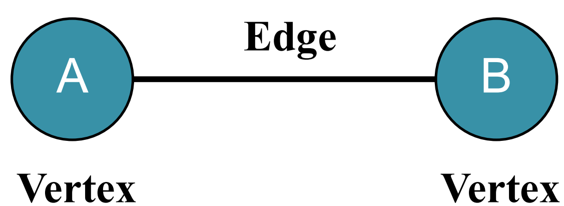
Chapter 7

Graph

Graph

- เป็นโครงสร้างข้อมูล ที่นำมาใช้แก้ปัญหาลักษณะงานแบบเครือข่าย (Network)
- ประกอบด้วย 2 ส่วน
  - Vertex เป็นกลุ่มโหนดในโครงสร้าง (A และ B)
  - Edge เป็นเส้นเชื่อมระหว่างโหนด (เส้นเชื่อมจาก A ไป B แทนเป็น (A,B))

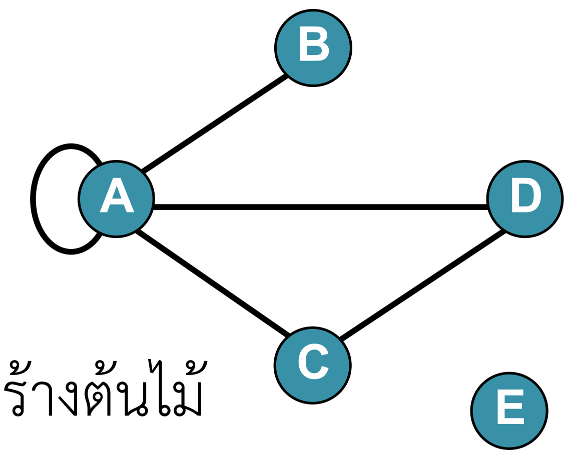


Graph (cont.)

- กำหนดให้ G เป็นสัญลักษณ์แทนกราฟ, V แทนกลุ่มเวอร์เทกซ์ และ E แทนเส้นเชื่อมเวอร์เทกซ์ จะได้ว่า
- G = (V,E)
  - V(G) คือ เซตของเวอร์เทกซ์ ไม่ใช่เซตว่าง และมีจำนวนจำกัด
  - E(G) คือ เซตของเส้นเชื่อมระหว่างเวอร์เทกซ์

Example

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



- ***<u>สังเกต</u>*** โครงสร้างแบบกราฟคล้ายกับโครงสร้างต้นไม้

ต่างกันที่โหนดหนึ่งๆ ในต้นไม้ จะเชื่อมจากโหนดก่อนหน้า

(โหนดพ่อแม่) ได้เพียงโหนดเดียวเท่านั้น

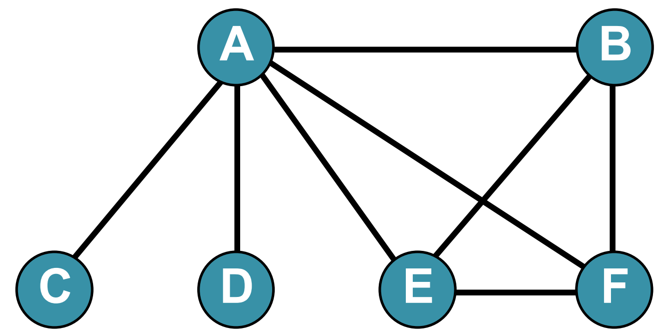
- โครงสร้างต้นไม้ เป็นโครงสร้างกราฟที่ไม่มี cycle นั่นเอง

Directed and Undirected graph

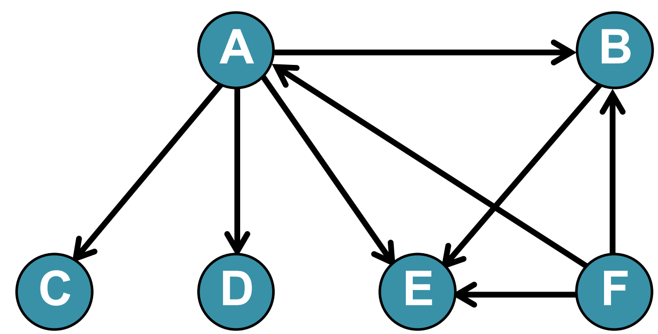
- กราฟแบ่งเป็น 2 แบบ
  - กราฟแบบไม่มีทิศทาง (Undirected graph)
    - Edge หรือเส้นที่เชื่อมต่อเวอร์เทกซ์ไม่มีทิศทาง (สามารถเดินได้ 2 ทิศ ไป-กลับ)
  - กราฟแบบมีทิศทาง (Directed graph : digraph)
    - Edge หรือเส้นที่เชื่อมต่อเวอร์เทกซ์มีทิศทาง (ต้องเดินตามทิศของหัวลูกศร)

