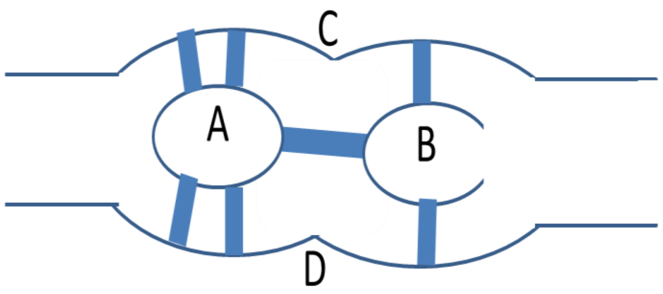
มหาวิทยาลัยราชภัฏนครปฐม

บทที่ 8
กราฟ

ผู้บรรยาย : ผศ.ดร.ณัฐชามญชุ์ ศรีจำเริญรัตนา
สาขาวิชาคอมพิวเตอร์ธุรกิจ คณะวิทยาการจัดการ
มหาวิทยาลัยราชภัฏนครปฐม

กราฟ

- กราฟ (graph) เป็นโครงสร้างข้อมูลที่ใช้เก็บข้อมูล แต่แตกต่างจากโครงสร้างข้อมูลแบบสแตก คิว ลิงค์ลิสต์ โดยทั่วไปกราฟจะเก็บข้อมูลโครงสร้างคล้ายกับต้นไม้อีกทั้งสามารถนำมาประยุกต์ใช้ในงานทางด้านคอมพิวเตอร์ได้อย่างกว้างขวาง
- โดยอาศัยความรู้ทางด้านคณิตศาสตร์เพื่อจำลองเหตุการณ์ในการแก้ปัญหา เช่น การแก้ปัญหาเส้นทางที่สั้นที่สุด การวิเคราะห์ข้อมูลเส้นทางวิกฤติ
- ในปี ค.ศ. 1736 เลออนฮาร์ด ออยเลอร์ (Leonhard Euler) นักคณิตศาสตร์ชาวสวิสเซอร์แลนด์ ได้นำเสนอทฤษฎีที่เรียกว่า “ทฤษฎีออยเลอร์” (ทฤษฎีกราฟ) ขึ้นมาเพื่อแก้ปัญหาสะพานเคอนิกส์เบอร์ก “Konigsberg Bridge Problem” ได้เป็นผลสำเร็จ



ภาพวาดสะพานเคอนิกส์เบอร์ก

กราฟ

- เลออนฮาร์ด ออยเลอร์
  - แทนฝั่งแม่น้ำและเกาะด้วยจุดยอด (vertex หรือ node)
  - แทนสะพานด้วยเส้นเชื่อมระหว่าง จุดยอด 2 จุด เรียกว่า ด้าน (edge หรือ arc)
- กราฟสามารถแสดงความสัมพันธ์ระหว่างข้อมูลหรือจุดที่สนใจ
- ความสัมพันธ์สามารถแสดงในรูปแบบที่ไม่มีความซับซ้อนจนถึงซับซ้อนได้เช่นกัน
- ในความสัมพันธ์ของแต่ละข้อมูลที่สนใจจะพบว่าจะมีความสัมพันธ์ได้มากกว่าหนึ่งจุด

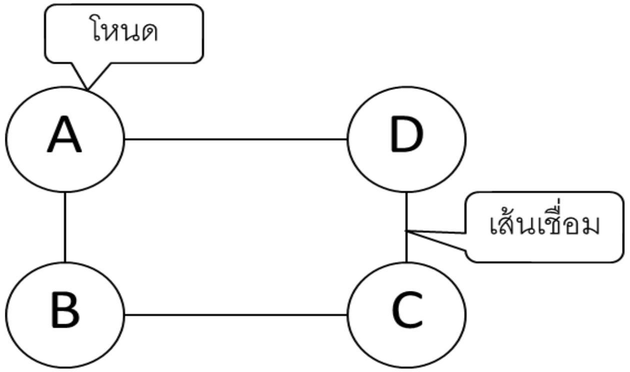
8.1 นิยามกราฟ

8.1 นิยามกราฟ

- กราฟ (graph) เป็นโครงสร้างข้อมูลคล้ายกับต้นไม้
- กำหนดให้ G ใช้แทนสัญลักษณ์ของกราฟ ประกอบด้วย
    - เซตของเวอร์เท็กซ์ V (โหนด)และเอดจ์ E (เส้นเชื่อม) ใช้เชื่อมระหว่างเวอร์เท็กซ์ 2 เวอร์เท็กซ์
- แสดง ความสัมพันธ์ของกราฟ G = (V,E)

โครงสร้างข้อมูลต้นไม้

- โหนด (V) = {A, B, C, D}
- เส้นเชื่อม (E) ={(A,B), (A,D), (B,C), (C,D)}

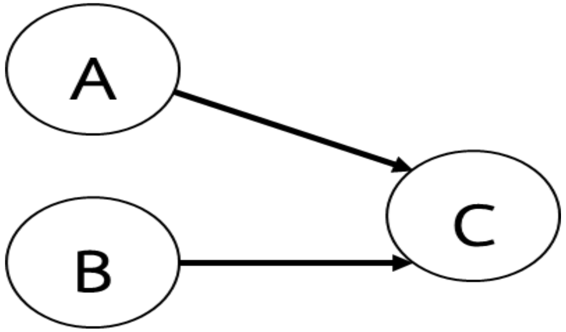


8.2 โครงสร้างกราฟ

- โดยปกติ กราฟ หมายถึงกราฟไม่ระบุทิศทางอย่างง่ายและจำกัด
- แต่เนื่องจากกราฟจึงแบ่งออกได้เป็น 2 ประเภท
  - กราฟแบบมีทิศทาง (Directed Graph)
  - กราฟแบบไม่มีทิศทาง (Undirected Graph)

8.2 โครงสร้างกราฟ(ต่อ)

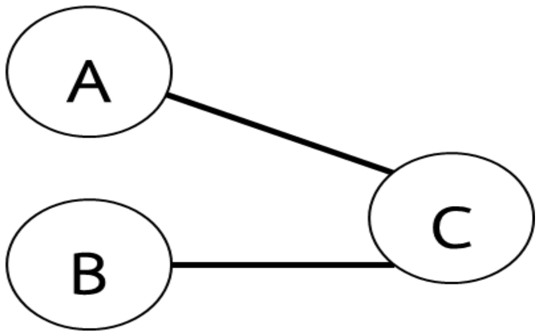
8.2.1 กราฟแบบมีทิศทาง



- กราฟที่เส้นเชื่อมระหว่างโหนดจะมีหัวลูกศรแสดงทิศทางของการบอกทาง จากกราฟประกอบด้วย
  - โหนด 3 โหนด คือ {A,B,C}
  - เซตของเอจ คือ {(A,C),(B,C)}
- กราฟแบบมีทิศทางจะเชื่อมระหว่างโหนดในทิศทางเดียวเท่านั้น เช่น เอจ (A,C)
- แต่ไม่สามารถเชื่อมเอจ (C,A) เนื่องจากทิศทางของหัวลูกศรทำการชี้จากโหนด A มายังโหนด C เท่านั้น

8.2 โครงสร้างกราฟ(ต่อ)

8.2.2 กราฟแบบไม่มีทิศทาง



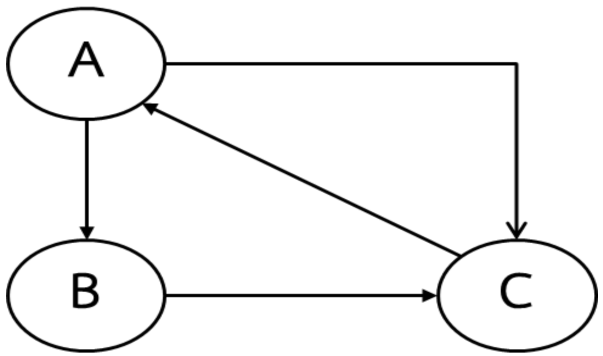
- กราฟที่เส้นเชื่อมระหว่างโหนดไม่สามารถบอกทิศทางของการเชื่อมต่อได้ เนื่องจากไม่มีหัวลูกศร
- กราฟแบบไม่มีทิศทางจะมีเส้นเชื่อมระหว่างโหนด

  เช่น

  - โหนด A จะมีเส้นทางไปโหนด C
  - โหนด C จะมีเส้นทางไปยังโหนด A และ โหนด B
  - โหนด B ยังมีเส้นทางโหนด C ในเส้นทางเดียวกัน

8.2 โครงสร้างกราฟ(ต่อ)

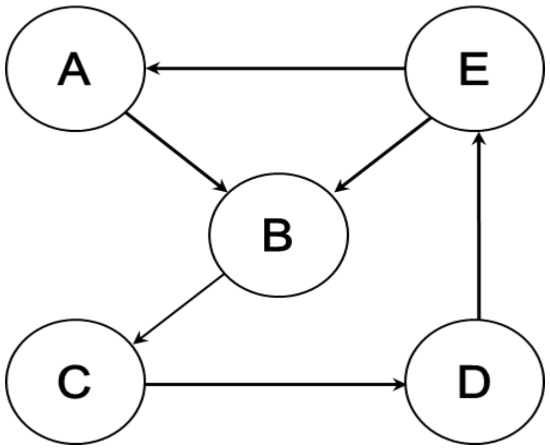
8.2.3 ไซเคิล



- ไซเคิลเป็นกราฟที่ที่จุดเริ่มต้นของเวอร์เท็กซ์มีเส้นทางย้อนกลับมาที่เป็นจุดเวอร์เท็กซ์นั้น ๆ
- จุดเริ่มต้นและจุดสิ้นสุดเป็นจุดเดียวกัน เช่น
  - A-B-C-A หรือ A-C-A

