is not suitable for large datasets.
- B) It tends to form clusters with equal sizes.
- C) It cannot handle non-linear boundaries.
- D) It does not consider the shape of the data distribution.

10. **In divisive clustering, when a cluster is split, the resulting sub-clusters are:**

- A) Always of equal size
- B) Determined based on predefined conditions
- C) Formed to maximize intra-cluster similarity
- D) Randomly assigned

Clustering

Why Agglomerative is more Famous?

- **Simplicity and Intuition**:
- **Agglomerative Clustering**: This method starts with each data point as its own cluster and merges pairs of clusters step by step until a single cluster is formed or a desired number of clusters is reached. This bottom-up approach is easy to understand and implement.
- **Divisive Clustering**: This method starts with all data points in one cluster and recursively splits them into smaller clusters. The top-down approach can be conceptually more complex and harder to implement efficiently.

Clustering

Why Agglomerative is more Famous?

- Computational Efficiency:

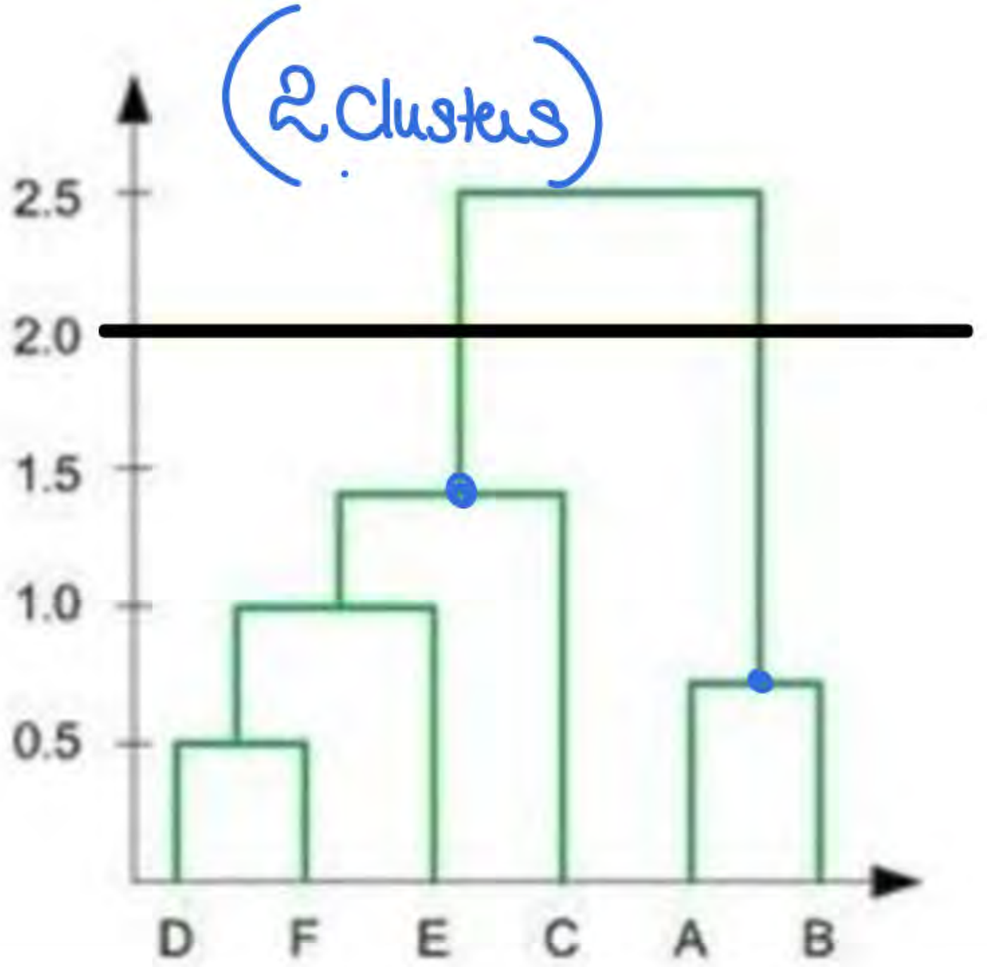
- Agglomerative Clustering: Although both methods can be computationally expensive, agglomerative clustering is generally considered more computationally feasible. It requires fewer decisions at each step (merging pairs of clusters) compared to divisive clustering, which involves finding the best way to split a cluster.
- xDivisive Clustering: This method can be very computationally intensive, especially for large datasets, because it involves evaluating numerous possible ways to split clusters at each step.