Example

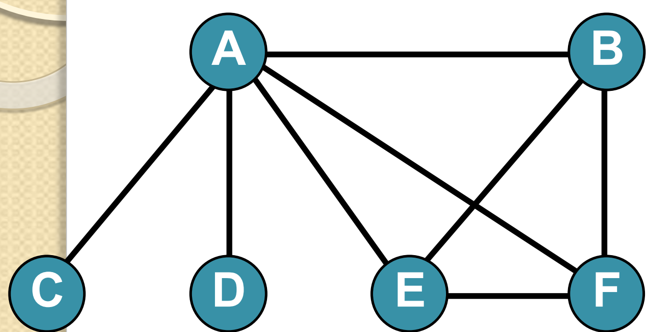
- V(G) = {A,B,C,D,E,F}
- E(G) = {(A,B), (A,C), (A,D), (A,E), (A,F), (B,E), (B,F), (E,F)}
- E(G) = {(B,A), (C,A), (D,A), (E,A), (F,A), (E,B), (B,F), (E,F)}

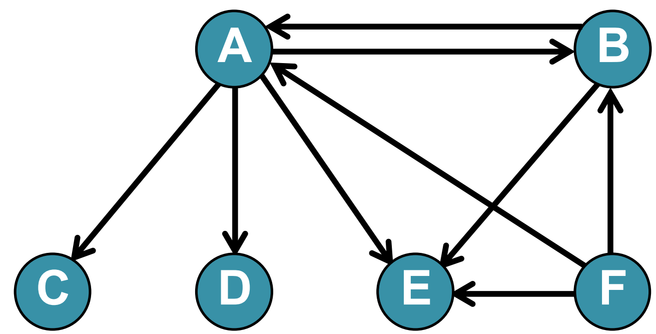


- V(G) = {A, B, C, D, E, F}
- E(G) = {(A,B), (A, C), (A, D), (A, E), (B, E), (F, A), (F, B), (F, E)}



Indegrees and outdegrees





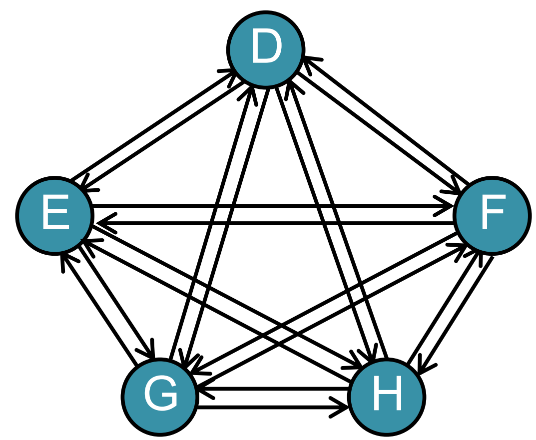
Vertex	Indegrees	Outdegrees
A	5	5
B	3	3
C	1	1
D	1	1
E	3	3
F	3	3