8.2 โครงสร้างกราฟ(ต่อ)

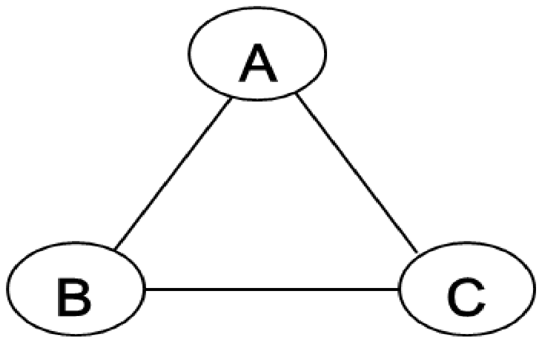
8.2.4 เส้นทาง

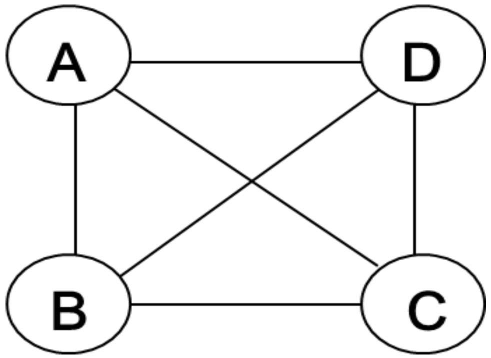


- เส้นทาง คือ เส้นทางที่บอกให้ทราบการเชื่อมอย่างเป็นลำดับโหนดใดเป็นจุดเริ่มต้นจะเป็นโหนดใดและจุดสิ้นสุดของโหนดปลายทางอยู่ที่โหนดใด เช่น path(A,B,C,D,E)
- เส้นทางจำนวนของเวอร์เท็กซ์สามารถเขียนด้วย $|V|$
- จำนวนของเอจสามารถเขียนด้วย $|E|$

8.2 โครงสร้างกราฟ(ต่อ)

8.2.5 กราฟสมบูรณ์

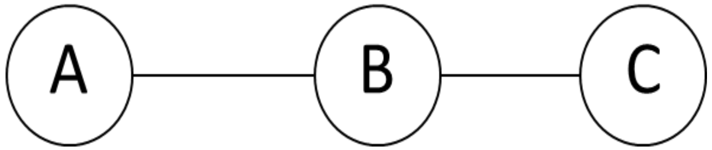




- กราฟที่ทุกโหนดในกราฟมีเส้นเชื่อมหากันครบทุกโหนด

8.2 โครงสร้างกราฟ(ต่อ)

8.2.6 โหนดประชิด



- โหนดประชิด (adjacent vertex)

  คือ โหนด 2 โหนดที่มีเส้นเชื่อมระหว่างสองโหนด

- โหนด A เป็น adjacent vertex กับโหนด B

  เมื่อมีเส้นเชื่อมระหว่างโหนดร่วมกัน

  คือ(A,B) และ (B,A)

8.2 โครงสร้างกราฟ(ต่อ)

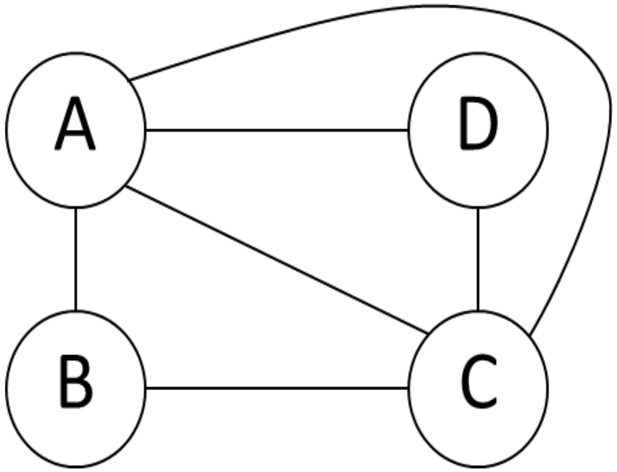
8.2.7 ดีกรี

ดีกรีของเวอร์เท็กซ์ คือ จำนวนเส้นเชื่อมหรือเอดจ์ที่เชื่อมอยู่กับเวอร์เท็กซ์นั้น ๆ แบ่งเป็น 2 แบบ

1) ดีกรีของกราฟแบบไม่มีทิศทาง
2) ดีกรีของกราฟแบบมีทิศทาง

8.2 โครงสร้างกราฟ(ต่อ)

1) ดีกรีของกราฟแบบไม่มีทิศทาง เป็นดีกรีของเส้นเชื่อมจากเวอร์เท็กซ์ปัจจุบันที่ทำการเชื่อมไปยังเวอร์เท็กซ์ต่าง



vertex	ดีกรี
A	4
B	2
C	4
D	2

8.2 โครงสร้างกราฟ(ต่อ)

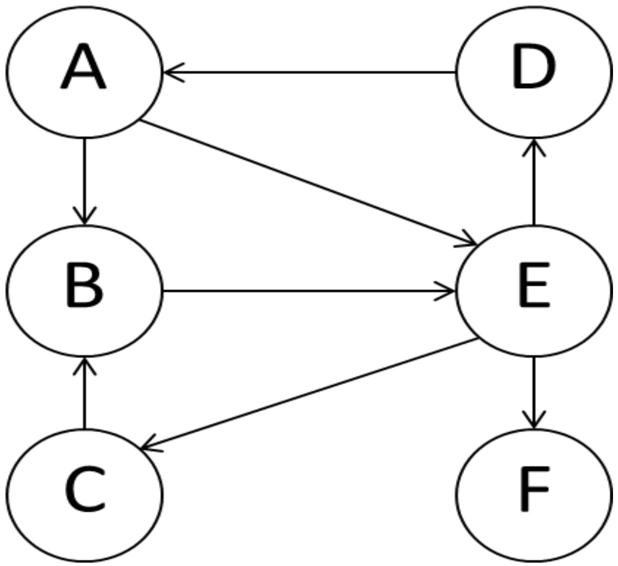
2) ดีกรีของกราฟแบบมีทิศทาง กราฟแบบมีทิศทางจะมีลูกศรแสดงเส้นที่เชื่อมโยงระหว่างโหนด แบ่งเป็น 2 แบบ

1) in-degree ของเวอร์เท็กซ์ใด ๆ คือ จำนวนเอดจ์ ที่เส้นเชื่อมโยงเชื่อมเข้ามายังโหนดใด ๆ

2) out-degree ของเวอร์เท็กซ์ใด ๆ คือ จำนวนเอดจ์ ที่เส้นเชื่อมโยงเชื่อมออกไปยังโหนดใด ๆ

8.2 โครงสร้างกราฟ(ต่อ)

ดีกรีของกราฟแบบมีทิศทาง



vertex	in-degree	out-degree	รวมดีกรี
A	1	2	3
B	2	1	3
C	1	1	2
D	1	1	2
E	2	3	5
F	1	0	1

8.3 การนำเสนอโครงสร้างข้อมูลชนิดกราฟ

- การนำเสนอโครงสร้างข้อมูลชนิดกราฟเพื่อใช้งานจำเป็นต้องมีการเปลี่ยนแปลงโครงสร้างเพื่อจัดเก็บในหน่วยความจำหรือเพื่อแสดงผลในลักษณะที่คอมพิวเตอร์สามารถเข้าใจได้ง่าย
- การแปลงโครงสร้างกราฟเพื่อนำไปประมวลผลด้วยระบบคอมพิวเตอร์ แบ่งการนำเสนอ 2 วิธีคือ
  - เมตริกซ์ประชิด (adjacency matrix)
  - ลิสต์ประชิด (adjacency matrix)

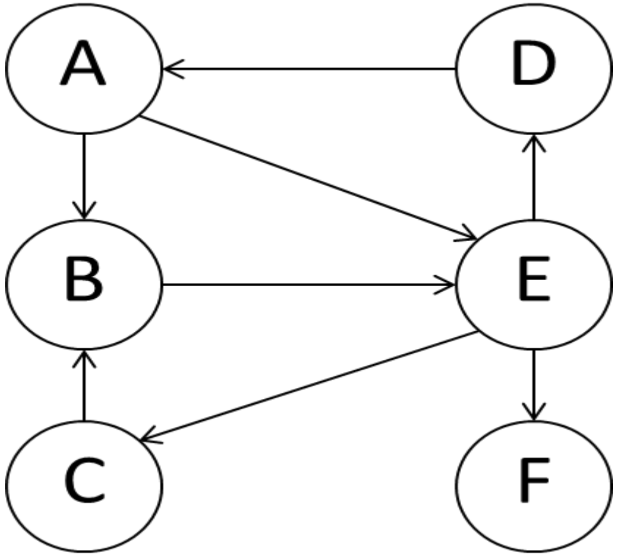
8.3 การนำเสนอโครงสร้างข้อมูลชนิดกราฟ(ต่อ)

8.3.1 เมตริกซ์ประชิด

- เมตริกซ์ประชิด หรือ adjacency matrix เป็นการแปลงโครงสร้างกราฟโดยใช้ข้อมูลต่าง ๆ ของกราฟ และนำเสนอด้วยโครงสร้างข้อมูลเมตริกซ์
- การนำเสนอสามารถใช้องค์ประกอบของกราฟ โดย
  - โหนดและเส้นเชื่อมเพื่อแทนข้อมูลของกราฟลงในเมตริกซ์ขนาด N X N เมตริกซ์
  - ความสัมพันธ์ระหว่างโหนดกำหนดค่าด้วยตัวเลข โดย
    - เมตริกซ์คู่ใดมีค่าเป็น 1 หมายถึง โหนดคู่นั้น**มี**เส้นเชื่อมโยงระหว่างกัน
    - เมตริกซ์คู่ใดมีค่าเป็น 0 หมายถึง โหนดคู่นั้น**ไม่มี**เส้นเชื่อมระหว่างกัน