1) What is true about K-Mean Clustering?

1. K-means is extremely sensitive to cluster center initializations
2. Bad initialization can lead to Poor convergence speed
3. Bad initialization can lead to bad overall clustering

a. 1 and 2
b. 1 and 3
c. All of the above
d. 2 and 3

In the figure below, if you draw a horizontal line on y-axis for y=2. What will be the number of clusters formed?



Assume, you want to cluster 7 observations into 3 clusters using K-Means clustering algorithm. After first iteration the clusters: C1, C2, C3 has the following observations:

C1: {(1,1), (4,4), (7,7)}

C2: {(0,4), (4,0)}

C3: {(5,5), (9,9)}

What will be the cluster centroids if you want to proceed for second iteration?

a. C1: (4,4), C2: (2,2), C3: (7,7)
b. C1: (2,2), C2: (0,0), C3: (5,5)
c. C1: (6,6), C2: (4,4), C3: (9,9)
d. None of these

Following Question 5, what will be the Manhattan distance for observation (9, 9) from cluster centroid C1 in the second iteration?

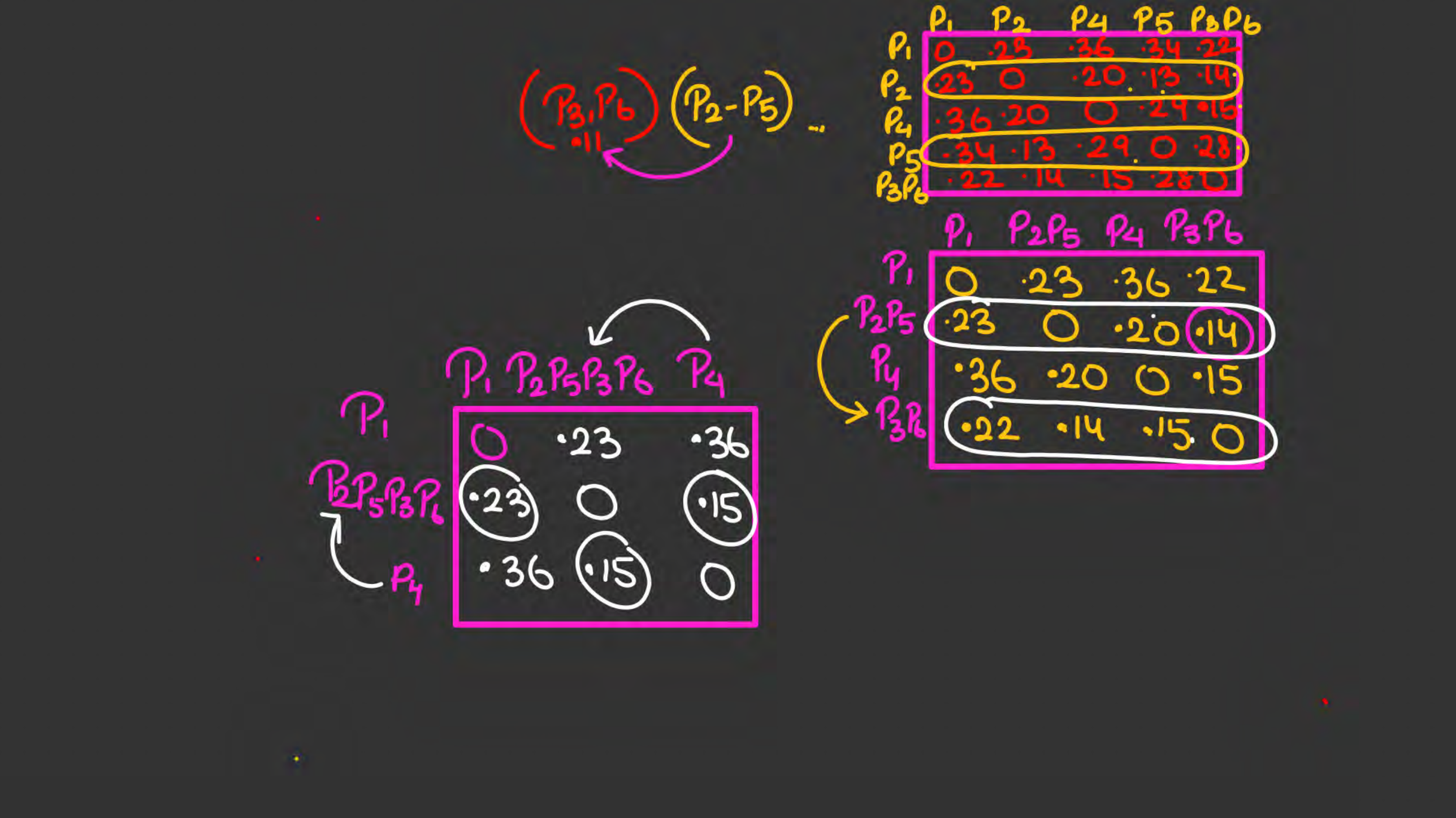
a. 10
b. 5
c. 6
d. 7

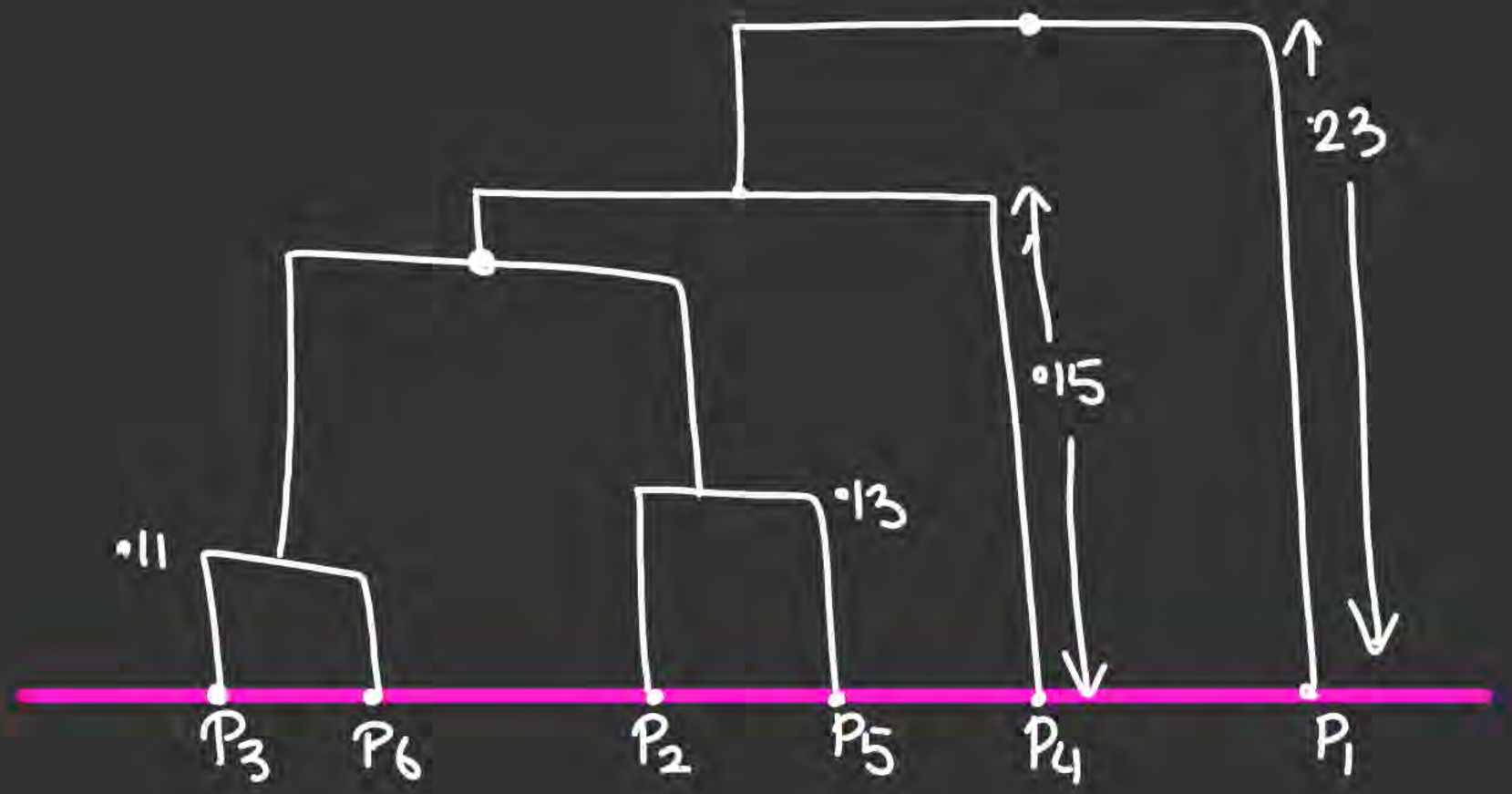
p2	0.2148	0.3854
p3	0.3457	0.3156
p4	0.2652	0.1875
p5	0.0789	0.4139
p6	0.4548	0.3022

