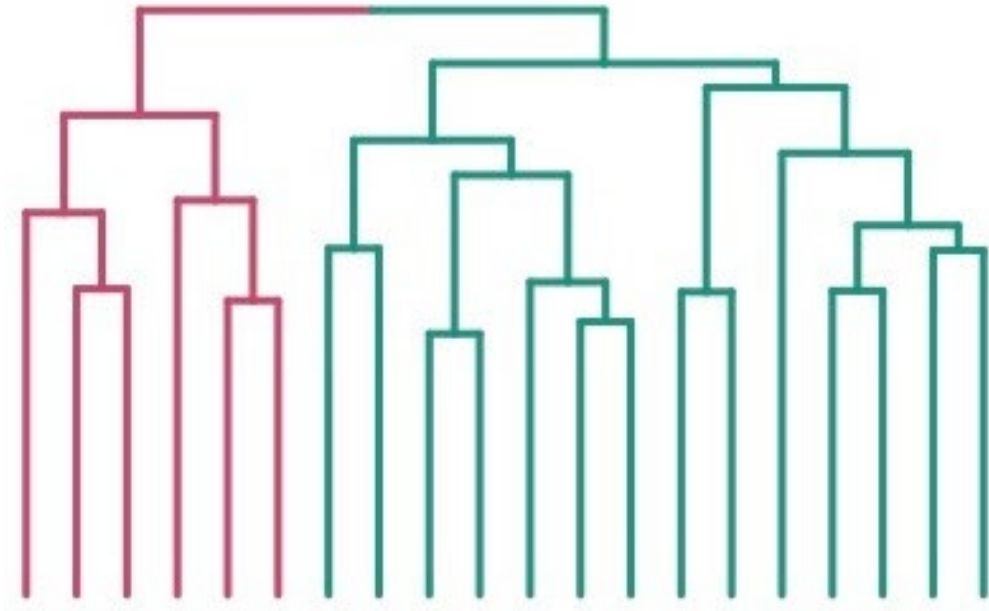




Unsupervised ML part II : Hierarchical Clustering



อ.ดร.ปัญญานต์ อ้นพงษ์

ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

aonpong_p@su.ac.th

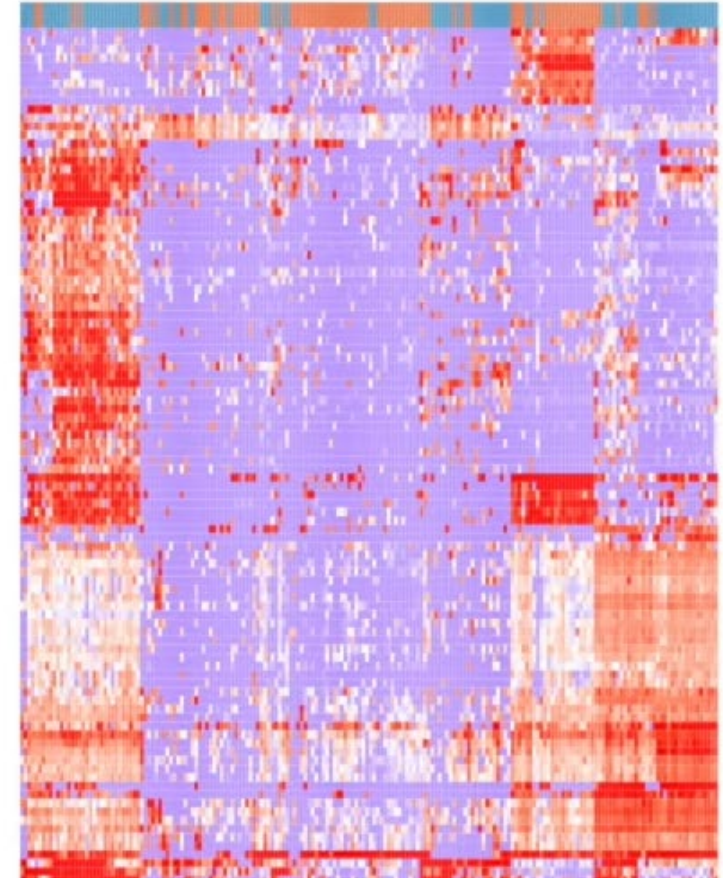
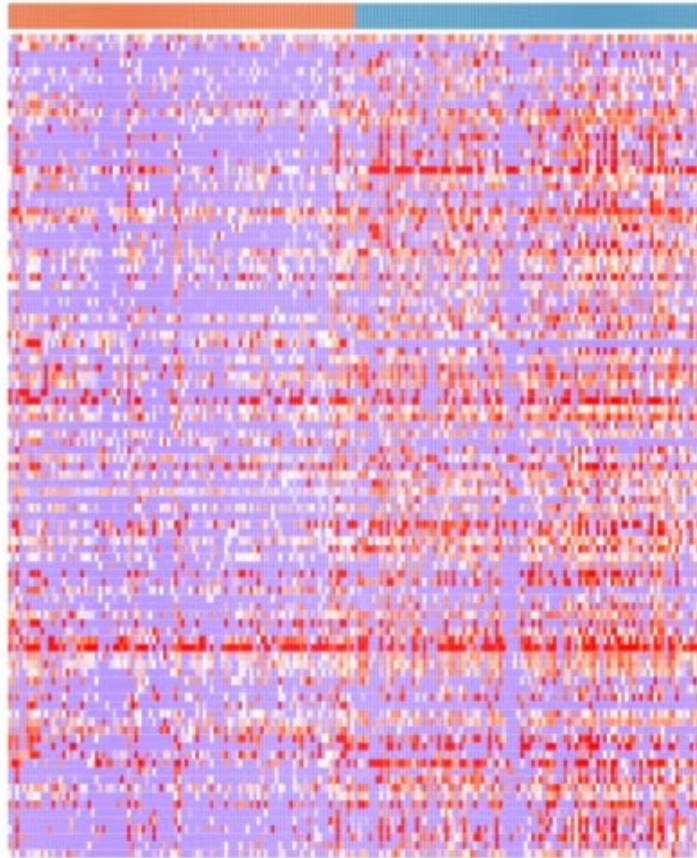
- Hierarchical Clustering
 - แนวคิดของ Hierarchical Clustering
 - Agglomerative
 - กระบวนการ
 - การอ่าน Dendrogram
 - Divisive
 - Minimum Spanning Tree
 - กระบวนการ

Idea of Hierarchical Clustering

- Hierarchical Clustering มีการแบ่งทั้งแบบ Agglomerative และ Divisive แตกต่างจาก k-mean ที่เป็น divisive เพียงอย่างเดียว
- เป็นการจัดกลุ่มโดยไม่ต้องมีการกำหนดจำนวนกลุ่มที่ต้องการจัดกลุ่มข้อมูลก่อน (ค่อย ๆ แบ่ง (Agnes) หรือค่อย ๆ รวม (Diana) ไปจนกว่ากลุ่มที่ได้จะเมคเซ็นส์)



Idea of Hierarchical Clustering



Outline



- Hierarchical Clustering
 - แนวคิดของ Hierarchical Clustering
 - **Agglomerative**
 - กระบวนการ
 - การอ่าน Dendrogram
 - Divisive
 - Minimum Spanning Tree
 - กระบวนการ

Agglomerative



- มีชื่อเล่นว่า Agnes
- ทำงานโดยเริ่มจากกำหนดคลัสเตอร์จำนวนมาก (เท่าจำนวนข้อมูล) แล้วค่อย ๆ เพิ่มขนาดของคลัสเตอร์ขึ้น โดยการจับกลุ่มข้อมูลที่มีความใกล้เคียงกันมากที่สุดเข้าด้วยกัน
- เมื่อทำซ้ำไปเรื่อย ๆ คลัสเตอร์ก็จะมีขนาดใหญ่ขึ้น และมีจำนวนน้อยลง

กระบวนการ Agglomerative



Hierarchical Clustering สมมติข้อมูลที่จะใช้เป็นดังนี้

	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6
รหัสพันธุ์กรรม 1	10	11	8	3	2	1
รหัสพันธุ์กรรม 2	6	4	5	3	2.8	1
รหัสพันธุ์กรรม 3	12	9	10	2.5	1.3	2

และเราต้องการจับกลุ่มรหัสพันธุ์กรรม

กระบวนการ Agglomerative

Hierarchical Clustering

1. จับคู่หาระยะห่างของรหัสพันธุกรรมแต่ละตัว โดยใช้สมการ Distance

Gene1 \Leftrightarrow Gene2 จะได้ว่า $(10 - 6)^2 + (11 - 4)^2 + (8 - 5)^2 + (3 - 3)^2 + (2 - 2.8)^2 + (1 - 1)^2 = 74.64$

Gene2 \Leftrightarrow Gene3 จะได้ว่า $(6 - 12)^2 + (4 - 9)^2 + (5 - 10)^2 + (3 - 2.5)^2 + (2.8 - 1.3)^2 + (1 - 2)^2 = 89.5$

Gene1 \Leftrightarrow Gene3 จะได้ว่า $(10 - 12)^2 + (11 - 9)^2 + (8 - 10)^2 + (3 - 2.5)^2 + (2 - 1.3)^2 + (1 - 2)^2 = 13.74$

	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6
รหัสพันธุกรรม 1	10	11	8	3	2	1
รหัสพันธุกรรม 2	6	4	5	3	2.8	1
รหัสพันธุกรรม 3	12	9	10	2.5	1.3	2