8.3 การนำเสนอโครงสร้างข้อมูลชนิดกราฟ(ต่อ)

ตัวอย่างที่ 8.1 จงแสดงเมตริกซ์ประชิด (adjacency matrix) จากกราฟแบบมีทิศทาง ต่อไปนี้



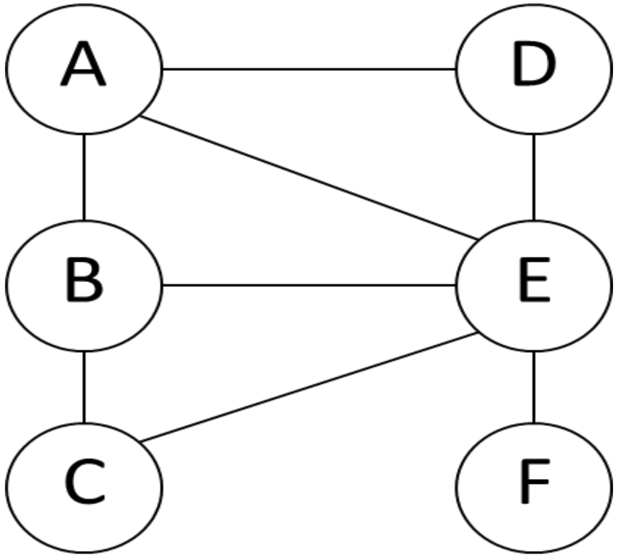
A
B
C
D
E
F

vertex vector

	A	B	C	D	E	F
A	0	1	0	0	1	0
B	0	0	0	0	1	0
C	0	1	0	0	0	0
D	1	0	0	0	0	0
E	0	0	1	1	0	1
F	0	0	0	0	0	0

8.3 การนำเสนอโครงสร้างข้อมูลชนิดกราฟ(ต่อ)

ตัวอย่างที่ 8.2 จงแสดงเมตริกซ์ประชิด (adjacency matrix) จากกราฟแบบไม่มีทิศทาง ต่อไปนี้



A
B
C
D
E
F

vertex vector

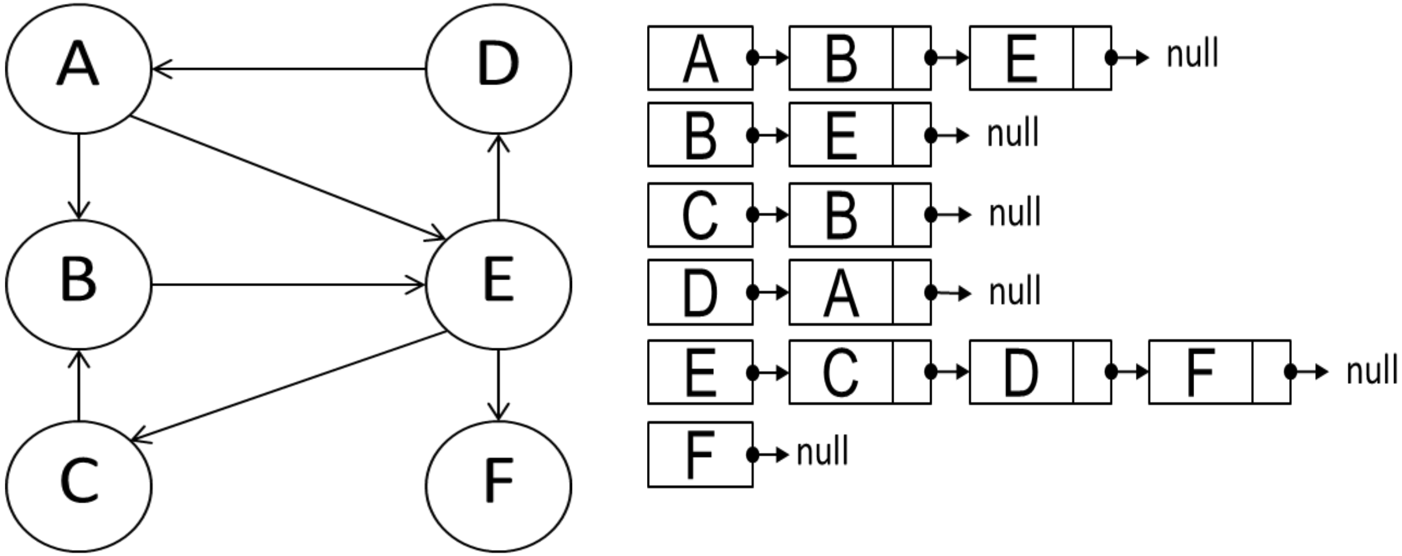
	A	B	C	D	E	F
A	0	1	0	1	1	0
B	1	0	1	0	1	0
C	0	1	0	0	1	0
D	1	0	0	0	1	0
E	1	1	1	1	0	1
F	0	0	0	0	1	0

8.3.2 ลิสต์ประชิด

- ลิสต์แบบประชิดสามารถใช้โครงสร้างข้อมูลอาร์เรย์แบบ link lists ในการเก็บข้อมูลการเชื่อมโยงระหว่างโหนดที่นำเข้าสู่ระบบประมวลผลด้วยคอมพิวเตอร์ ประกอบด้วยข้อมูล 2 ชนิด คือ
  - list of vertices เพื่อแสดงข้อมูลของเวอร์เท็กซ์
  - list of adjacent vertices เพื่อแสดงความสัมพันธ์ระหว่างแต่ละเวอร์เท็กซ์

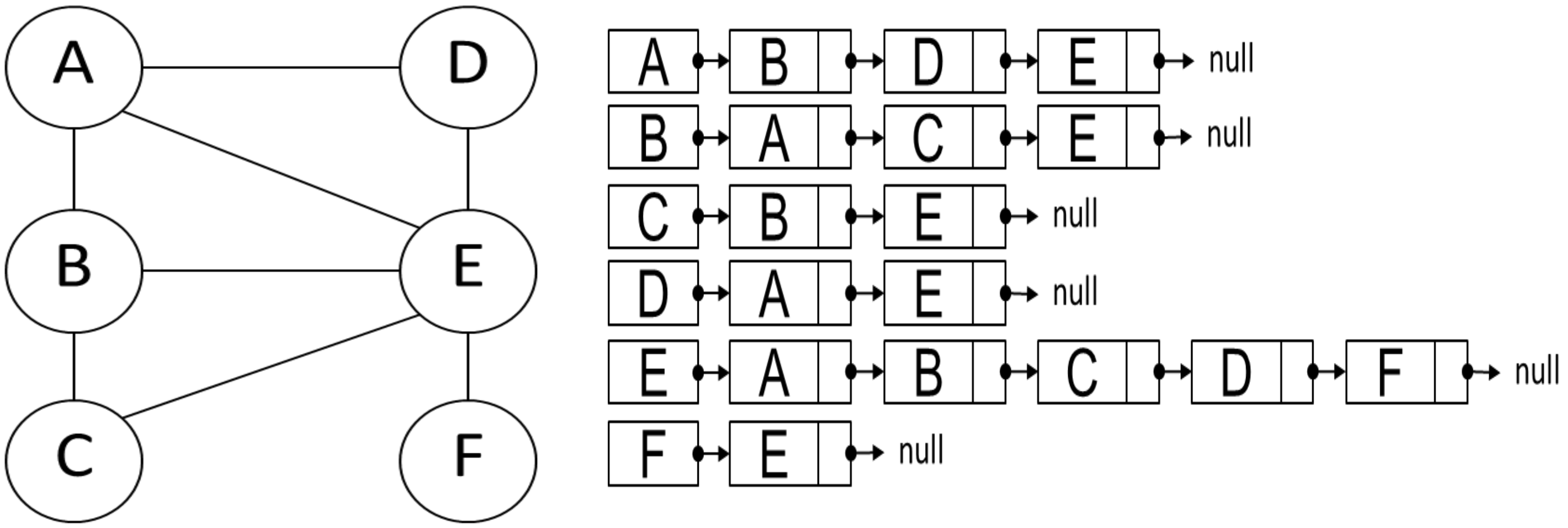
8.3 การนำเสนอโครงสร้างข้อมูลชนิดกราฟ(ต่อ)

ตัวอย่างที่ 8.3 จงแสดงลิสต์แบบประชิด (adjacency list) จากกราฟแบบมีทิศทาง ต่อไปนี้



8.3 การนำเสนอโครงสร้างข้อมูลชนิดกราฟ(ต่อ)

ตัวอย่างที่ 8.4 จงแสดงลิสต์แบบประชิด (adjacency list) จากกราฟแบบไม่มีทิศทาง ต่อไปนี้



8.4 การท่องเข้าไปในกราฟ

การท่องเข้าไปในกราฟ หรือ traverse graph

- ความคล้ายกับการท่องไปใน binary tree
- แต่ละโหนดมีการประมวลผล 1 ครั้งและทำเครื่องหมาย แสดงให้รู้ว่ามีการเยือนโหนดนั้น ๆ แล้ว
- วิธีมาตรฐานในการท่องเข้าไปในกราฟ คือ
    - การท่องในแนวลึก (depth first traversal)
    - การท่องในแนวกว้าง (breadth first traversal)

8.4 การท่องเข้าไปในกราฟ(ต่อ)