Vertex	Indegrees	Outdegrees
A	2	4
B	2	2
C	1	0
D	1	0
E	3	0
F	0	3

Complete Graphs

- กราฟที่ทุกเวอร์เทกซ์มีเอดจ์เชื่อมโยงไปยังเวอร์เทกซ์ที่เหลือทั้งหมด

Undirected graph



Directed graph

จน. vertex

Number of edges = $\frac{N(N-1)}{2}$

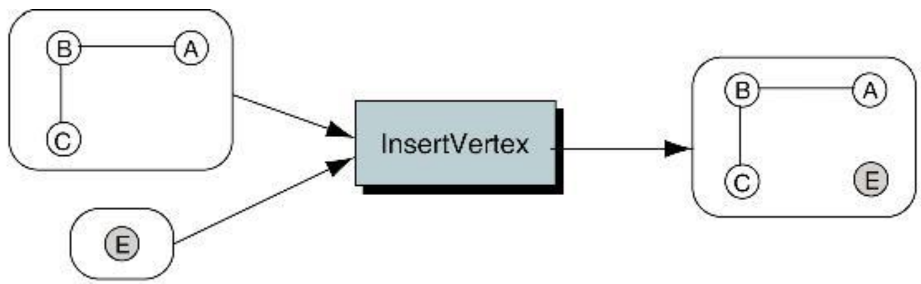
Number of edges = $N(N-1)$

Operations

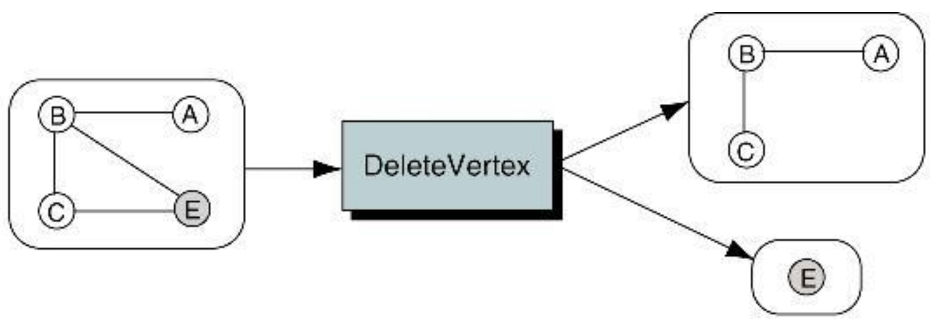
- Insert Vertex
- Delete Vertex
- Add Edge
- Delete Edge
- Find Vertex
- Traverse

Operations (cont.)

- Insert Vertex : แทรกเวอร์เทกซ์เข้าไปในกราฟ แต่เป็นการแทรกแบบยังไม่มีเอดจ์เชื่อมต่อ



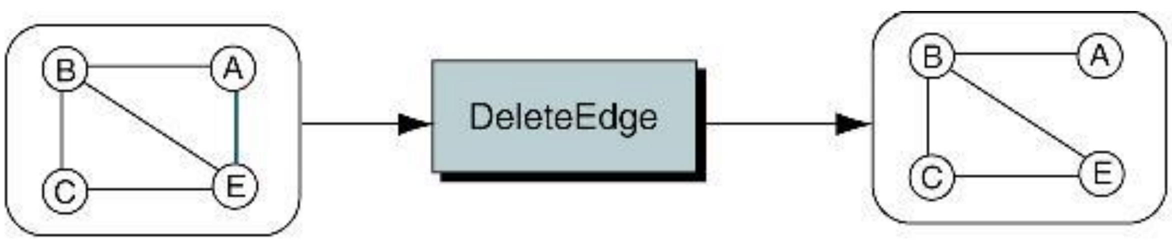
- Delete Vertex : ลบเวอร์เทกซ์ออกจากกราฟ ซึ่งจะลบทั้งเวอร์เทกซ์ และเอดจ์ที่เชื่อมต่อเวอร์เทกซ์นั้นด้วย



Operations (cont.)

- Add Edge : เพิ่มเส้นเชื่อมต่อระหว่างเวอร์เทกซ์ ซึ่งเพิ่มได้ทีละเอดจ์ ถ้าต้องการสร้างเอดจ์ในกราฟที่มีทิศทาง ต้องระบุว่าเวอร์เทกซ์ไหนเป็นส่วนเริ่มต้นและปลายทางให้ชัดเจน