กระบวนการ Agglomerative

Hierarchical Clustering

1. จับคู่หาระยะห่างของรหัสพันธุกรรมแต่ละตัว โดยใช้สมการ Distance
2. เลือกคู่ที่มีระยะห่างต่อกันน้อยที่สุด

$$\text{Gene1} \Leftrightarrow \text{Gene2} \text{ จะได้ว่า } (10 - 6)^2 + (11 - 4)^2 + (8 - 5)^2 + (3 - 3)^2 + (2 - 2.8)^2 + (1 - 1)^2 = 74.64$$

$$\text{Gene2} \Leftrightarrow \text{Gene3} \text{ จะได้ว่า } (6 - 12)^2 + (4 - 9)^2 + (5 - 10)^2 + (3 - 2.5)^2 + (2.8 - 1.3)^2 + (1 - 2)^2 = 89.5$$

$$\text{Gene1} \Leftrightarrow \text{Gene3} \text{ จะได้ว่า } (10 - 12)^2 + (11 - 9)^2 + (8 - 10)^2 + (3 - 2.5)^2 + (2 - 1.3)^2 + (1 - 2)^2 = 13.74$$

กระบวนการ Agglomerative

Hierarchical Clustering

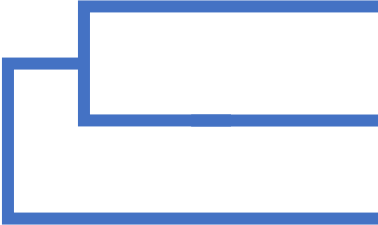
1. จับคู่หาระยะห่างของรหัสพันธุกรรมแต่ละตัว โดยใช้สมการ Distance
2. เลือกคู่ที่มีระยะห่างต่อกันน้อยที่สุด
3. ย้ายสองตัวดังกล่าวมาไว้ติดกันและเขียนเส้นเชื่อมโยง (อาจเขียน Distance กำกับไว้ด้วยก็ได้)

13.74			หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6
		รหัสพันธุกรรม 1	10	11	8	3	2	1
		รหัสพันธุกรรม 3	12	9	10	2.5	1.3	2
		รหัสพันธุกรรม 2	6	4	5	3	2.8	1

กระบวนการ Agglomerative

Hierarchical Clustering

1. จับคู่หาระยะห่างของรหัสพันธุกรรมแต่ละตัว โดยใช้สมการ Distance
2. เลือกคู่ที่มีระยะห่างต่อกันน้อยที่สุด
3. ย้ายสองตัวดังกล่าวมาไว้ติดกันและเขียนเส้นเชื่อมโยง (อาจเขียน Distance กำกับไว้ด้วยก็ได้)
4. มองข้อมูลในคลัสเตอร์ที่เชื่อมโยงกันเป็นข้อมูลตัวเดียว จากนั้นทำข้อ 1-3 ซ้ำใหม่จนสร้างเส้นเชื่อมครบทุกข้อมูลที่มี

		หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6
	รหัสพันธุกรรม 1	10	11	8	3	2	1
	รหัสพันธุกรรม 3	12	9	10	2.5	1.3	2
	รหัสพันธุกรรม 2	6	4	5	3	2.8	1

กระบวนการ Agglomerative

Hierarchical Clustering

คำถาม ถ้าต้องมองทั้งกลุ่มที่จับไปแล้วเป็นข้อมูลตัวเดียว แล้วเราจะหา Distance ยังไง?



	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6
รหัสพันธุ์กรรม 1	10	11	8	3	2	1
รหัสพันธุ์กรรม 3	12	9	10	2.5	1.3	2
รหัสพันธุ์กรรม 2	6	4	5	3	2.8	1

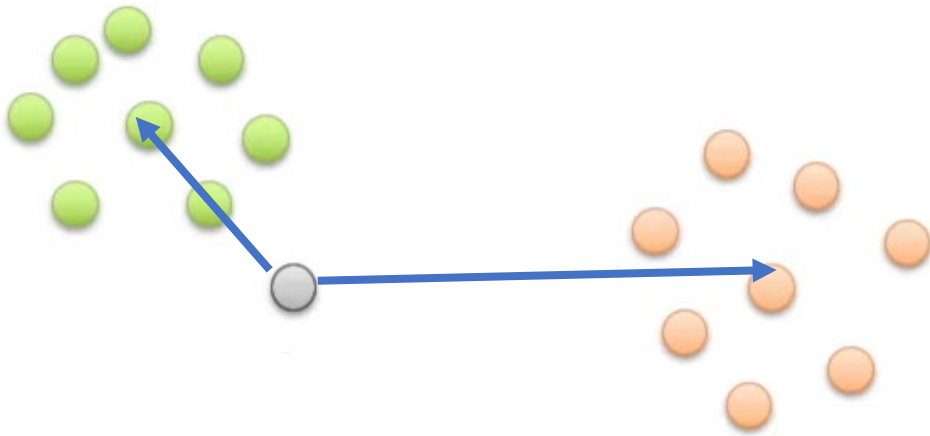
กระบวนการ Agglomerative

Hierarchical Clustering

คำถาม ถ้าต้องมองทั้งกลุ่มที่จับไปแล้วเป็นข้อมูลตัวเดียว แล้วเราจะหา Distance ยังไง?

คำตอบ มีวิธีดูหลายแบบ!

1. ดูค่าเฉลี่ยของทั้งคลัสเตอร์ (Centroid)



	หนู 1	หนู ...
รหัสพันธุ์กรรม 1	10	...
รหัสพันธุ์กรรม 3	11 12	...
รหัสพันธุ์กรรม 2	6	...

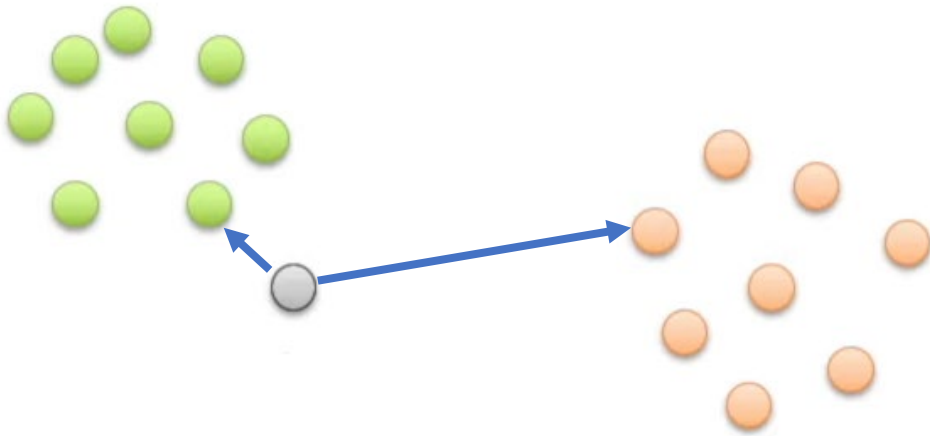
กระบวนการ Agglomerative

Hierarchical Clustering

คำถาม ถ้าต้องมองทั้งกลุ่มที่จับไปแล้วเป็นข้อมูลตัวเดียว แล้วเราจะหา Distance ยังไง?

คำตอบ มีวิธีดูหลายแบบ!

1. ดูค่าเฉลี่ยของทั้งคลัสเตอร์ (Centroid)
2. ดูค่าใกล้ที่สุด (Single-Linkage)



	หนู 1	หนู ...
รหัสพันธุ์กรรม 1	10	...
รหัสพันธุ์กรรม 3	12	...
รหัสพันธุ์กรรม 2	6	...

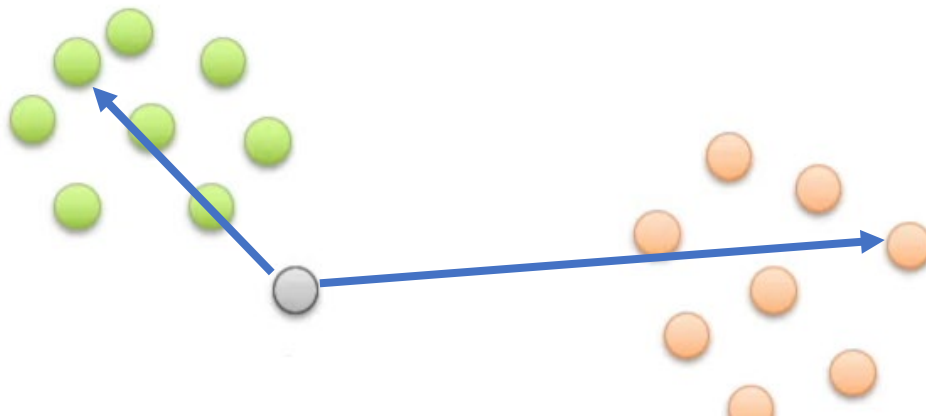
กระบวนการ Agglomerative

Hierarchical Clustering

คำถาม ถ้าต้องมองทั้งกลุ่มที่จับไปแล้วเป็นข้อมูลตัวเดียว แล้วเราจะหา Distance ยังไง?

คำตอบ มีวิธีดูหลายแบบ!

1. ดูค่าเฉลี่ยของทั้งคลัสเตอร์ (Centroid)
2. ดูค่าใกล้ที่สุด (Single-Linkage)
3. ดูค่าไกลที่สุด (Complete-Linkage)



	หนู 1	หนู ...
รหัสพันธุ์กรรม 1	10	...
รหัสพันธุ์กรรม 3	12	...
รหัสพันธุ์กรรม 2	6	...

กระบวนการ Agglomerative

คำถาม ถ้าต้องมองทั้งกลุ่มที่จับไปแล้วเป็นข้อมูลตัวเดียว แล้วเราจะหา Distance ยังไง?

คำตอบ มีวิธีดูหลายแบบ!

1. ดูค่าเฉลี่ยของทั้งคลัสเตอร์ (Centroid)
2. ดูค่าใกล้ที่สุด (Single-Linkage)
3. ดูค่าไกลที่สุด (Complete-Linkage)
4. ดู Sum of square น้อยที่สุดของคลัสเตอร์ทั้งหมด (Ward)



	หนู 1	หนู ...
รหัสพันธุ์กรรม 1	10	...
รหัสพันธุ์กรรม 3	12	...
รหัสพันธุ์กรรม 2	6	...