8.4.1 การท่องในแนวลึก

การท่องในแนวลึก (depth first traversal) มีวิธีการดังนี้

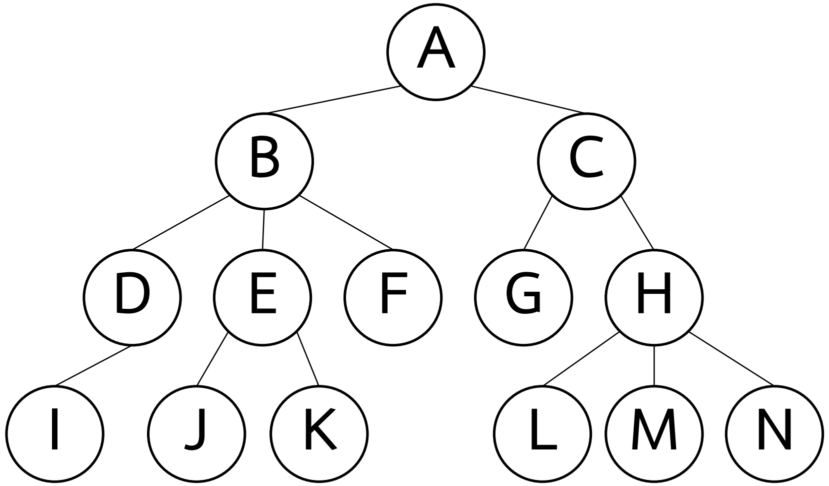
- เริ่มต้นจากโหนดใดโหนดหนึ่งในกราฟและกำหนดให้โหนดนี้เป็นโหนดเริ่มต้น

  (ปกติจะกำหนดให้โหนดเริ่มต้นเริ่มที่ root ก่อน)

- ตรวจสอบโหนดปัจจุบันที่เยือนนั้นมีโหนด adjacenct หรือไม่
  - ให้กำหนดการเยือนโหนดจากซ้ายก่อนจนครบทุกโหนดด้านซ้าย
  - ให้มาทำการเยือนโหนดด้านขวา

8.4 การท่องเข้าไปในกราฟ(ต่อ)

ตัวอย่างที่ 8.5 วิธีการท่องในแนวลึก โดยกำหนดให้ A เป็นโหนดเริ่มต้น



- A B D I E J K F C G H L M N

8.4 การท่องเข้าไปในกราฟ(ต่อ)

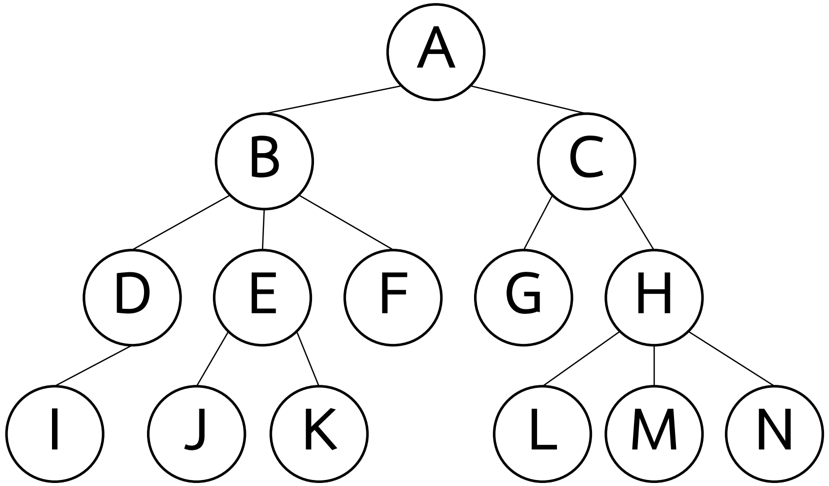
8.4.2 การท่องในแนวกว้าง

การท่องในแนวกว้าง (breadth-first traversal) มีวิธีการดังนี้

- เริ่มจากโหนดใดที่เป็นโหนดเริ่มต้นหรือ root ของกราฟ
    - ทำการเยือนโหนด adjacent ของโหนดเริ่มต้นนั้นทุกโหนดก่อน
    - ทำการเยือนโหนด adjacent จากโหนดภายในกราฟจนหมด

8.4 การท่องเข้าไปในกราฟ(ต่อ)

ตัวอย่างที่ 8.6 วิธีการท่องในแนวกว้าง



- A B C D E F G H I J K L M N

8.5 การค้นหาเส้นทางที่สั้นที่สุด

- การประยุกต์ใช้งานกราฟเพื่อหาระยะทางระหว่างจุดเริ่มต้นและจุดสิ้นสุดที่มีระยะทางรวมน้อยที่สุด ในการใช้งานเพื่อค้นหาเส้นทางที่สั้นที่สุดเพื่อนำไปใช้งานได้ในหลาย ๆ รูปแบบ เช่น การหาเส้นทางการบิน การจัดจำหน่ายสินค้าของพนักงานขายในบริษัท เป็นต้น
- การใช้กราฟเพื่อหาเส้นเชื่อมระหว่างเวอร์เท็ก 2 เวอร์เท็กซ์ จะต้องกำหนดให้มีค่าน้ำหนักกำกับอยู่ทีละเส้นของกราฟให้มีความสัมพันธ์ระหว่างเวอร์เท็กซ์
- การค้นหาเส้นทางที่สั้นที่สุดนั้น สามารถทำการค้นหาได้ด้วยวิธีต่อไปนี้
  - ขั้นตอนวิธีของครูสกาล
  - ขั้นตอนวิธีของพริม
  - ขั้นตอนวิธีของไดก์สตรา

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

8.5.1 ขั้นตอนวิธีของครูสกาล

- ครูสกาล (Kruskal's algorithm) เป็นอัลกอริทึมที่ โจเซฟ ครูสกาล นำเสนอในนิตยสารทางวิทยาศาสตร์ชื่อ Proceedings of the American Mathematical Society ตอนกำลังศึกษาในระดับปริญญาตรี
- การสร้างต้นไม้ทอดข้ามน้อยที่สุด (mininum spanning trees) โดยพิจารณาค่าน้ำหนักของแต่ละเอดจ์ที่จะนำมาสร้างเป็นกราฟ ซึ่งกราฟที่สร้างจะมีคุณสมบัติเป็นอัลกอริทึมแบบละโมบ
- เนื่องจากแต่ละขั้นตอนที่มีการเลือกเอดจ์ เพื่อสร้าง spanning-tree จะเลือกจากน้ำหนักที่มีค่าน้อยที่สุดและเส้นเชื่อม (edge) ที่สร้างขึ้นต้องไม่เกิด cycle ดังนั้น การสร้างต้นไม้ทอดข้ามน้อยที่สุดจะทำการ disjoint-path เพื่อทำการเรียงลำดับค่าน้ำหนักของแต่ละเส้นจากน้ำหนักน้อยไปมากก่อน ทำให้ผลลัพธ์การสร้างป่ามีค่าน้ำหนักรวมของเส้นเชื่อมมีค่ารวมน้อยที่สุดด้วยเช่นกัน

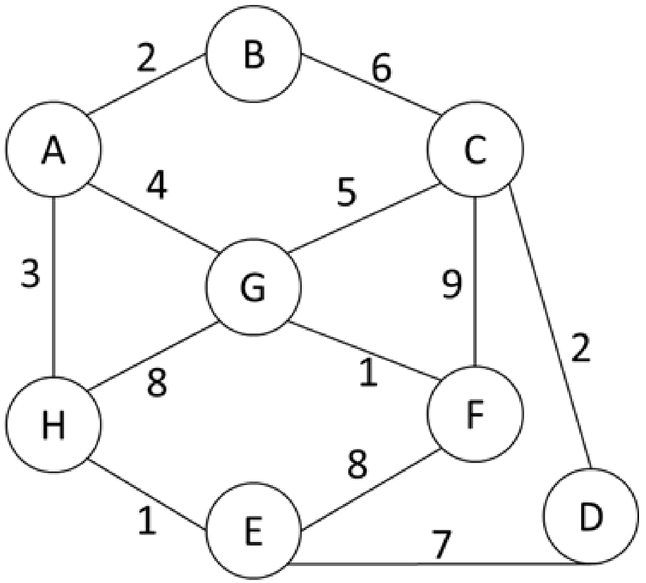
8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ขั้นตอนการทำงานของครูสกาล

1) นำเส้นเชื่อมทั้งหมดในกราฟมาทำการเรียงลำดับน้ำหนักจากน้อยไปหามาก
2) เลือกเส้นเชื่อมจากข้อ 1 มา 1 เส้น จากที่มีค่าน้ำหนักน้อยที่สุด
3) ทำการเลือกเส้นเชื่อมซ้ำข้อ 1 ตามเงื่อนไขต่อไปนี้
    3.1) เลือกเส้นเชื่อมที่มีค่าน้ำหนักน้อยที่สุดและยังไม่ถูกเลือกนำไปให้
    3.2) เส้นเชื่อมที่เลือกจะต้องไม่ทำให้เกิด cycle ของเวอร์เท็กซ์ที่เชื่อมไป
4) จบการทำงาน เมื่อทำซ้ำจนครบทุกเวอร์เท็กซ์

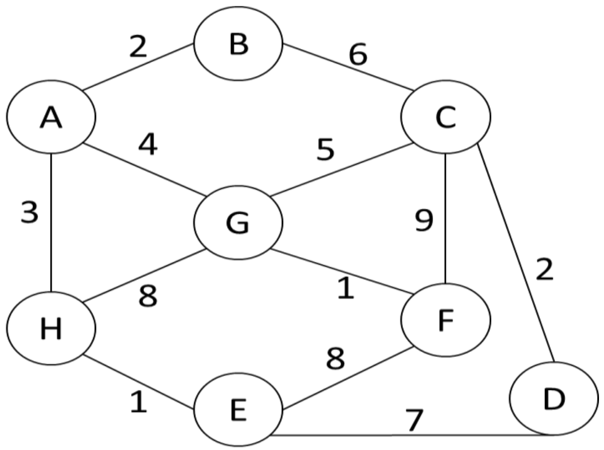
ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด

จากกราฟต่อไปนี้



8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด (ต่อ)