- Delete Edge : ลบเอดจ์ออกจากกราฟ

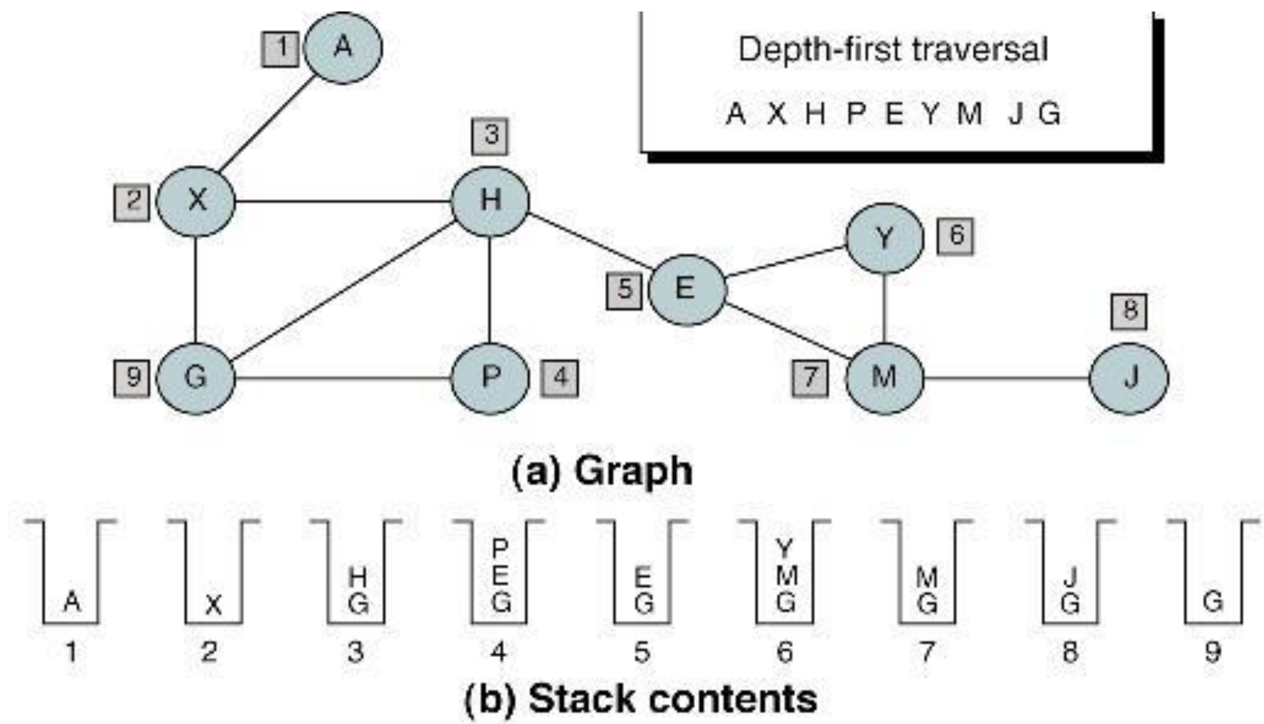


Operations (cont.)

- Find Vertex : ค้นหาเวอร์เทกซ์ที่ต้องการ ถ้าพบแล้วจะคืนค่าข้อมูลนั้น

Traverse Graph

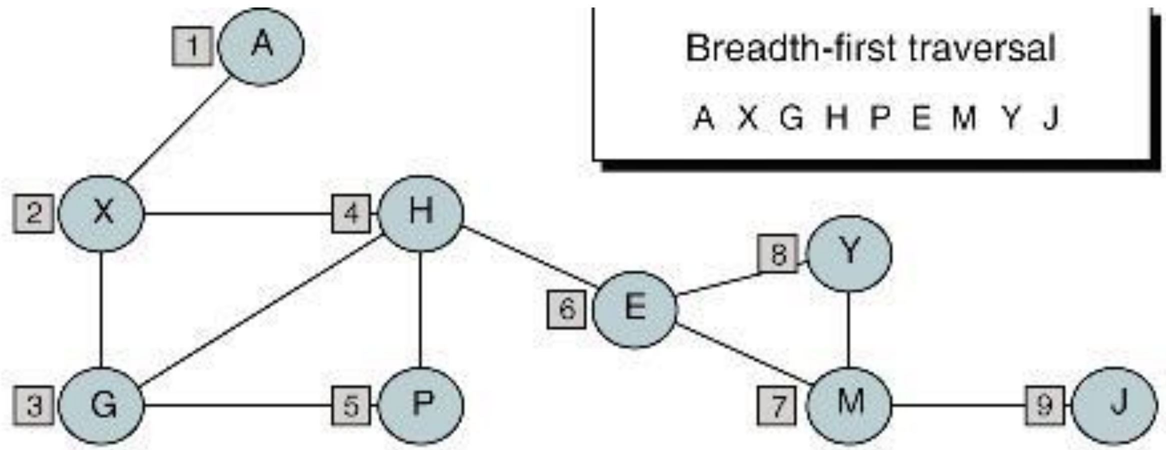
- Depth-first Traversal : ท่องเข้าไปในเวอร์เทกซ์ข้างเคียงตัวใดตัวหนึ่งให้หมดก่อน จึงย้ายไปท่องในเวอร์เทกซ์ที่เหลือในระดับเดียวกัน
- นำ Stack มาช่วยในการท่องเข้าไปในกราฟ



Traverse Graph

FIFO

- Breadth-first Traversal : ท่องเข้าไปยังเวอร์เทกซ์ข้างเคียงในระดับเดียวกันทั้งหมดก่อน จึงท่องเข้าไปยังเวอร์เทกซ์ในระดับต่อไป
- นำ Queue มาช่วยในการท่องเข้าไปในกราฟ