กระบวนการ Agglomerative

คำถาม ควรเลือกวิธีไหน?

1. ค่าเฉลี่ยของทั้งคลัสเตอร์ (Centroid)
2. ค่าใกล้ที่สุด (Single-Linkage)
3. ค่าไกลที่สุด (Complete-Linkage)
4. ค่า Sum of square น้อยที่สุดของคลัสเตอร์ทั้งหมด (Ward)

กระบวนการ Agglomerative

คำถาม ควรเลือกวิธีไหน?

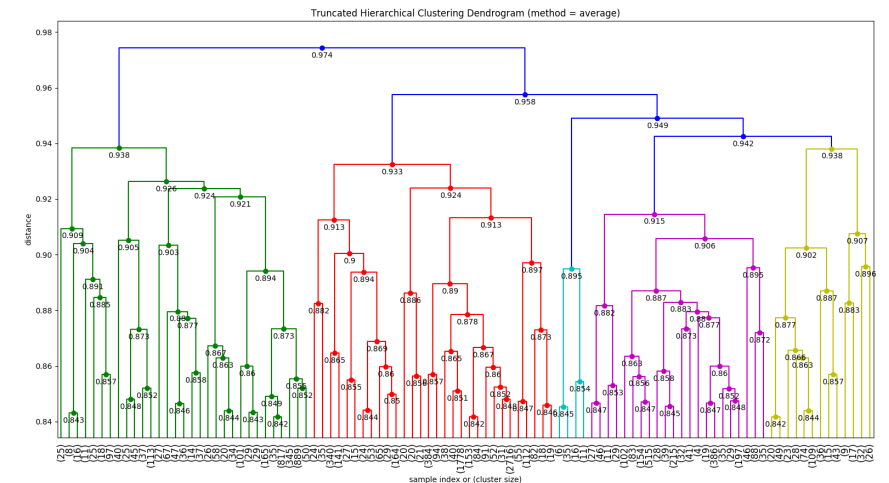
1. ดูค่าเฉลี่ยของทั้งคลัสเตอร์ (Centroid)
2. ดูค่าใกล้ที่สุด (Single-Linkage)
3. ดูค่าไกลที่สุด (Complete-Linkage)
4. ดู Sum of square น้อยที่สุดของคลัสเตอร์ทั้งหมด (Ward)

คำตอบ ไม่มีคำตอบที่เหมาะสมที่สุด แล้วแต่ลักษณะของการทำงาน (เลือกโดยการทดลอง)

กระบวนการ Agglomerative

- อย่างไรก็ตามกราฟเชื่อมโยงในลักษณะนี้เรียกว่า Dendrogram มันมีหน้าที่เชื่อมต่อข้อมูลที่คล้ายกันไว้ด้วยกัน เพื่อให้รองรับการอ่านเพื่อแบ่งคลัสเตอร์

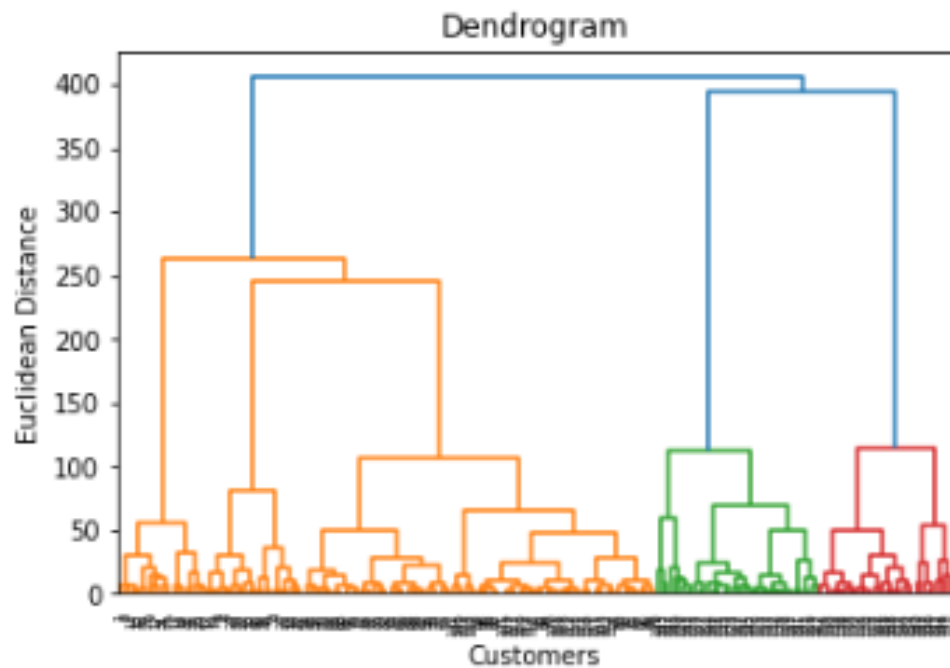
	หนู 1	หนู 2	หนู 3	หนู 4	หนู 5	หนู 6
รหัสพันธุ์กรรม 1	10	11	8	3	2	1
รหัสพันธุ์กรรม 3	12	9	10	2.5	1.3	2
รหัสพันธุ์กรรม 2	6	4	5	3	2.8	1



กระบวนการ Agglomerative



Hierarchical Clustering



Outline

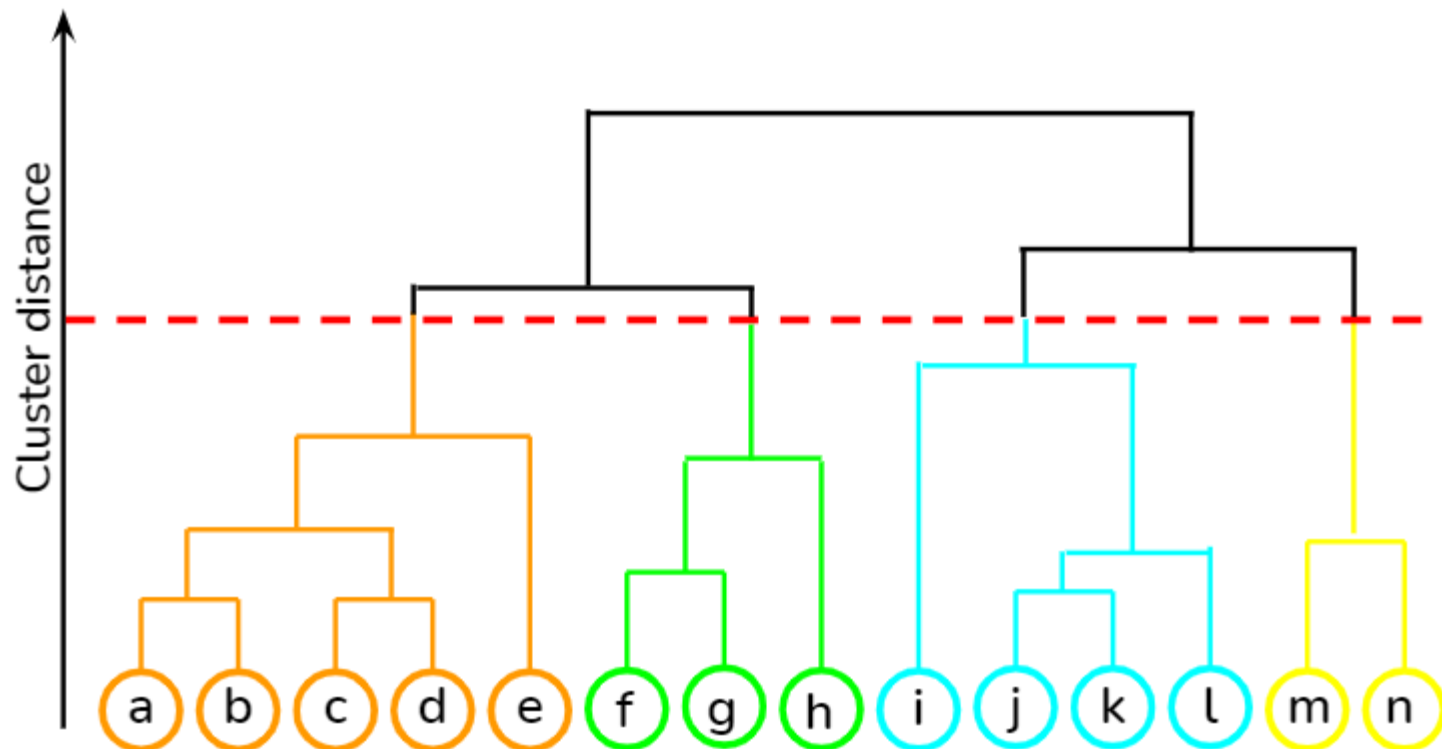


- Hierarchical Clustering
 - แนวคิดของ Hierarchical Clustering
 - **Agglomerative**
 - กระบวนการ
 - การอ่าน Dendrogram
 - Divisive
 - Minimum Spanning Tree
 - กระบวนการ

การอ่าน Dendrogram



- เราสามารถเลือกกลุ่มข้อมูลตามจำนวนคลัสเตอร์ที่ต้องการได้ผ่านการอ่าน Dendrogram
 - แค่เลื่อนเส้นสีแดงขึ้นลง ให้ได้จำนวนคลัสเตอร์ที่ต้องการ ข้อมูลที่ถูกจัดกลุ่มโดยเส้นที่เชื่อมกันจะถือเป็นคลัสเตอร์เดียวกัน



ตัวอย่างโค้ด



```
class sklearn.cluster.AgglomerativeClustering(n_clusters=2, *, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', distance_threshold=None, compute_distances=False)
```

Parameters::

n_clusters : int or None, default=2

The number of clusters to find. It must be `None` if `distance_threshold` is not `None`.

affinity : str or callable, default='euclidean'

Metric used to compute the linkage. Can be "euclidean", "l1", "l2", "manhattan", "cosine", or "precomputed". If linkage is "ward", only "euclidean" is accepted. If "precomputed", a distance matrix (instead of a similarity matrix) is needed as input for the fit method.