(P_3, P_6) $(P_2 - P_5)$
·11

	P_1	P_2	P_4	P_5	P_3P_6
P_1	0	·23	·36	·34	·22
P_2	·23	0	·20	·13	·14
P_4	·36	·20	0	·29	·15
P_5	·34	·13	·29	0	·28
P_3P_6	·22	·14	·15	·28	0

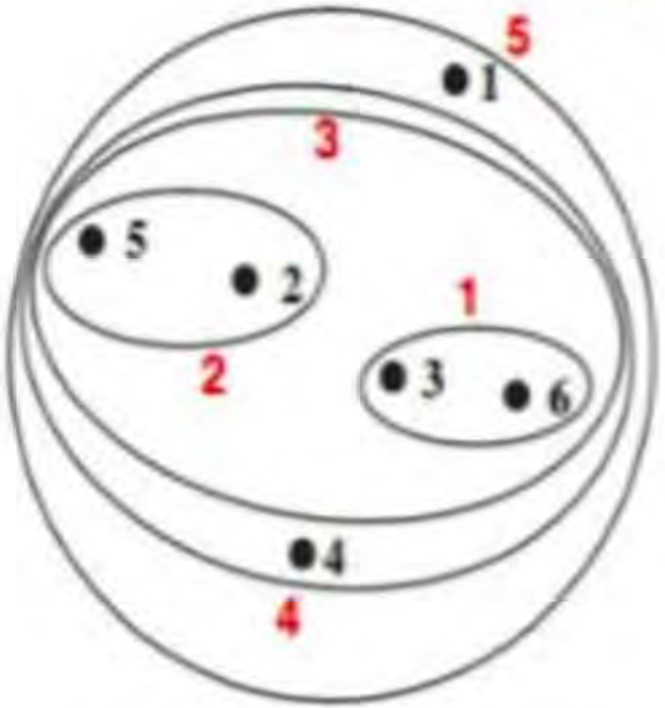
	p1	p2	p3	p4	p5	p6
p1	0.0000	0.2357	0.2218	0.3688	0.3421	0.2347
p2	0.2357	0.0000	0.1483	0.2042	0.1388	0.2540
p3	0.2218	0.1483	0.0000	0.1513	0.2843	0.1100
p4	0.3688	0.2042	0.1513	0.0000	0.2932	0.2216
p5	0.3421	0.1388	0.2843	0.2932	0.0000	0.3921
p6	0.2347	0.2540	0.1100	0.2216	0.3921	0.0000