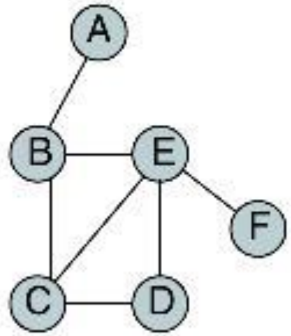
(a) Graph

(b) Queue contents

Graph Storage Structures

- สามารถสร้างกราฟได้โดยใช้
  - Adjacency Matrix :
    - ใช้อาร์เรย์ 1 มิติ เก็บเวอร์เทกซ์ต่างๆ
    - ใช้อาร์เรย์ 2 มิติ เก็บเอดจ์ทั้งหมดในกราฟ
  - Adjacency List :
    - ใช้ลิงค์ลิสต์ เก็บเวอร์เทกซ์ต่างๆ
    - ใช้ลิงค์ลิสต์ เก็บเอดจ์ทั้งหมดในกราฟ

Adjacency Matrix



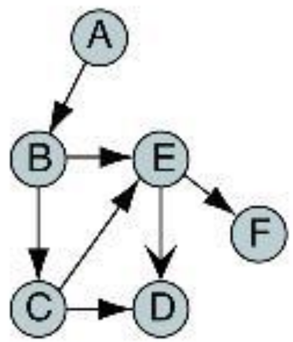
A
B
C
D
E
F

Vertex vector

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

Adjacency matrix

(a) Adjacency matrix for nondirected graph



A
B
C
D
E
F

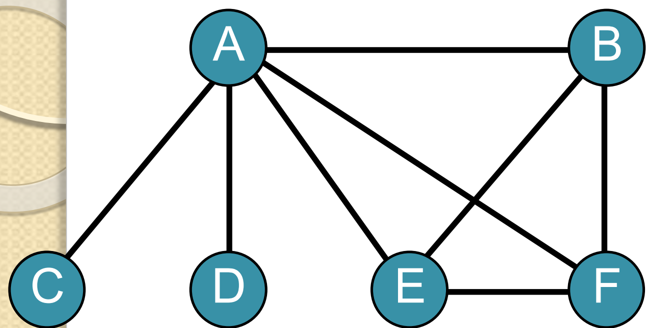
Vertex vector

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

Adjacency matrix

(b) Adjacency matrix for directed graph

Adjacency matrix