ตัวอย่างโค้ด



```
class sklearn.cluster.AgglomerativeClustering(n_clusters=2, *, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', distance_threshold=None, compute_distances=False)
```

Attributes::

n_clusters_ : int

The number of clusters found by the algorithm. If `distance_threshold=None`, it will be equal to the given `n_clusters`.

labels_ : ndarray of shape (n_samples)

Cluster labels for each point.

n_leaves_ : int

Number of leaves in the hierarchical tree.

n_connected_components_ : int

The estimated number of connected components in the graph.

New in version 0.21: `n_connected_components_` was added to replace `n_components_`.

n_features_in_ : int

Number of features seen during `fit`.

New in version 0.24.

ตัวอย่างโค้ด



*class sklearn.cluster.AgglomerativeClustering(n_clusters=2, *, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', distance_threshold=None, compute_distances=False)*

```
>>> from sklearn.cluster import AgglomerativeClustering
>>> import numpy as np
>>> X = np.array([[1, 2], [1, 4], [1, 0],
...               [4, 2], [4, 4], [4, 0]])
>>> clustering = AgglomerativeClustering().fit(X)
>>> clustering
AgglomerativeClustering()
>>> clustering.labels_
array([1, 1, 1, 0, 0, 0])
```

Outline

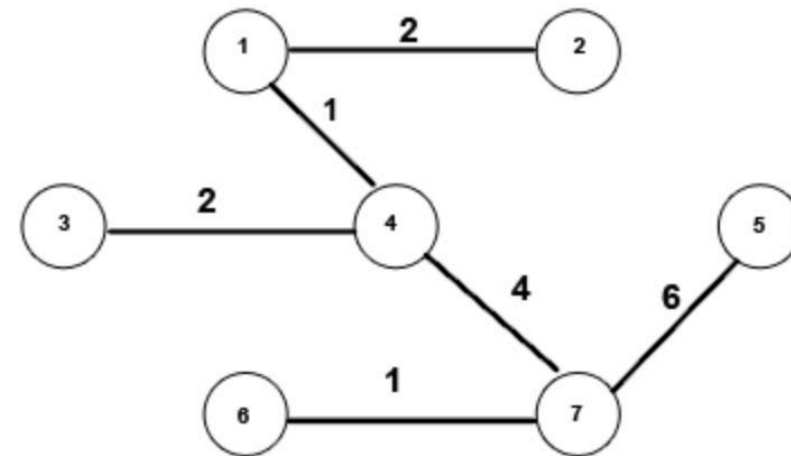
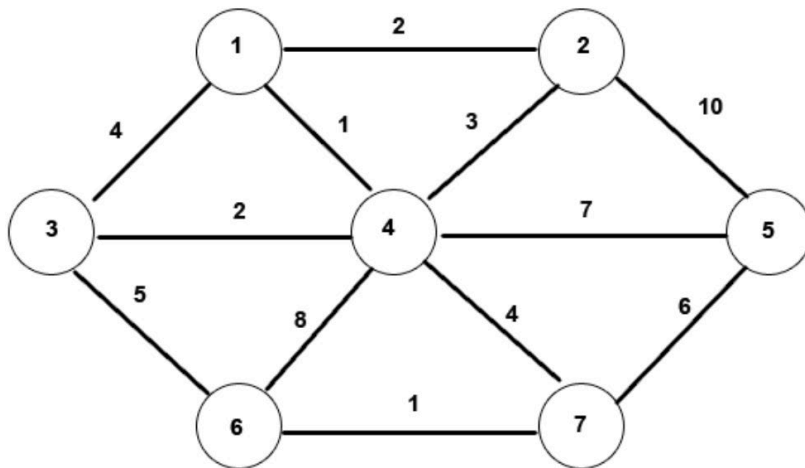


- Hierarchical Clustering
 - แนวคิดของ Hierarchical Clustering
 - Agglomerative
 - กระบวนการ
 - การอ่าน Dendrogram
 - Divisive
 - Minimum Spanning Tree (MST)
 - กระบวนการ

Minimum Spanning Tree

- Minimum Spanning Tree หรือต้นไม้แผ่ทั่ว
- คือการหากิ่งของต้นไม้ที่มีผลรวมน้อยที่สุด ที่สามารถเข้าถึงทุก node ได้

• ขั้นตอนการหา Minimum Spanning Tree



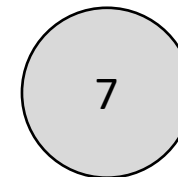
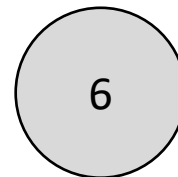
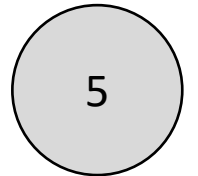
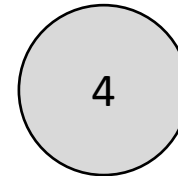
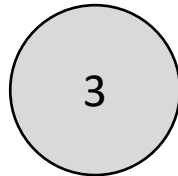
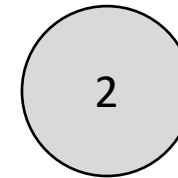
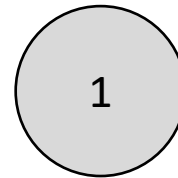
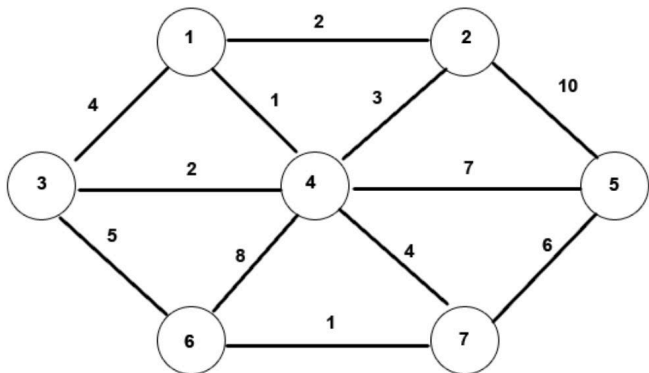
MST: Kruskal's algorithm

- เป็นวิธีการที่นิยม เพราะเรียบง่ายและมีประสิทธิภาพ
- หลักการง่ายมาก เพียงแค่ขีดเส้นที่มี cost น้อยที่สุดไปเรื่อย ๆ
- หากเส้นไหนที่ขีดแล้วจะสร้าง loop ก็ข้ามเส้นนั้นไป
- ทำวนไปเรื่อย ๆ จนทุก node เชื่อมต่อกัน

MST: Kruskal's algorithm

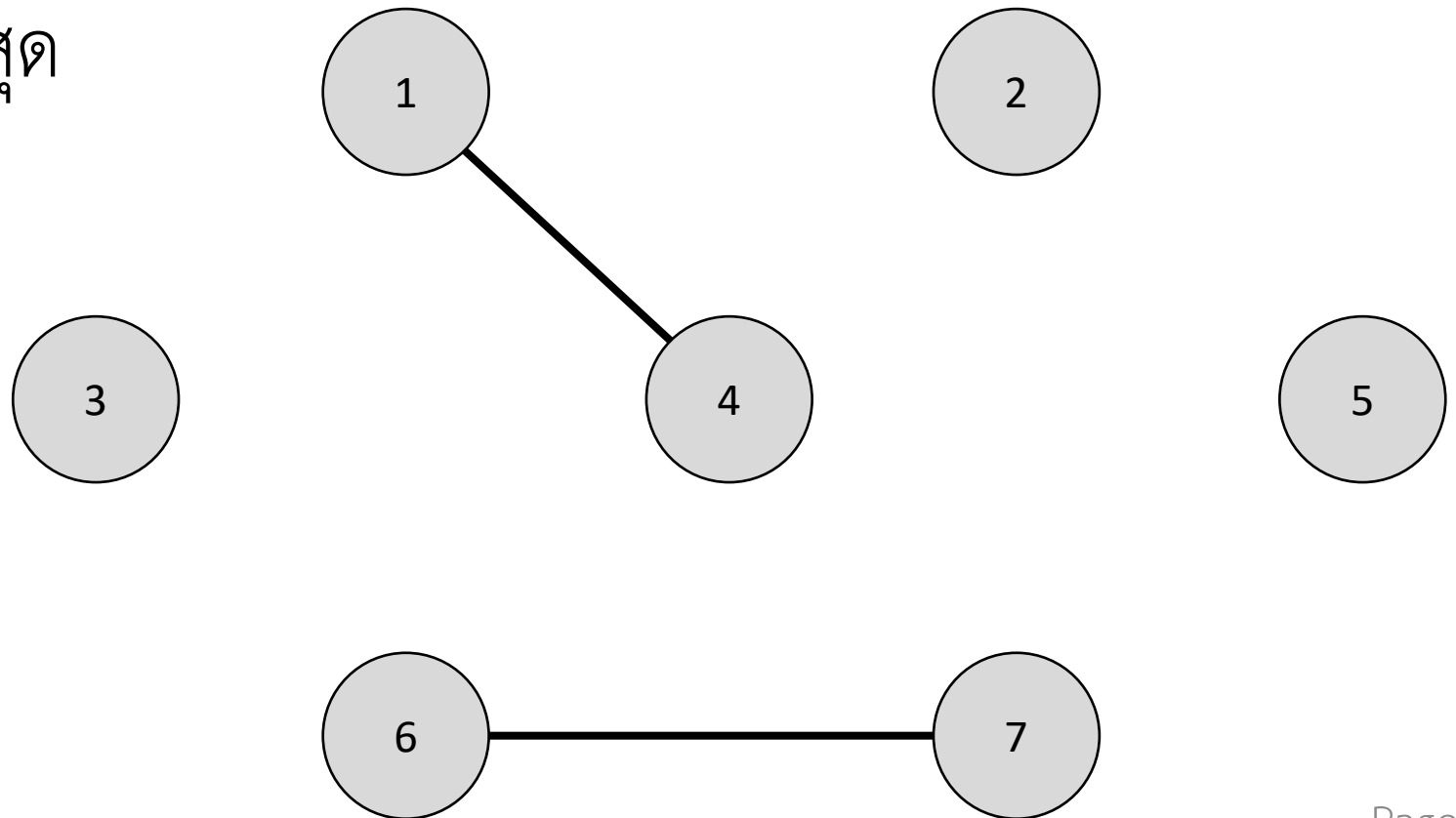
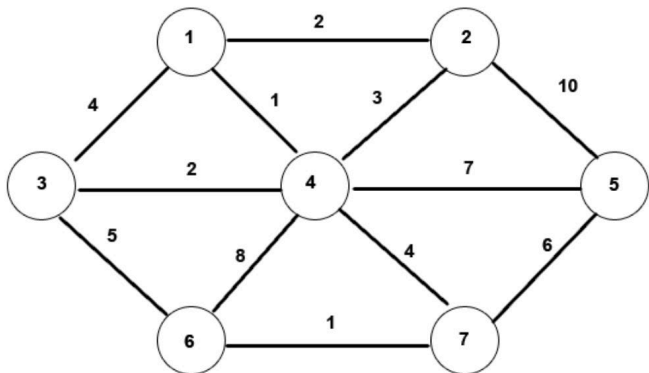