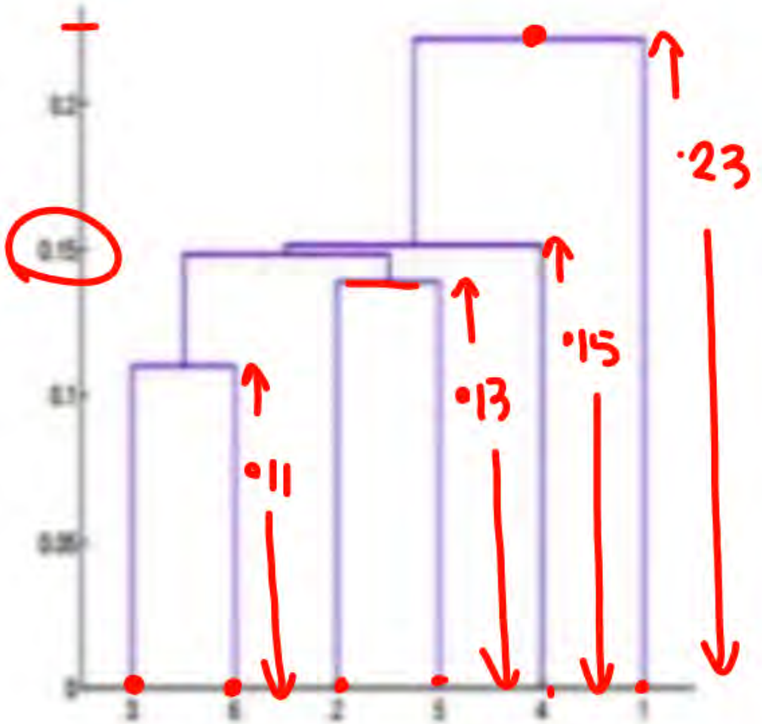

P1 P2 P4 P5 P3P6
P1 0 ·23 ·36 ·34 ·22
P2 ·23 0 ·20 ·13 ·14
P4 ·36 ·20 0 ·29 ·15
P5 ·34 ·13 ·29 0 ·28
P3P6 ·22 ·14 ·15 ·28 0
(P3,P6) ·11
(P2-P5)
P1 P2P5 P4 P3P6
P1 0 ·23 ·36 ·22
P2P5 ·23 0 ·20 ·14
P4 ·36 ·20 0 ·15
P3P6 ·22 ·14 ·15 0
P1 P1 P2P5P3P6 P4
P2P5P3P6 ·23 0 ·15
P4 ·36 ·15 0


·23
·15
·13
·11
P3
P6
P2
P5
P4
P1

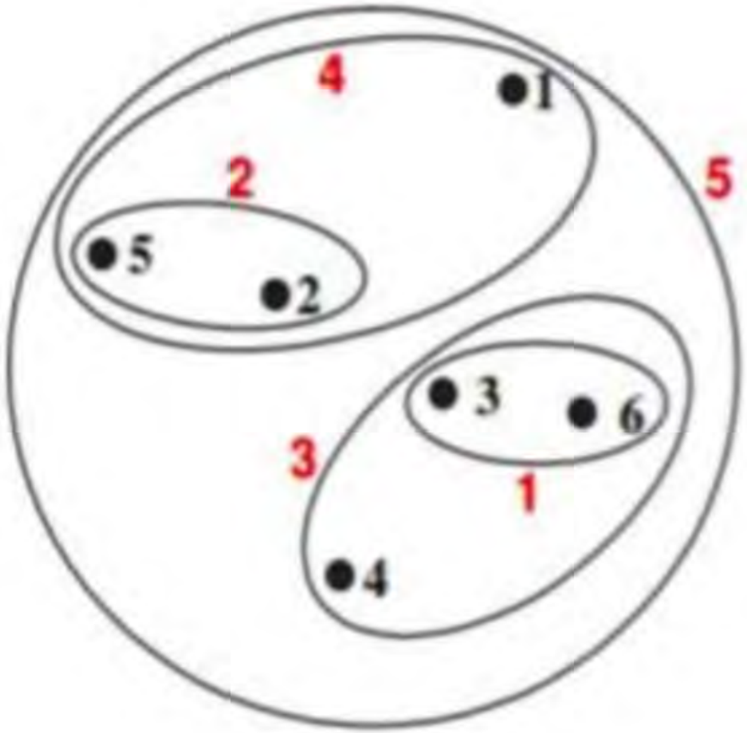
Which of the following clustering representations and dendrogram depicts the use of MIN or Single link proximity function in hierarchical clustering:

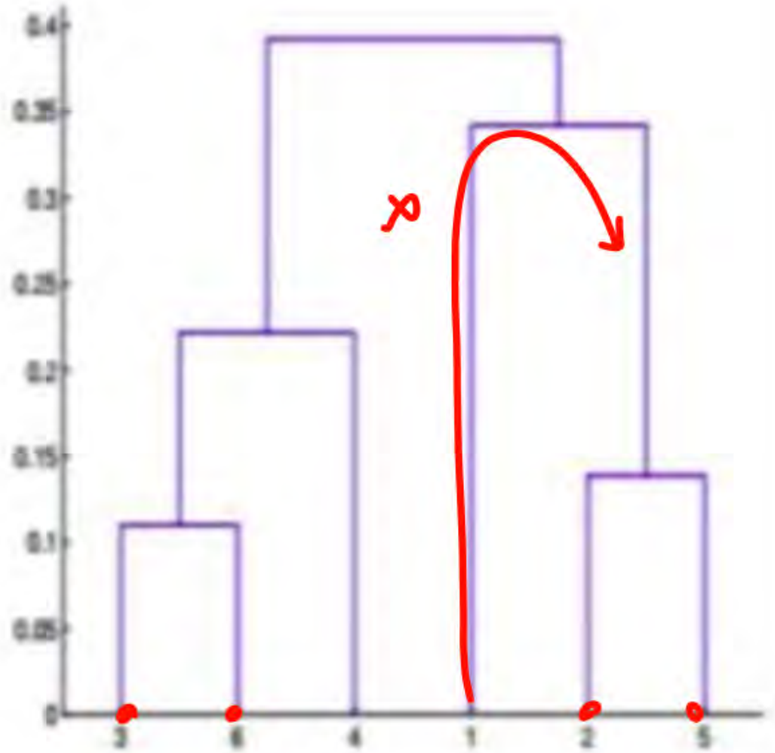
A.