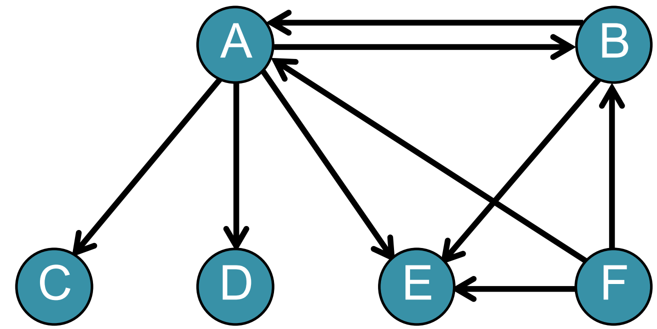
A
B
C
D
E
F

Vertex vector

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

Adjacency matrix

Adjacency matrix for undirected graph



A
B
C
D
E
F

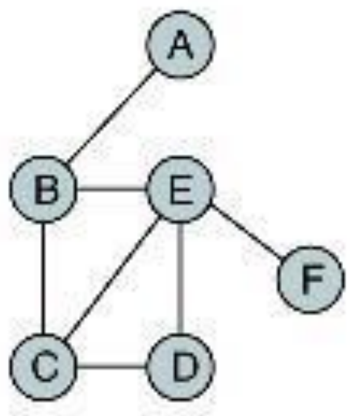
Vertex vector

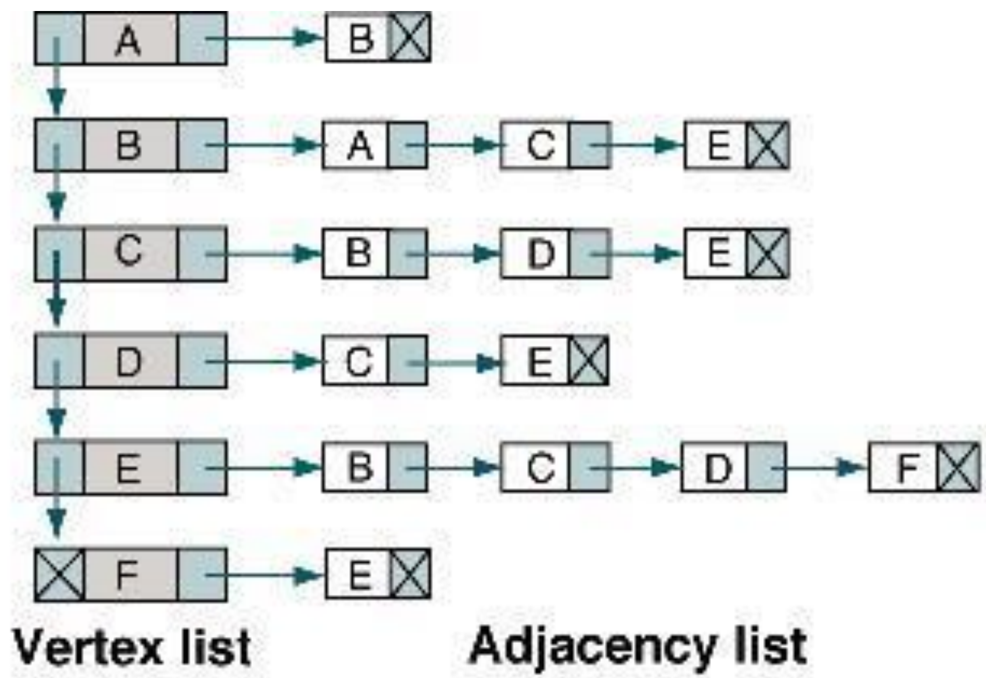
	A	B	C	D	E	F
A	0	1	1	1	1	0
B	1	0	0	0	1	0
C	0	0	0	0	0	0
D	0	0	0	0	0	0
E	0	0	0	0	0	0
F	1	1	0	0	1	0

Adjacency matrix

Adjacency matrix for directed graph

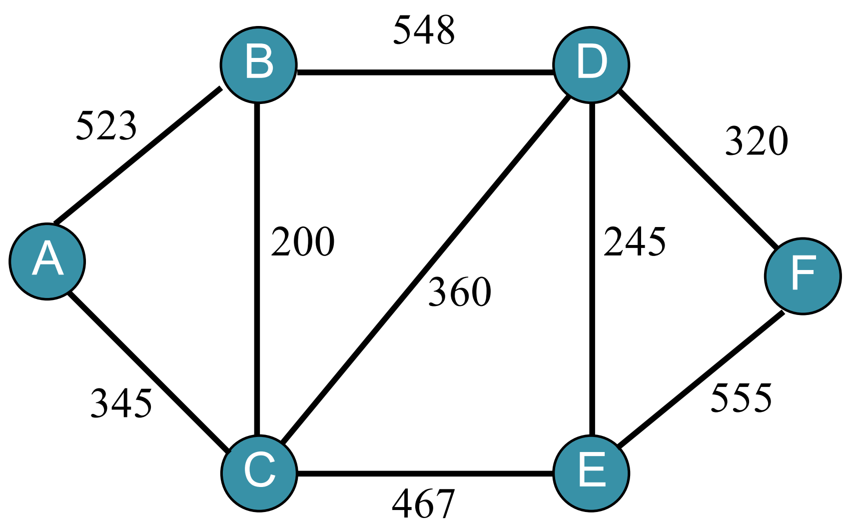
Adjacency List

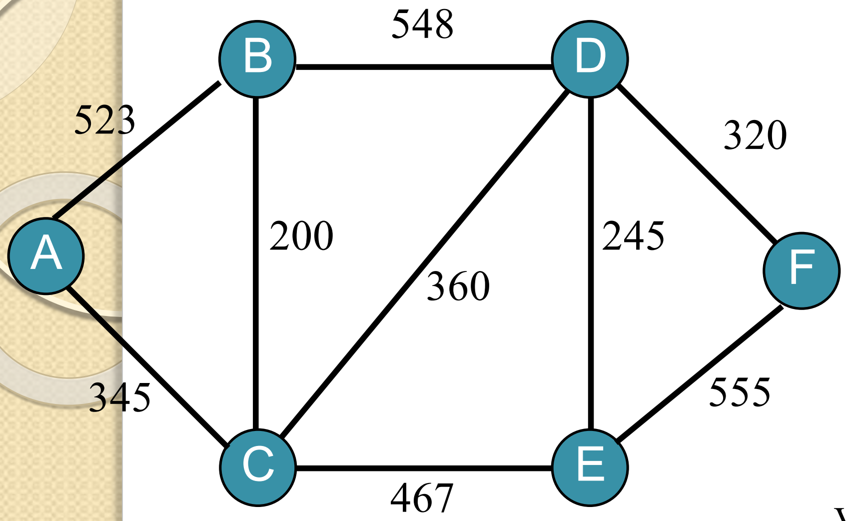




Network

- เป็นกราฟที่มีการถ่วงน้ำหนัก (Weight) บนเอดจ์ด้วย หรือเรียกว่า Weighted graph
- ค่าน้ำหนักที่กำหนดจะขึ้นอยู่กับการประยุกต์ใช้งาน เช่น กราฟแสดงเส้นทางของเครื่องบิน ค่าน้ำหนักจะเป็นระยะทางระหว่างแต่ละเวอร์เทกซ์ (เมืองต่างๆ)