1. สร้าง node เท่าโจทย์



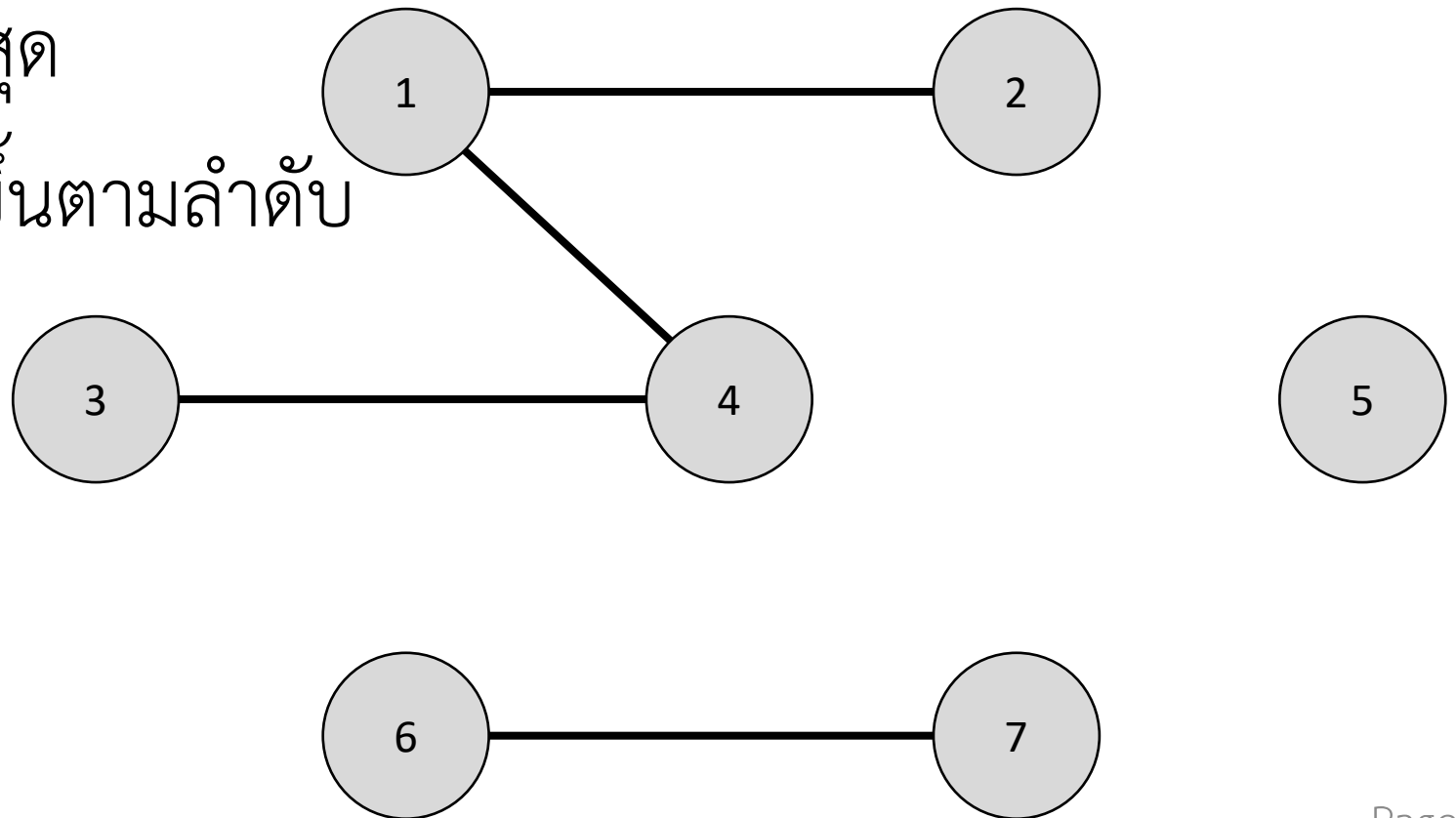
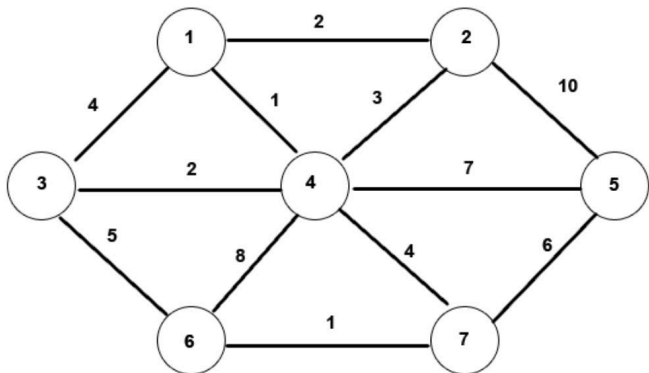
MST: Kruskal's algorithm

1. สร้าง node เท่าโจทย์
2. ขีดเส้นที่ cost น้อยสุด



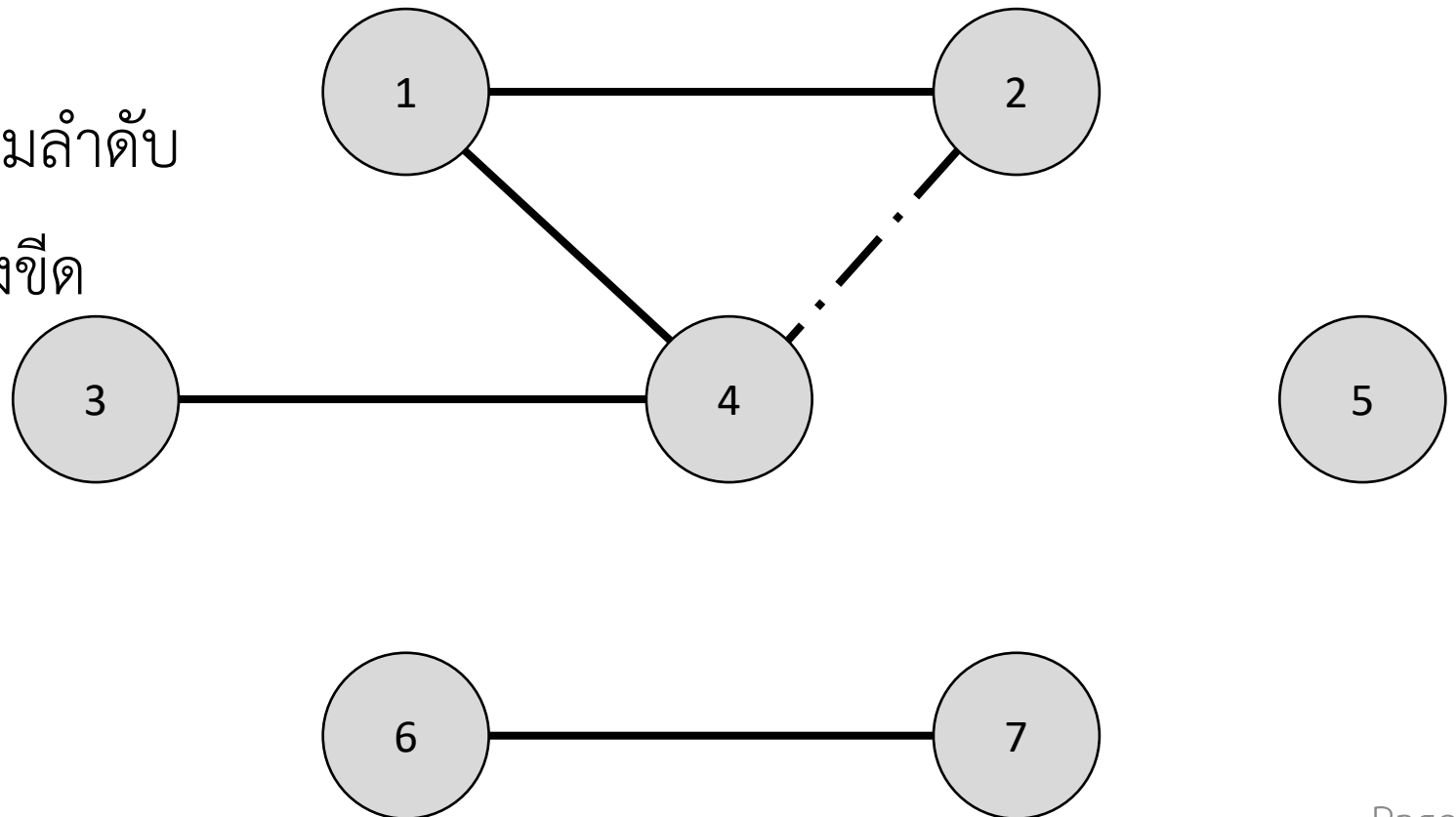
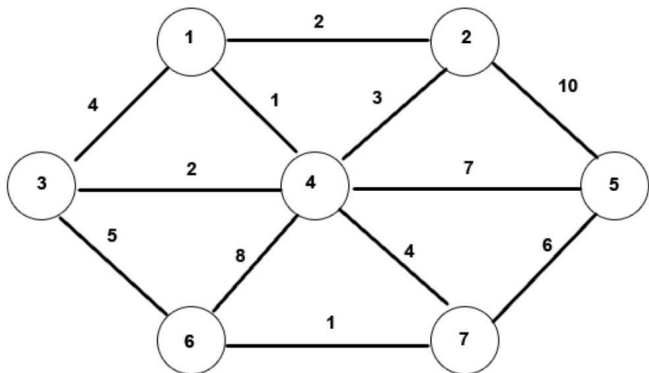
MST: Kruskal's algorithm