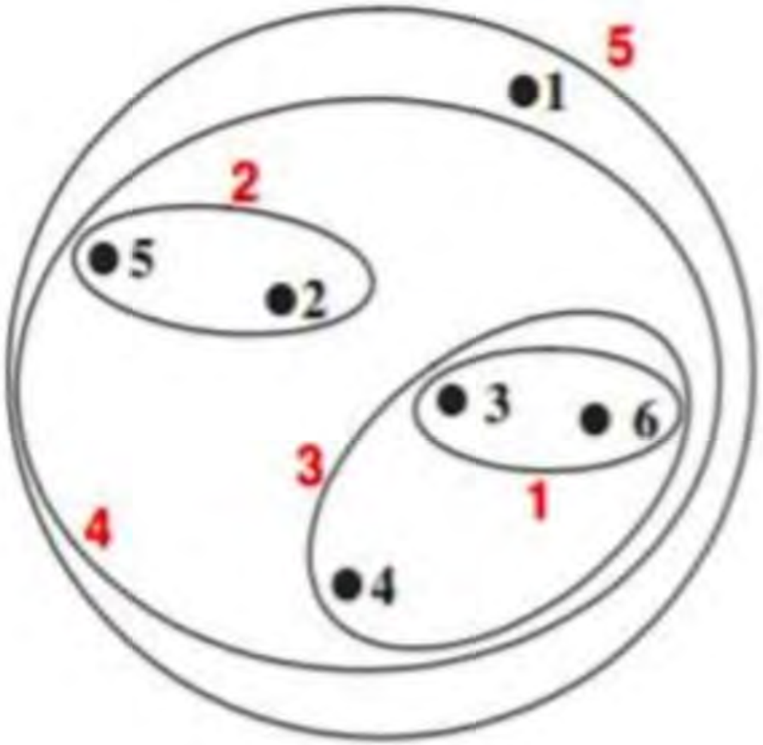


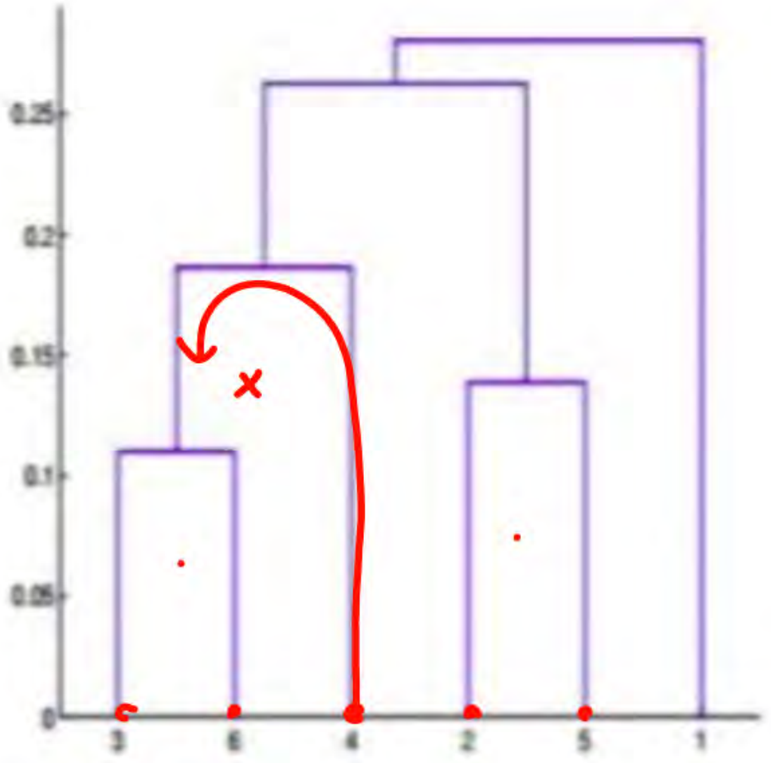
B.