วิธีทำ

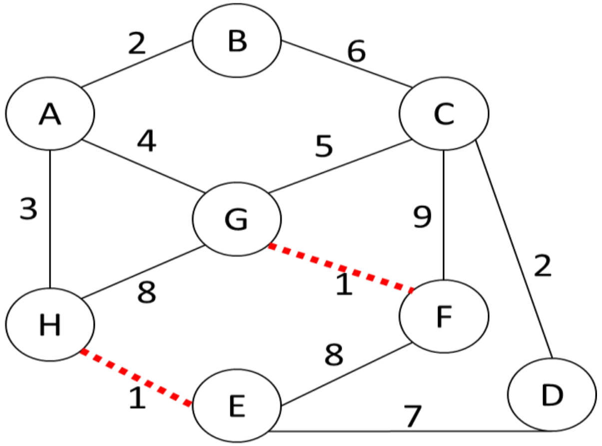
ทำการเรียงลำดับน้ำหนักของเส้นเชื่อมแต่ละเส้น

จากน้ำหนักน้อยสุดไปหาน้ำหนักมากสุด

เส้นเชื่อม	น้ำหนักของเส้นเชื่อม	เส้นเชื่อม	น้ำหนักของเส้นเชื่อม
E-H	1	C-G	5
G-F	1	B-C	6
A-B	2	D-E	7
C-D	2	E-F	8
A-H	3	G-H	8
A-G	4	C-F	9

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด (ต่อ)



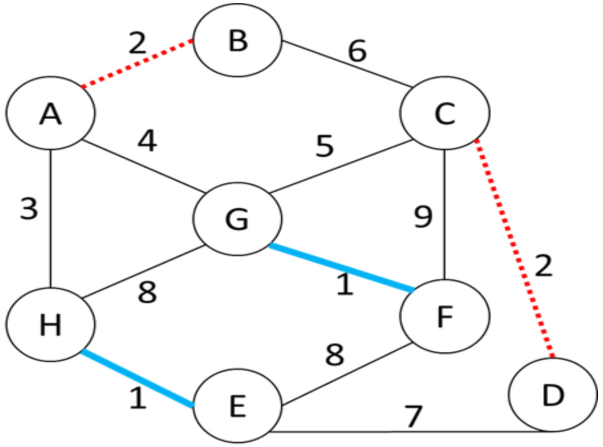
เส้นเชื่อม EH และ GF มีลักษณะเป็นเส้นประเพื่อทำการแสดงให้เห็นการเป็นเส้นที่มีน้ำหนักน้อยและถูกเลือก

ทำการเลือกเส้นเชื่อม รอบที่ 1

EH และ GF มีน้ำหนักของเส้นเชื่อมเท่ากับ 1 เลือก EH และ GF

เส้นเชื่อม	น้ำหนักของเส้นเชื่อม	เส้นเชื่อม	น้ำหนักของเส้นเชื่อม
E-H	1	C-G	5
G-F	1	B-C	6
A-B	2	D-E	7
C-D	2	E-F	8
A-H	3	G-H	8
A-G	4	C-F	9

ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด (ต่อ)



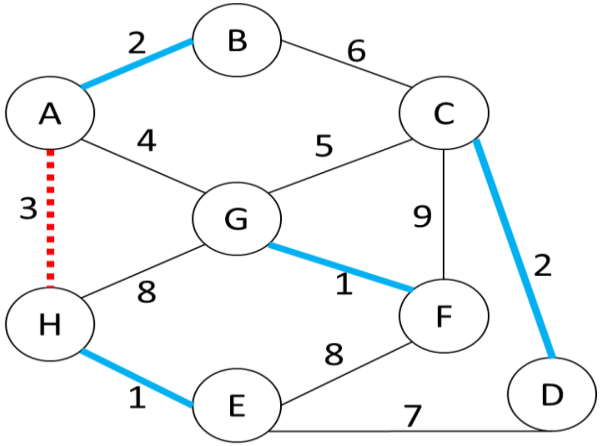
เส้นเชื่อม AB และ CD มีลักษณะเป็นเส้นประเพื่อทำการแสดงให้เห็นการเป็นเส้นที่มีน้ำหนักน้อยและถูกเลือก

ทำการเลือกเส้นเชื่อม รอบที่ 2

AB และ CD มีน้ำหนักของเส้นเชื่อมเท่ากับ 2 เลือก AB และ CD

เส้นเชื่อม	น้ำหนักของเส้นเชื่อม	เส้นเชื่อม	น้ำหนักของเส้นเชื่อม
E-H	1	C-G	5
G-F	1	B-C	6
A-B	2	D-E	7
C-D	2	E-F	8
A-H	3	G-H	8
A-G	4	C-F	9

ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด (ต่อ)



เส้นเชื่อม AH มีลักษณะเป็นเส้นประเพื่อทำการแสดงให้เห็นการเป็นเส้นที่มีน้ำหนักน้อยและถูกเลือก

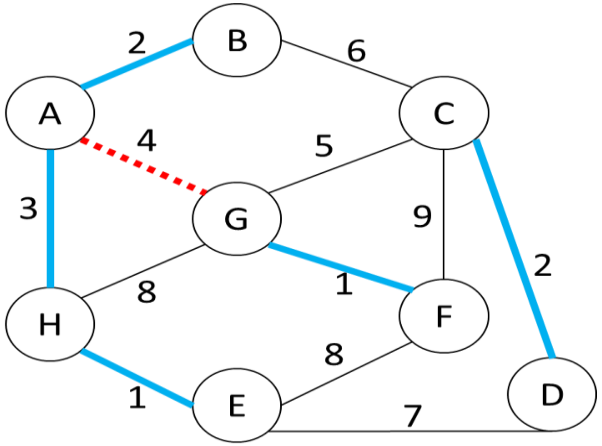
ทำการเลือกเส้นเชื่อม รอบที่ 3

AH มีน้ำหนักของเส้นเชื่อมเท่ากับ 3 เลือก AH

เส้นเชื่อม	น้ำหนักของเส้นเชื่อม	เส้นเชื่อม	น้ำหนักของเส้นเชื่อม
E-H	1	C-G	5
G-F	1	B-C	6
A-B	2	D-E	7
C-D	2	E-F	8
A-H	3	G-H	8
A-G	4	C-F	9

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด (ต่อ)



เส้นเชื่อม AG มีลักษณะเป็นเส้นประเพื่อทำการแสดงให้เห็นการเป็นเส้นที่มีน้ำหนักน้อยและถูกเลือก

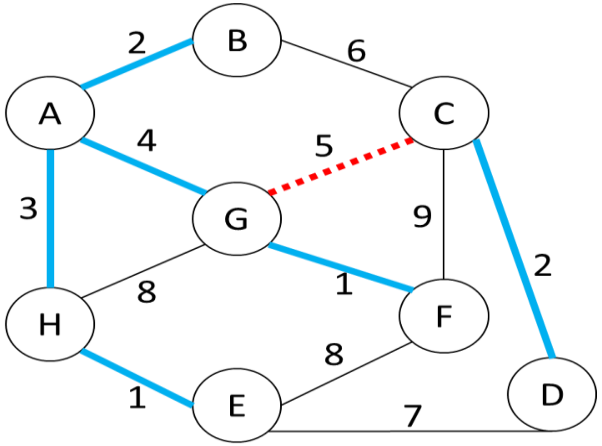
ทำการเลือกเส้นเชื่อม รอบที่ 4

AG มีน้ำหนักของเส้นเชื่อมเท่ากับ 4 เลือก AG

เส้นเชื่อม	น้ำหนักของเส้นเชื่อม	เส้นเชื่อม	น้ำหนักของเส้นเชื่อม
E-H	1	C-G	5
G-F	1	B-C	6
A-B	2	D-E	7
C-D	2	E-F	8
A-H	3	G-H	8
A-G	4	C-F	9

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด (ต่อ)



เส้นเชื่อม CG มีลักษณะเป็นเส้นประเพื่อทำการแสดงให้เห็นการเป็นเส้นที่มีน้ำหนักน้อยและถูกเลือก

ทำการเลือกเส้นเชื่อม รอบที่ 5

CG มีน้ำหนักของเส้นเชื่อมเท่ากับ 5 เลือก CG

เส้นเชื่อม	น้ำหนักของเส้นเชื่อม	เส้นเชื่อม	น้ำหนักของเส้นเชื่อม
E-H	1	C-G	5
G-F	1	B-C	6
A-B	2	D-E	7
C-D	2	E-F	8
A-H	3	G-H	8
A-G	4	C-F	9

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด (ต่อ)

ทำการเลือกเส้นเชื่อม รอบที่ 6

เมื่อพิจารณาเส้นเชื่อมที่เหลือพบว่าน้ำหนักเส้นเชื่อมลำดับถัดไป ซึ่งไม่สามารถทำการเลือกเป็นเส้นเชื่อมต่อได้ คือ