1. สร้าง node เท่าโจทย์
2. ขีดเส้นที่ cost น้อยสุด
3. ขีดเส้นที่ cost มากขึ้นตามลำดับ



MST: Kruskal's algorithm

1. สร้าง node เท่าโจทย์
2. ขีดเส้นที่ cost น้อยสุด
3. ขีดเส้นที่ cost มากขึ้นตามลำดับ
4. ถ้าทำให้เกิด loop ไม่ต้องขีด

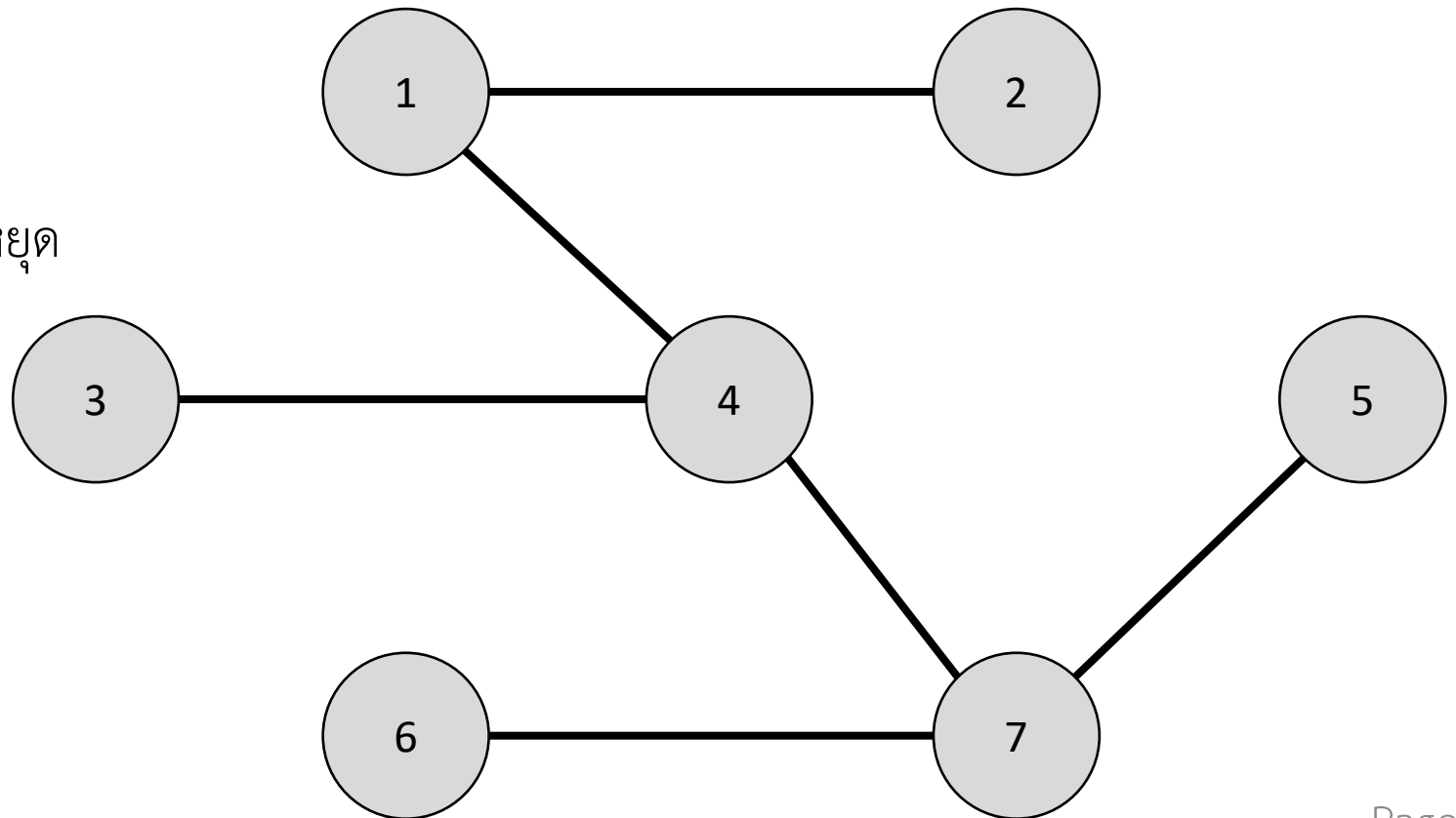
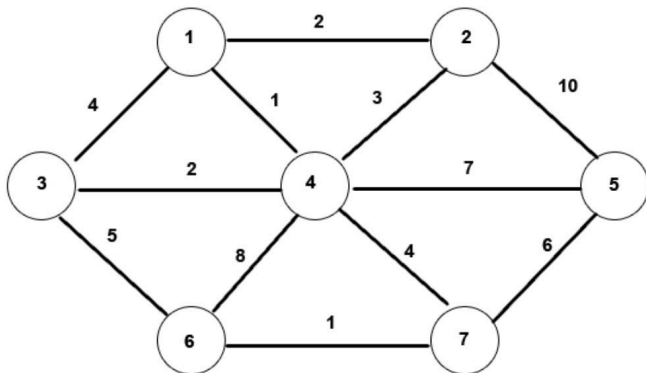


MST: Kruskal's algorithm

1. สร้าง node เท่าไร
2. ขีดเส้นที่ cost น้อยสุด
3. ขีดเส้นที่ cost มากขึ้นตามลำดับ
4. ถ้าทำให้เกิด loop ไม่ต้องขีด
5. เมื่อทุก node เชื่อมกันหมดแล้วก็หยุด

สังเกตว่า มี 7 node จะมี 6 เส้นเชื่อม

MST จะมีจำนวนเส้นเชื่อม = จำนวน node - 1 เสมอ



Outline



- Hierarchical Clustering
 - แนวคิดของ Hierarchical Clustering
 - Agglomerative
 - กระบวนการ
 - การอ่าน Dendrogram
 - Divisive
 - Minimum Spanning Tree
 - กระบวนการ

Divisive Analysis



- มีชื่อเล่นว่า Diana
- ทำงานตรงข้ามกับ Agnes
- เริ่มจากกำหนดคลัสเตอร์จำนวนน้อย (1 คลัสเตอร์ ครอบคลุมข้อมูลทั้งหมด) แล้วค่อย ๆ แบ่งแยกคลัสเตอร์ที่ต่างกันมากที่สุดออกจากกัน
- เมื่อทำซ้ำไปเรื่อย ๆ คลัสเตอร์ก็จะมีขนาดเล็กลง และมีจำนวนของคลัสเตอร์จะค่อย ๆ เพิ่มขึ้น
- ในการแบ่งคลัสเตอร์ออกจากกัน มีวิธีที่หลากหลาย ในที่นี้จะเสนอวิธีการใช้ทฤษฎีกราฟ (Minimum spanning tree) คู่กับ Proximity (Distance)

กระบวนการ Divisive Analysis

สมมติเรามีข้อมูลอยู่ชุดหนึ่ง มี proximity matrix ดังนี้

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

กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)

สร้าง MST ได้ยังไง? Kruskal? ยังไง?

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

กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)

- เทคนิค เมทริกซ์มันดูยาก ทำให้ดูง่ายขึ้นก่อน

A -> B = 1	B -> C = 2	C -> E = 5
A -> C = 2	B -> D = 4	D -> E = 3
A -> D = 2	B -> E = 3	
A -> E = 3	C -> D = 1	

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

กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)

- เทคนิค เมทริกซ์มันดูยาก ทำให้ดูง่ายขึ้นก่อน
- เรียงลำดับเลขน้อยไปมาก

A -> B = 1	B -> C = 2	B -> D = 4
C -> D = 1	A -> E = 3	C -> E = 5
A -> C = 2	D -> E = 3	
A -> D = 2	B -> E = 3	

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

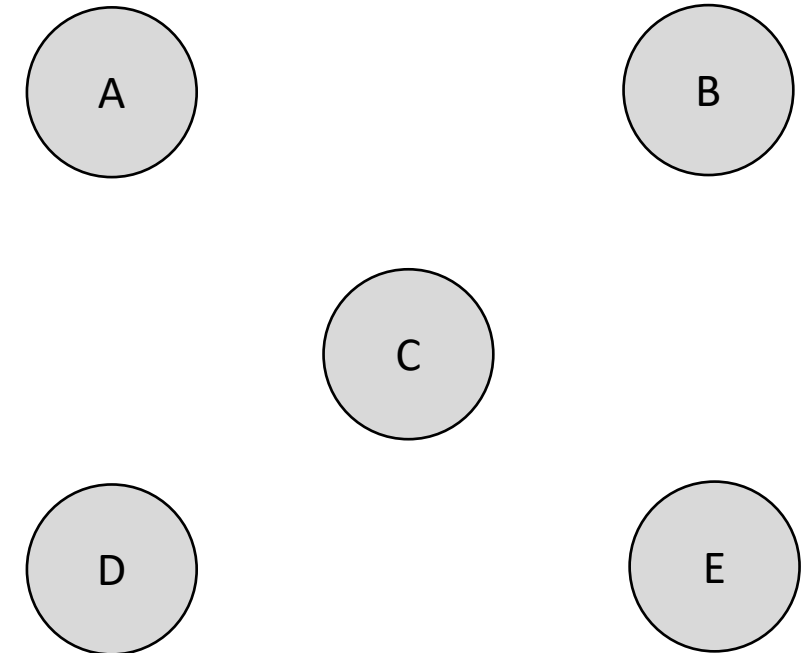
กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)