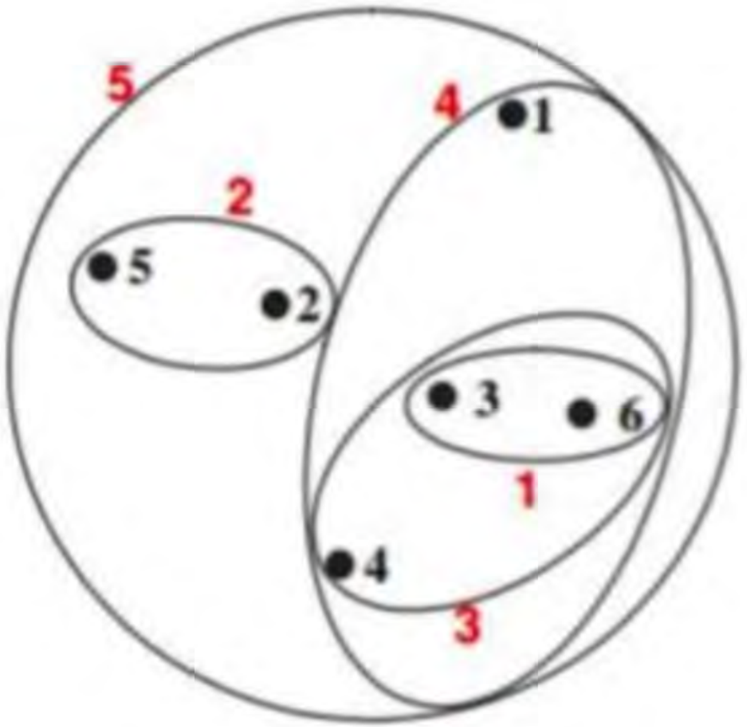


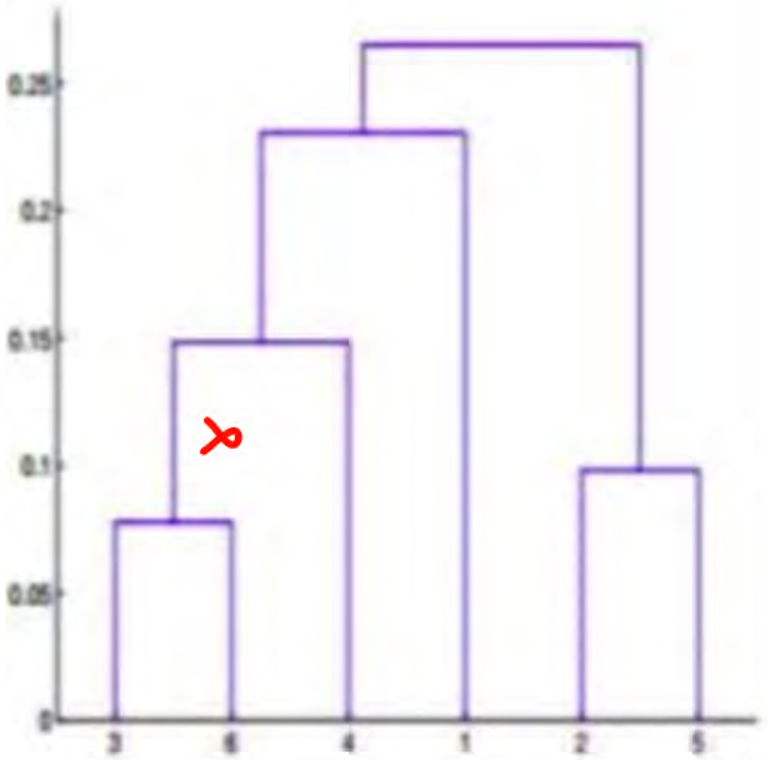
C.





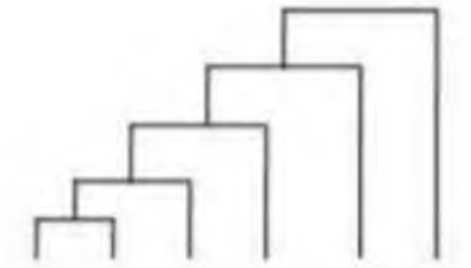
D.