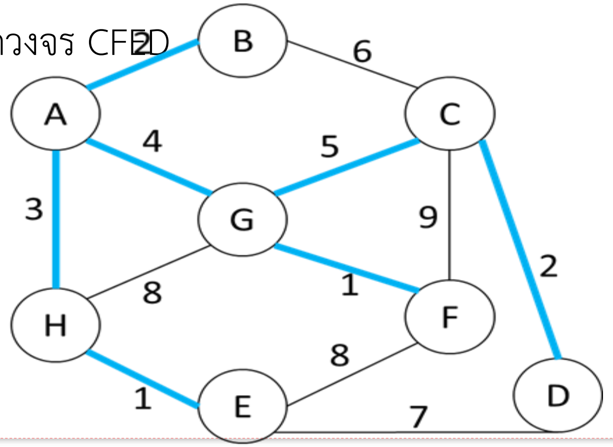
BC เนื่องจาก BC จะทำให้เกิดวงจร BCGA

DE เนื่องจาก DE จะทำให้เกิดวงจร DEFC

EF เนื่องจาก EF จะทำให้เกิดวงจร EFCD

GH เนื่องจาก GH จะทำให้เกิดวงจร GHEF

CF เนื่องจาก CF จะทำให้เกิดวงจร CFED

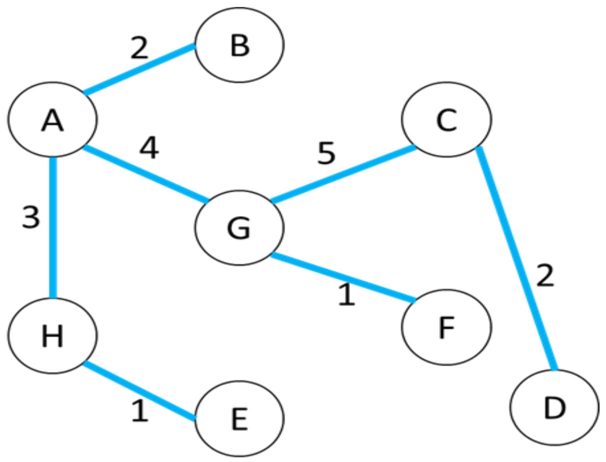


เส้นเชื่อม	น้ำหนักของเส้นเชื่อม	เส้นเชื่อม	น้ำหนักของเส้นเชื่อม
E-H	1	C-G	5
G-F	1	B-C	6
A-B	2	D-E	7
C-D	2	E-F	8
A-H	3	G-H	8
A-G	4	C-F	9

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.7 จงใช้ขั้นตอนวิธีของครูสกาลในการสร้าง tree ที่มีระยะทางที่สั้นที่สุด (ต่อ)

ผลการเลือกเส้นเชื่อม จึงทำให้เส้นทางการเชื่อมโยงโหนดภายในกราฟขั้นตอนวิธีของครูสกาลได้ดังนี้



ดังนั้นให้ตรวจสอบเส้นเชื่อมอีกครั้งทุกเวอร์เท็กซ์มีการเชื่อมต่อกันทุกจุดและคำนวณน้ำหนักรวมเป็น 1+1+2+2+3+4+5=18

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

8.5.2 ขั้นตอนวิธีของพริม

ขั้นตอนวิธีของพริม (prim's algorithm) ใช้วิธีการเชิงละโมบเพื่อสร้างเป็น minimum-spanning-tree อีกวิธีหนึ่งใช้หา shortest path

- ความคล้ายของ prim's algorithm กับ kruskal's algorithm คือ การสร้าง tree ย่อย ๆ และนำมารวมเป็นต้นไม้แผ่ที่มีขนาดเล็กที่สุด
- ความแตกต่าง คือ prim's algorithm จะพิจารณาเวอร์เท็กซ์ปัจจุบันที่มี edge เชื่อมกับกลุ่มเวอร์เท็กซ์ถัดไป จากนั้นทำการเรียง weight ของ edge ตามลำดับน้อยไปหามาก และทำการเลือกเวอร์เท็กซ์ที่มี weight น้อยที่สุด เพื่อสร้าง tree และทำการค้นหาเวอร์เท็กซ์ถัดไปที่มี weight น้อยที่สุดต่อไปจนครบทุกเวอร์เท็กซ์

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

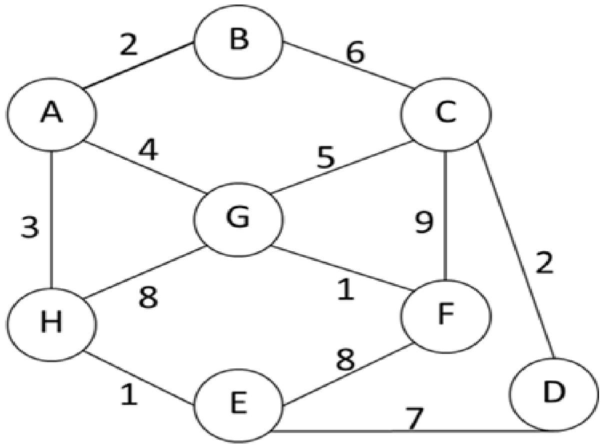
ซูโดโค้ดขั้นตอนวิธีของพริม มีขั้นตอนดังนี้

```
Prim(G, w, f)
PQ  ← Ø
For each u ∈ V|G| - |r| do
            key[u] ← ∞
            π[u] ← null
key[r] ← 0
PQ ←V|G|
While PQ ≠ Ø do
            u ← dequeuer PQ
for each v ∈ Adj[u] do
    if v ∈ PQ and w(u,v) < key[v]
    then π [v] ← u
           key[v] ← w(u,v)
```

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด

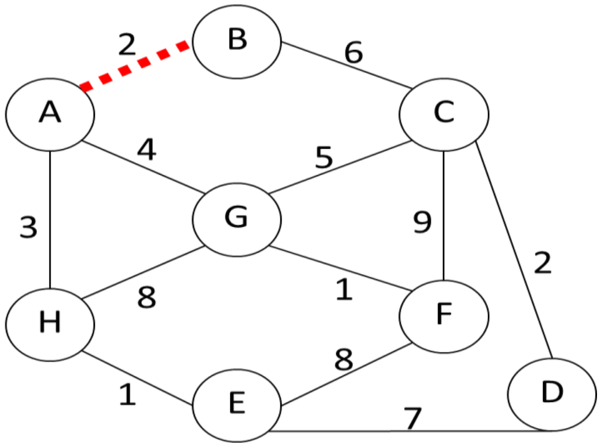
จากกราฟต่อไปนี้



กำหนดให้เวอร์เท็กซ์ A เป็นเวอร์เท็กซ์เริ่มต้น

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)

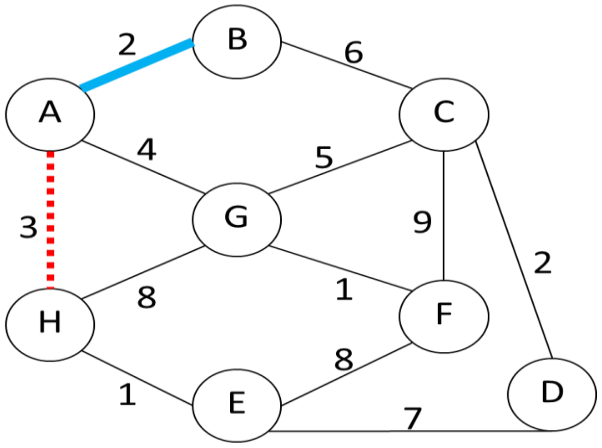


- **ทำการเลือกเส้นเชื่อม รอบที่ 1**
- เวอร์เท็กซ์ A ทำการเลือกเส้นเชื่อม AB, AH, AG เข้าสู่ลำดับการเลือก
- แต่น้ำหนักเส้นเชื่อม AB มีค่าน้ำหนักน้อยสุดให้ทำการเลือกเส้นเชื่อมนี้

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	เอดจ์ weight	เลือก
A	B	2	*
A	H	3	
A	G	4	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)

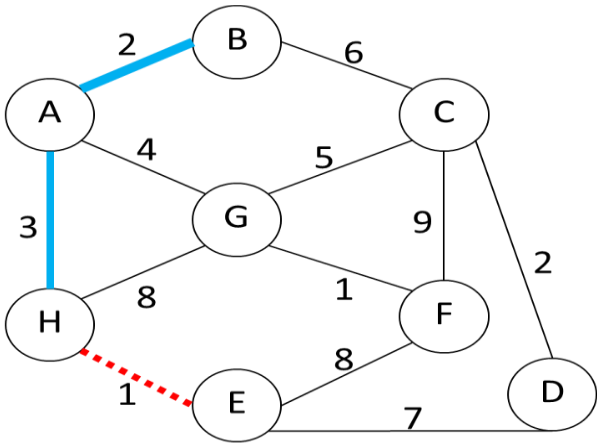


- **ทำการเลือกเส้นเชื่อม รอบที่ 2**
- เวอร์เท็กซ์ A ทำการเลือกเส้นเชื่อม AH, AG, BC เข้าสู่ลำดับการเลือก
- แต่น้ำหนักเส้นเชื่อม AH มีค่าน้ำหนักน้อยสุดให้ทำการเลือกเส้นเชื่อมนี้

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	เอดจ์ weight	เลือก
A	H	3	*
A	G	4	
B	C	6	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)

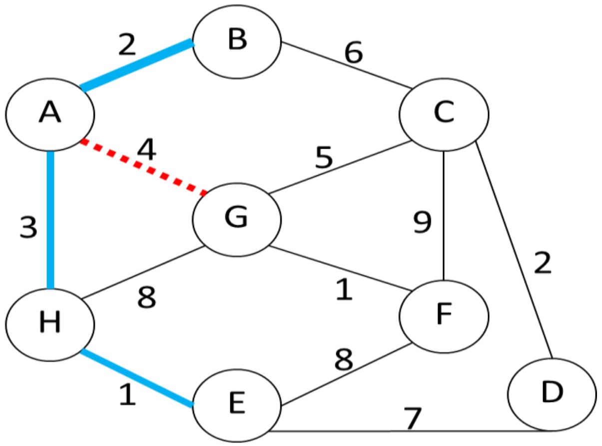


- **ทำการเลือกเส้นเชื่อม รอบที่ 3**
- เวอร์เท็กซ์ H ทำการเลือกเส้นเชื่อม HE, HG, AG, BC เข้าสู่ลำดับการเลือก
- แต่น้ำหนักเส้นเชื่อม HE มีค่าน้ำหนักน้อยสุดให้ทำการเลือกเส้นเชื่อมนี้

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	เอดจ์ weight	เลือก
H	E	1	*
H	G	8	
A	G	4	
B	C	6	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)