- **เทคนิค** เมทริกซ์มันดูยาก ทำให้ดูง่ายขึ้นก่อน
- เรียงลำดับเลขน้อยไปมาก
- สร้าง node ให้ครบ

$A \rightarrow B = 1$	$B \rightarrow C = 2$	$B \rightarrow D = 4$
$C \rightarrow D = 1$	$A \rightarrow E = 3$	$C \rightarrow E = 5$
$A \rightarrow C = 2$	$D \rightarrow E = 3$	
$A \rightarrow D = 2$	$B \rightarrow E = 3$	



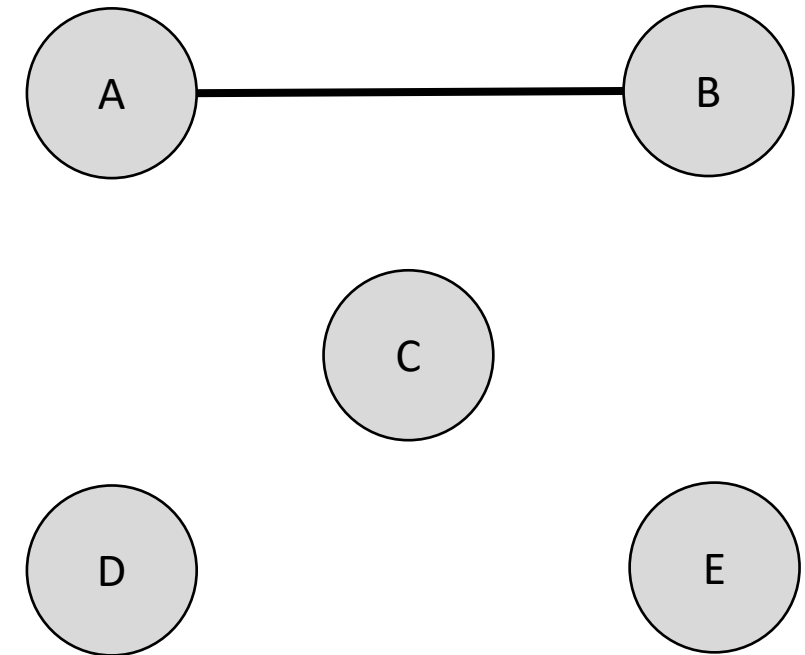
กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)

- **เทคนิค** เมทริกซ์มันดูยาก ทำให้ดูง่ายขึ้นก่อน
- เรียงลำดับเลขน้อยไปมาก
- สร้าง node ให้ครบ
- ขีดเส้นไล่ตามลำดับไป โดยอ้างอิงกฎ Kruskal

A -> B = 1	B -> C = 2	B -> D = 4
C -> D = 1	A -> E = 3	C -> E = 5
A -> C = 2	D -> E = 3	
A -> D = 2	B -> E = 3	



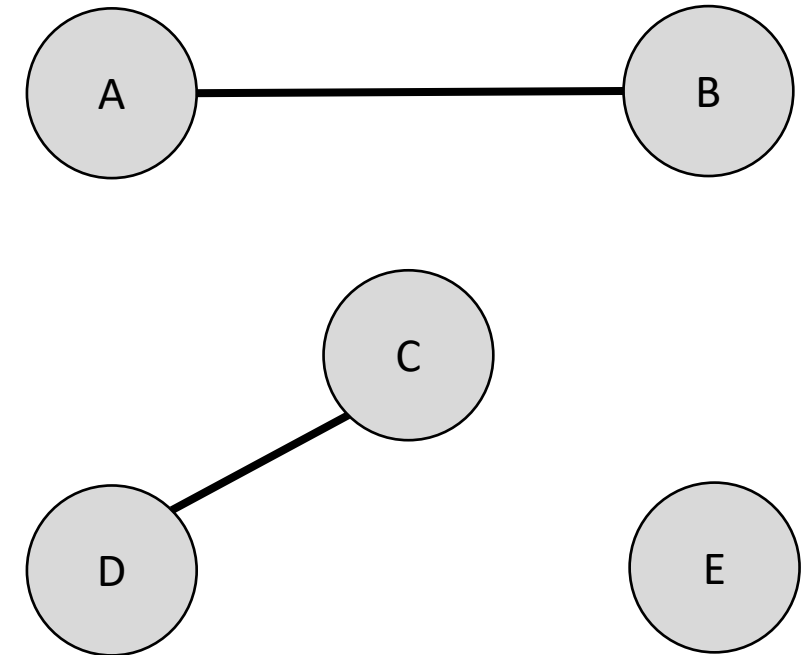
กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)

- **เทคนิค** เมทริกซ์มันดูยาก ทำให้ดูง่ายขึ้นก่อน
- เรียงลำดับเลขน้อยไปมาก
- สร้าง node ให้ครบ
- ขีดเส้นไล่ตามลำดับไป โดยอ้างอิงกฎ Kruskal

$A \rightarrow B = 1$	$B \rightarrow C = 2$	$B \rightarrow D = 4$
$C \rightarrow D = 1$	$A \rightarrow E = 3$	$C \rightarrow E = 5$
$A \rightarrow C = 2$	$D \rightarrow E = 3$	
$A \rightarrow D = 2$	$B \rightarrow E = 3$	



กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)

- **เทคนิค** เมทริกซ์มันดูยาก ทำให้ดูง่ายขึ้นก่อน
- เรียงลำดับเลขน้อยไปมาก
- สร้าง node ให้ครบ
- ขีดเส้นไล่ตามลำดับไป โดยอ้างอิง Kruskal