City Network

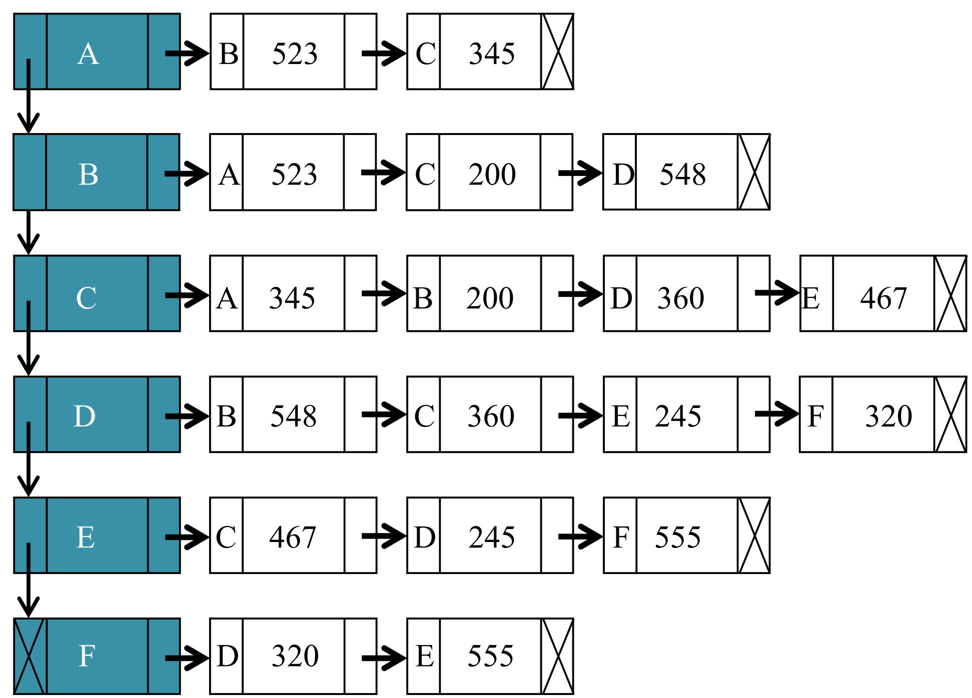
	A	B	C	D	E	F
A	0	523	345	0	0	0
B	523	0	200	548	0	0
C	345	200	0	360	467	0
D	0	548	360	0	245	320
E	0	0	467	245	0	555
F	0	0	0	320	555	0