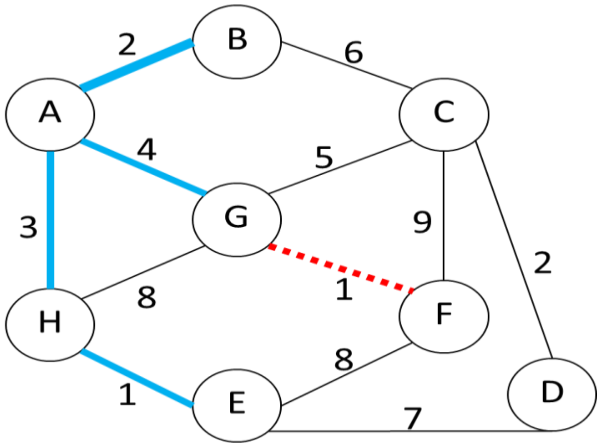


- **ทำการเลือกเส้นเชื่อม รอบที่ 4**
- เวอร์เท็กซ์ E ทำการเลือกเส้นเชื่อม AG, BC, ED, EF, HG เข้าสู่ลำดับการเลือก แ
- ต่น้ำหนักเส้นเชื่อม AG มีค่าน้ำหนักน้อยสุดให้ทำการเลือกเส้นเชื่อมนี้

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	เอดจ์ weight	เลือก
A	G	4	*
B	C	6	
E	D	7	
E	F	8	
H	G	8	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

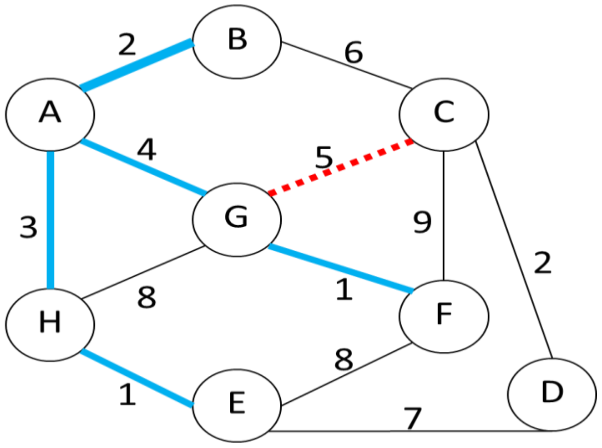
ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)



- **ทำการเลือกเส้นเชื่อม รอบที่ 5**
- เวอร์เท็กซ์ G ทำการเลือกเส้นเชื่อม GF, GC, BC, ED, EF, HG เข้าสู่ลำดับการเลือก
- แต่น้ำหนักเส้นเชื่อม GF มีค่าน้ำหนักน้อยสุดให้ทำการเลือกเส้นเชื่อมนี้

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	เอดจ์ weight	เลือก
G	F	1	*
G	C	5	
B	C	6	
E	D	7	
E	F	8	
H	G	8	

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)

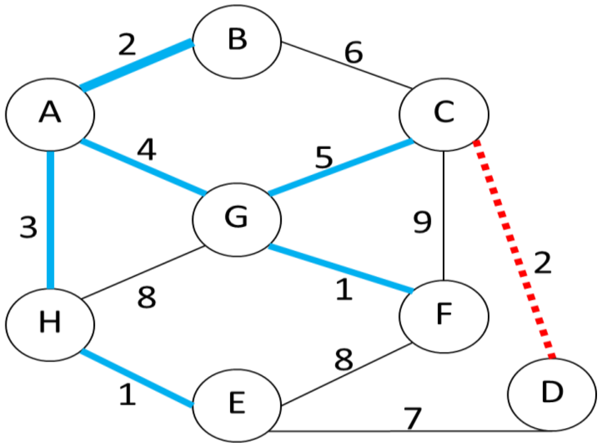


- **ทำการเลือกเส้นเชื่อม รอบที่ 6**
- เวอร์เท็กซ์ F ทำการเลือกเส้นเชื่อม GC, BC, ED, EF, HG, FC เข้าสู่ลำดับการเลือก
- แต่น้ำหนักเส้นเชื่อม GC มีค่าน้ำหนักน้อยสุดให้ทำการเลือกเส้นเชื่อมนี้

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	เอดจ์ weight	เลือก
G	C	5	*
B	C	6	
E	D	7	
E	F	8	
H	G	8	
F	C	9	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)

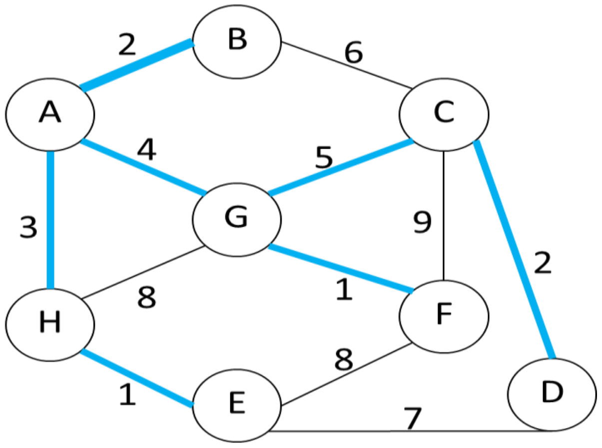


- **ทำการเลือกเส้นเชื่อม รอบที่ 7**
- เวอร์เท็กซ์ E ทำการเลือกเส้นเชื่อม AG, BC, ED, EF, HG เข้าสู่ลำดับการเลือก
- แต่น้ำหนักเส้นเชื่อม AG มีค่าน้ำหนักน้อยสุดให้ทำการเลือกเส้นเชื่อมนี้

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	เอดจ์ weight	เลือก
C	D	2	*
B	C	6	
E	D	7	
E	F	8	
H	G	8	
F	C	9	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)



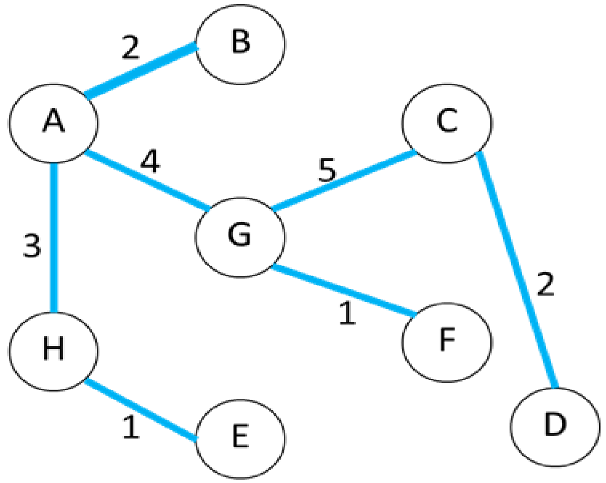
- **ทำการเลือกเส้นเชื่อม รอบที่ 8**
- พบว่าทุกเวอร์เท็กซ์ถูกทำการเชื่อมต่อกันแล้ว

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	เอดจ์ weight	เลือก
B	C	6	
E	D	7	
E	F	8	
H	G	8	
F	C	9	

ตัวอย่างที่ 8.8 จงใช้ขั้นตอนวิธีของพริมในการสร้างต้นไม้ ที่มีระยะทางที่สั้นที่สุด (ต่อ)

ผลการหาเส้นทางที่สั้นที่สุดด้วยขั้นตอนวิธีของพริม

พบว่าทำการเชื่อมต่อทุกเวอร์เท็กซ์แล้ว ดังนั้นน้ำหนักรวม =2+3+1+4+1+5+2 =18 ดังนี้

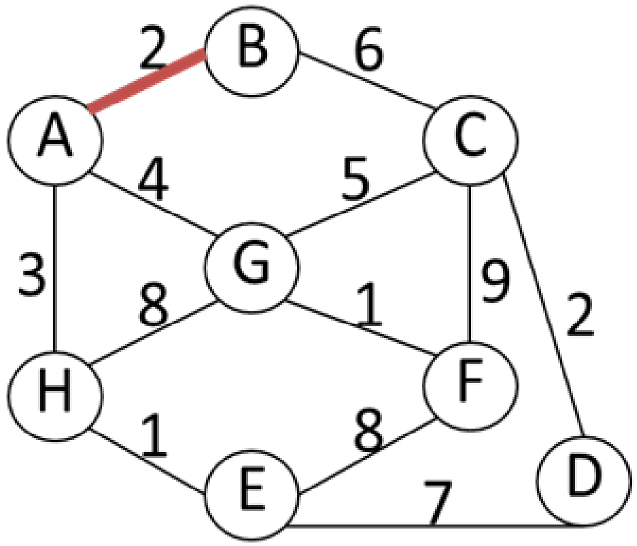


8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

8.5.3 ขั้นตอนวิธีของไดก์สตรา

- ขั้นตอนวิธีของไดก์สตรา (dijkstra algorithm) เป็นการหาระยะทางสั้นที่สุด
- มีความคล้ายกับอัลกอริทึมการค้นหาในแนวกว้าง
- มีความแตกต่างคือจะมีน้ำหนักกำกับในแต่ละเส้นเชื่อม
- อัลกอริทึมไดก์สตราถูกนำเสนอโดย Edsger Dijkstra นักวิทยาการคอมพิวเตอร์ ในปี ค.ศ. 1950 เพื่อหาเส้นทางสั้นที่สุดในการเดินทางเชื่อมต่อแต่ละสถานที่ โดยมีเงื่อนไขที่น้ำหนักของเส้นทางเชื่อมต้องไม่น้อยกว่า 0 และกราฟสามารถเป็นแบบมีทิศทางหรือไม่มีทิศทางได้เช่นเดียวกัน