A -> B = 1

B -> C = 2

B -> D = 4

C -> D = 1

A -> E = 3

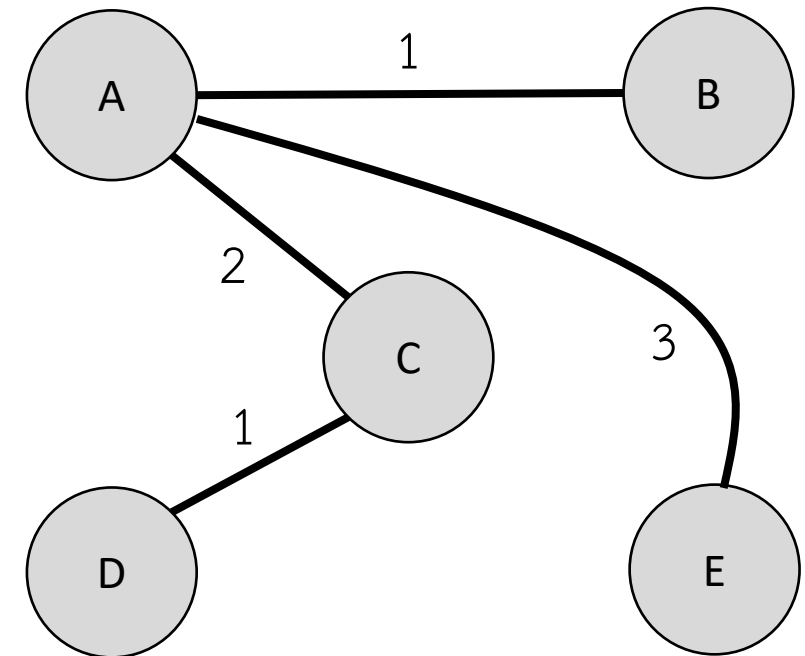
C -> E = 5

A -> C = 2

D -> E = 3

A -> D = 2

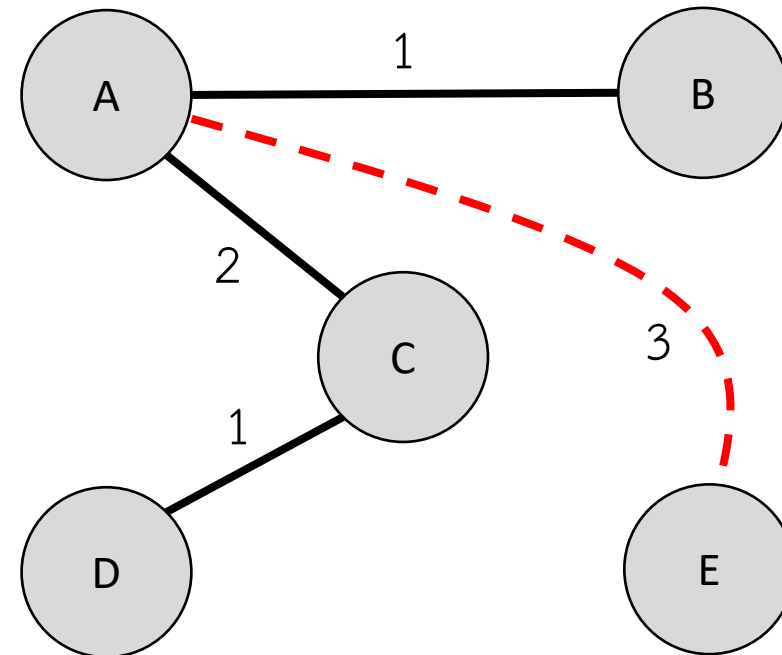
B -> E = 3



กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

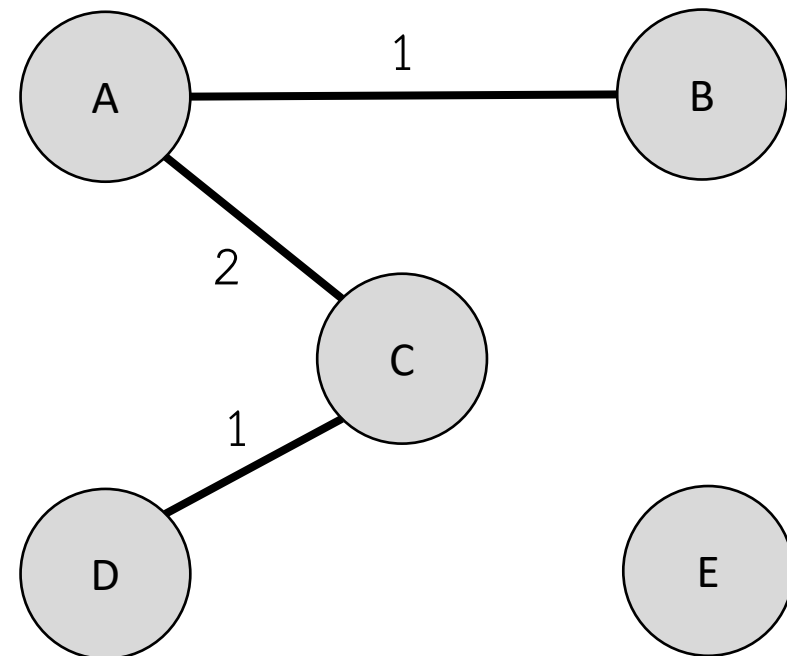
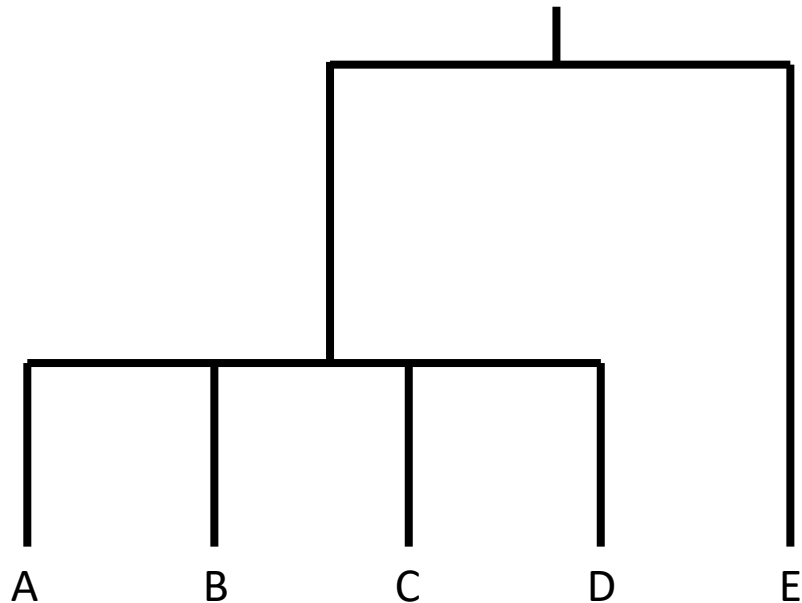
1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)
2. ตัดเส้นเชื่อมที่มี cost สูงสุดออก



กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

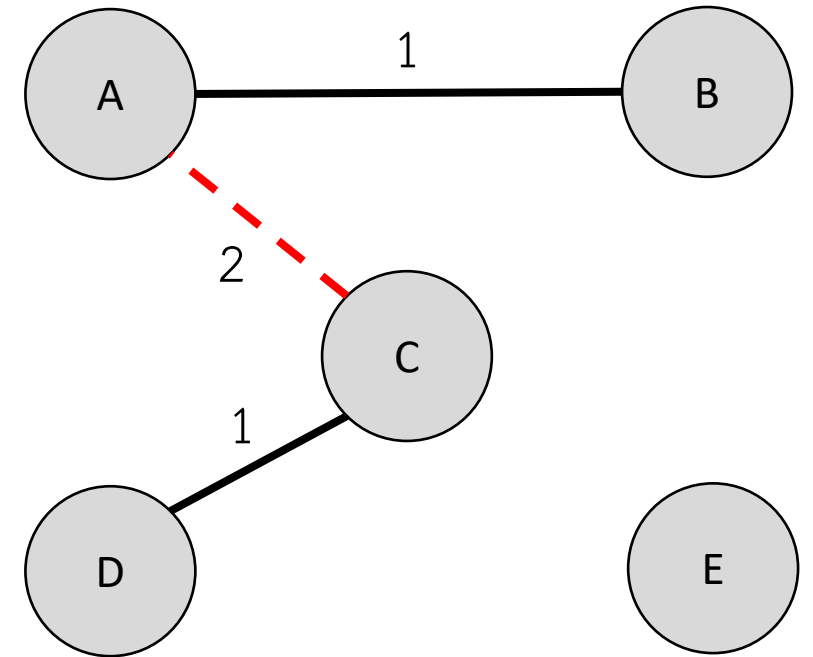
1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)
2. ตัดเส้นเชื่อมที่มี cost สูงสุดออก ตอนนี้ข้อมูลจะถูกแยกเป็น 2 คลัสเตอร์ (วาด Dendrogram)



กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

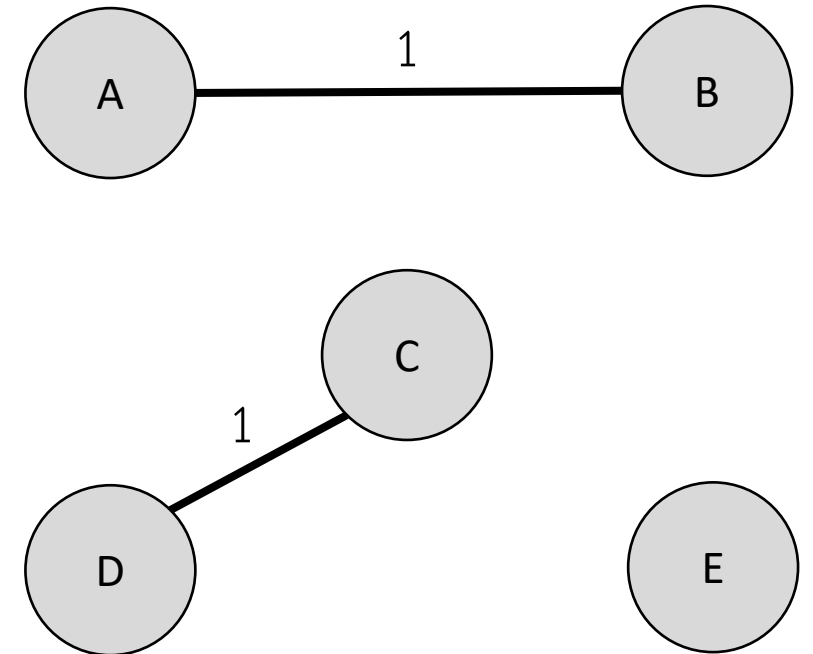
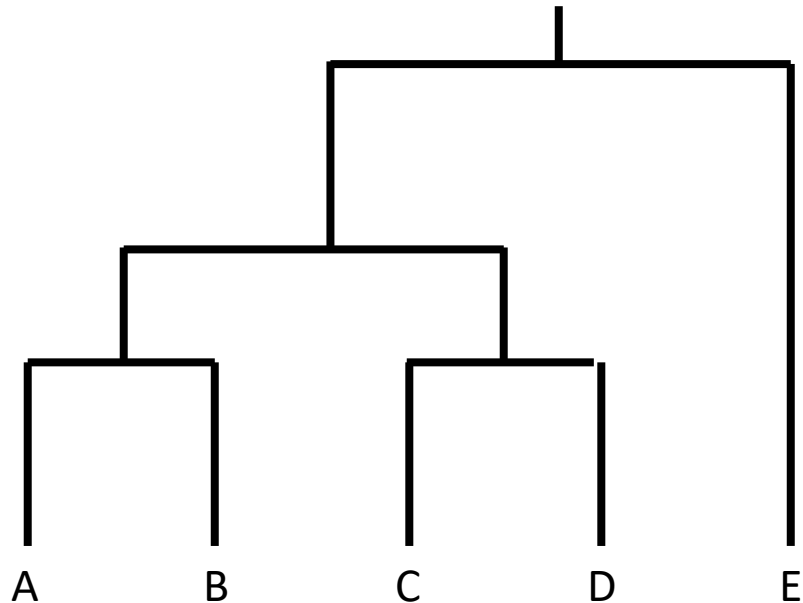
1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)
2. ตัดเส้นเชื่อมที่มี cost สูงสุดออก
3. ทำข้อ 2 ซ้ำ จนกว่าจะได้จำนวนคลัสเตอร์ที่ต้องการ
(หรือแบ่งไม่ได้อีก)



กระบวนการ Divisive Analysis

ขั้นตอนการทำงานของ Diana

1. สร้างกราฟที่มีขนาดเล็กที่สุด ที่เชื่อมต่อระหว่างข้อมูลทุกจุด (Minimum Spanning Tree)
2. ตัดเส้นเชื่อมที่มี cost สูงสุดออก
3. ทำข้อ 2 ซ้ำ จนกว่าจะได้จำนวนคลัสเตอร์ที่ต้องการ (DD)



Conclusion



- Hierarchical Clustering
 - แนวคิดของ Hierarchical Clustering
 - Agglomerative
 - กระบวนการ
 - การอ่าน Dendrogram
 - Divisive
 - Minimum Spanning Tree
 - กระบวนการ