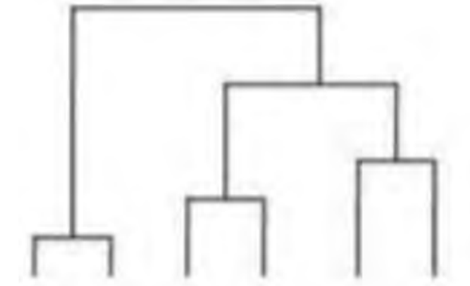


The pairwise distance between 6 points is given below. Which of the option shows the hierarchy of clusters created by single link clustering algorithm?

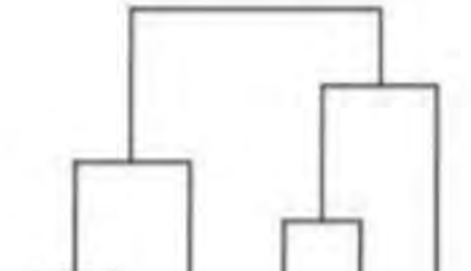
	P1	P2	P3	P4	P5	P6
P1	0	3	8	9	5	4
P2	3	0	9	8	10	9
P3	8	9	0	1	6	7
P4	9	8	1	0	7	8
P5	5	10	6	7	0	2
P6	4	9	7	8	2	0



(a)



(b)

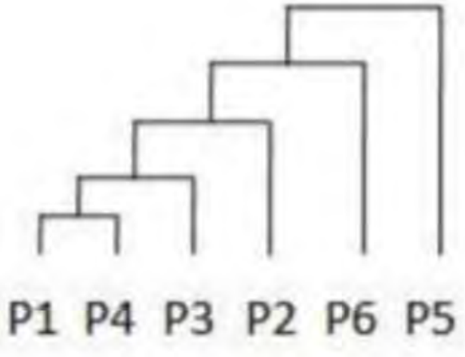


(c)

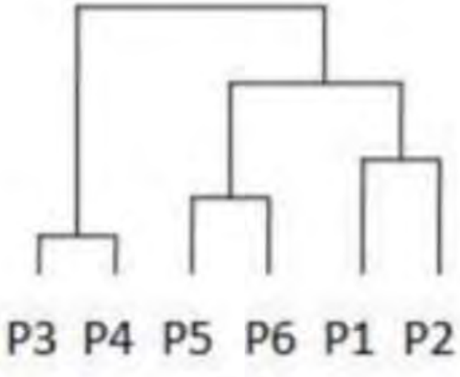
(d)

For the pairwise distance matrix given in the previous question, which of the following shows the hierarchy of clusters created by the complete link clustering algorithm.

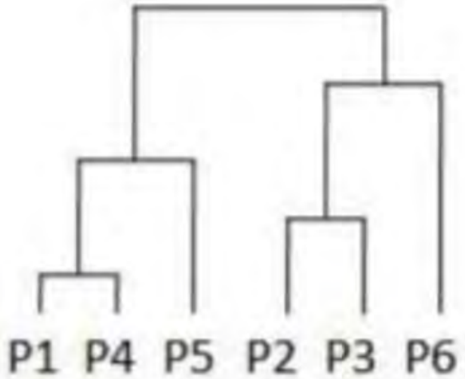
	P1	P2	P3	P4	P5	P6
P1	0	3	8	9	5	4
P2	3	0	9	8	10	9
P3	8	9	0	1	6	7
P4	9	8	1	0	7	8
P5	5	10	6	7	0	2
P6	4	9	7	8	2	0



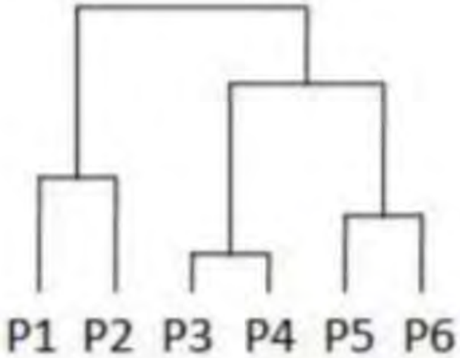
(a)



(b)



(c)



(d)

THANK - YOU