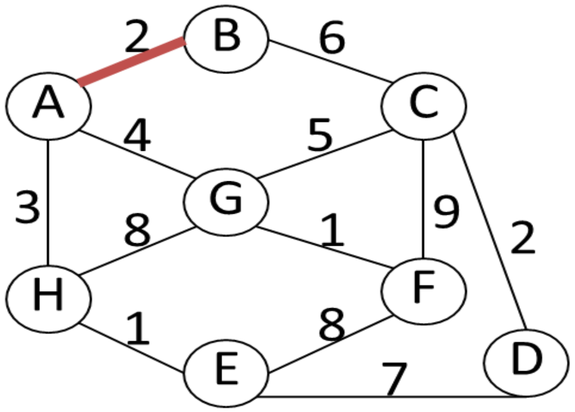
ตัวอย่างที่ 8.9 จงสร้าง tree ที่มีระยะทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตรา



กำหนดให้เวอร์เท็กซ์ A เป็นเวอร์เท็กซ์เริ่มต้น

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.9 จงสร้าง tree ที่มีระยะทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตรา (ต่อ)

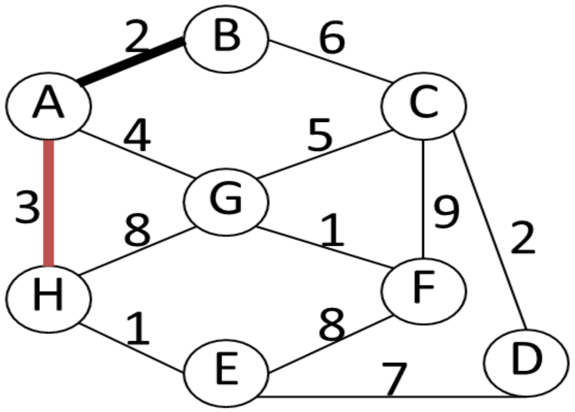


- **รอบที่ 1** ทำการเลือกเวอร์เท็กซ์ที่เชื่อมต่อกับเวอร์เท็กซ์ A
- ทำการรวมระยะทางและทำการเลือกเส้นทางที่สั้นที่สุด คือ AB=0+2=2

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	Weightเดิม +weight ใหม่ =weight รวม	เลือก
A	B	0+2=2	*
A	H	0+3=3	
A	G	0+4=4	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.9 จงสร้าง tree ที่มีระยะทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตรา (ต่อ)

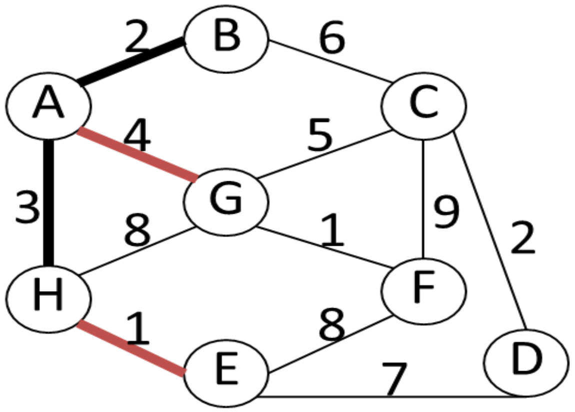


- **รอบที่** 2 ทำการเลือกเวอร์เท็กซ์ที่เชื่อมต่อกับเวอร์เท็กซ์ A และ A, B
- ทำการรวมระยะทางและทำการเลือกเส้นทางที่สั้นที่สุด คือ AH=0+3=3

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	Weightเดิม +weight ใหม่ =weight รวม	เลือก
A	H	0+3=3	*
A	G	0+4=4	
A,B	C	2+6=8	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.9 จงสร้าง tree ที่มีระยะทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตรา (ต่อ)

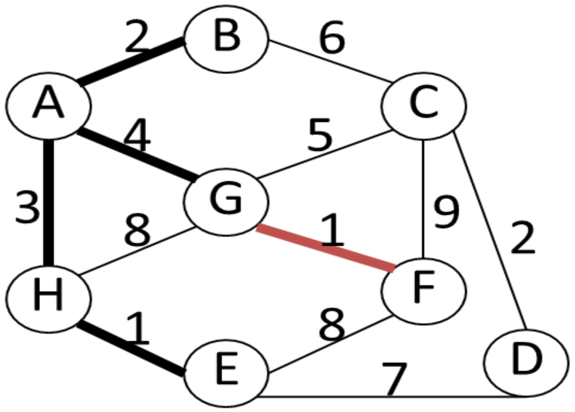


- **รอบที่ 3** ทำการเลือกเวอร์เท็กซ์ที่เชื่อมต่อกับเวอร์เท็กซ์ A และ A, H และ A, B
- ทำการรวมระยะทางและทำการเลือกเส้นทางที่สั้นที่สุด คือ AHE=3+1=4 และ AG=0+4=4

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	Weightเดิม +weight ใหม่ =weight รวม	เลือก
A,H	G	3+8=11	
A,H	E	3+1=4	*
A	G	0+4=4	*
A,B	C	2+6=8	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.9 จงสร้าง tree ที่มีระยะทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตรา (ต่อ)

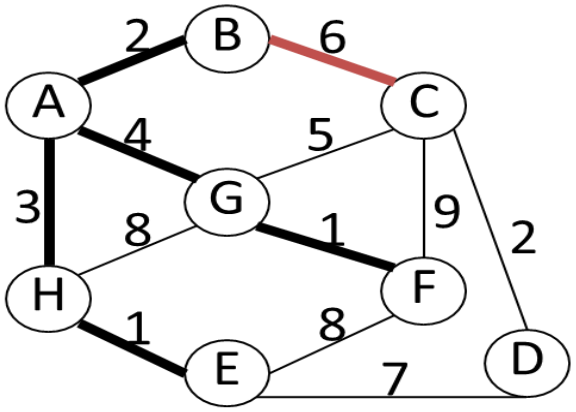


- **รอบที่ 4** ทำการเลือกเวอร์เท็กซ์ที่เชื่อมต่อกับเวอร์เท็กซ์ A, H, E และ A, H และ A, B และ A, G
- ทำการรวมระยะทางและทำการเลือกเส้นทางที่สั้นที่สุด คือ A, G, F =4+1=5

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	Weightเดิม +weight ใหม่ =weight รวม	เลือก
A,H,E	F	4+8=12	
A,H,E	D	4+7=11	
A,H	G	3+8=11	
A,B	C	2+6=8	
A,G	C	4+5=9	
A,G	F	4+1=5	*

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.9 จงสร้าง tree ที่มีระยะทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตรา (ต่อ)

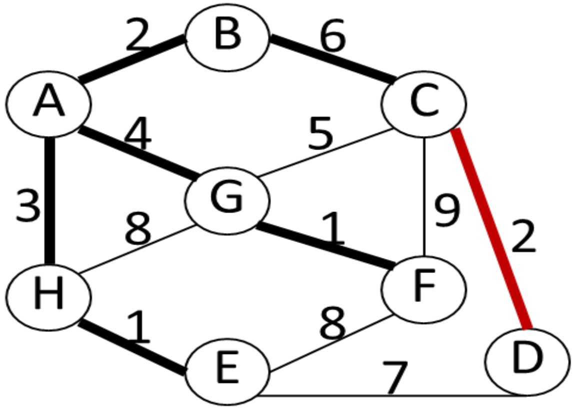


- **รอบที่ 5** ทำการเลือกเวอร์เท็กซ์ที่เชื่อมต่อกับเวอร์เท็กซ์ A,B และ A,G และ A,G,F และ A,H,E
- ทำการรวมระยะทางและทำการเลือกเส้นทางที่สั้นที่สุด คือ A,B,C =2+6=8

เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	Weightเดิม +weight ใหม่ =weight รวม	เลือก
A,B	C	2+6=8	*
A,G	C	4+5=9	
A,G	H	Cycle	
A,G,F	C	5+9=14	
A,G,F	E	5+8=13	
A,H,E	F	4+8=12	
A,H,E	D	4+7=11	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.9 จงสร้าง tree ที่มีระยะทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตรา (ต่อ)



- **รอบที่ 6** ทำการเลือกเวอร์เท็กซ์ที่เชื่อมต่อกับเวอร์เท็กซ์ A,B,C และ A,H,E
- ทำการรวมระยะทางและทำการเลือกเส้นทางที่สั้นที่สุด คือ A,B,C,D =8+2=10

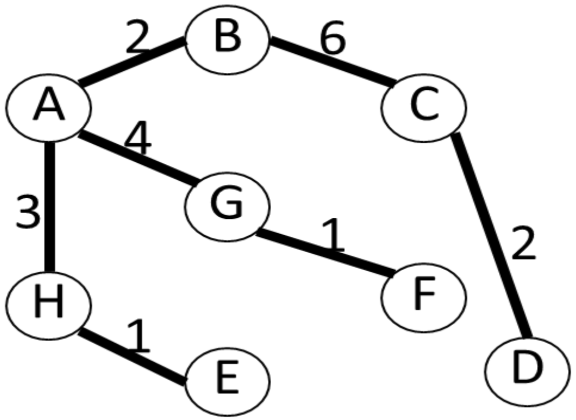
เวอร์เท็กซ์เริ่ม	เวอร์เท็กซ์ถัดไป	Weightเดิม +weight ใหม่ =weight รวม	เลือก
A,B,C	D	8+2=10	*
A,B,C	G	Cycle	
A,B,C	F	Cycle	
A,H,E	F	Cycle	
A,H,E	D	4+7=11	

8.5 การค้นหาเส้นทางที่สั้นที่สุด(ต่อ)

ตัวอย่างที่ 8.9 จงสร้าง tree ที่มีระยะทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตรา (ต่อ)

ผลการหาเส้นทางที่สั้นที่สุดด้วยขั้นตอนวิธีของไดก์สตราพบว่าทำการเชื่อมต่อทุกเวอร์เท็กซ์แล้ว

ดังนั้นน้ำหนักรวม =2+6+2+4+1+3+1 =19 ดังนี้



มหาวิทยาลัยราชภัฏนครปฐม