Vertex vector
A
B
C
D
E
F

Vertex vector

Adjacency Matrix

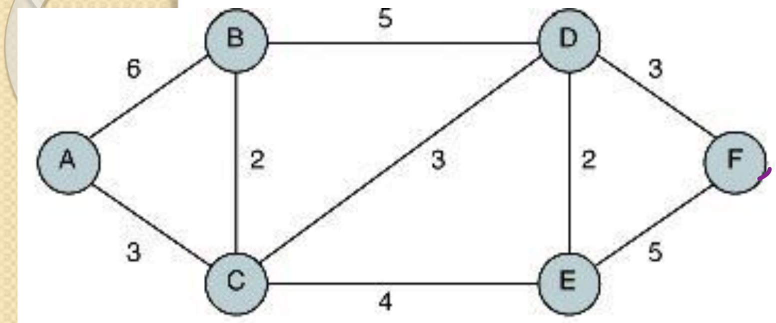
Vertex vector Adjacency List

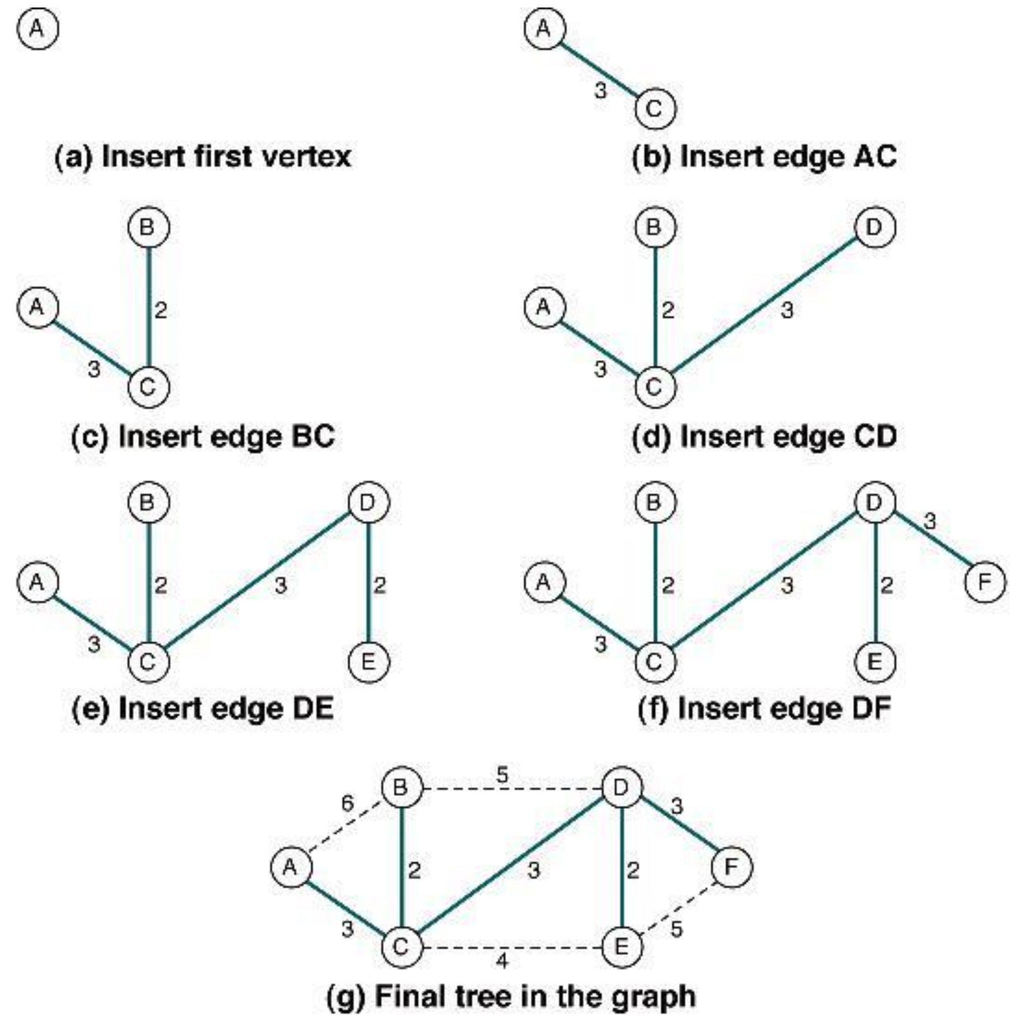


Minimum Spanning Tree

- เป็นโครงสร้างต้นไม้ที่ประกอบด้วยเวอร์เทกซ์ทั้งหมดในกราฟ
- รับประกันว่าค่าน้ำหนักรวมของเอดจ์น้อยที่สุด
- สามารถสร้าง Minimum Spanning Tree จากกราฟได้ดังนี้
  - กำหนดเวอร์เทกซ์เริ่มต้น
  - หาเวอร์เทกซ์ข้างเคียงของเวอร์เท็กซ์ที่มี แล้วเลือกใช้เส้นทางที่มีค่าน้ำหนักน้อยที่สุด *<u>ซึ่งต้องไม่ใช่เส้นทางที่เชื่อมไปยังเวอร์เทกซ์ที่เคยเลือกมาแล้ว</u>*
  - ทำไปเรื่อยๆ จนครบทุกเวอร์เทกซ์

Minimum Spanning Tree





(a) Insert first vertex

(b) Insert edge AC

(c) Insert edge BC

(d) Insert edge CD

(e) Insert edge DE

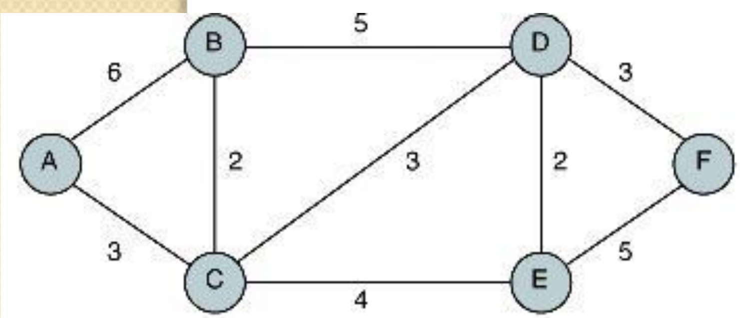
(f) Insert edge DF

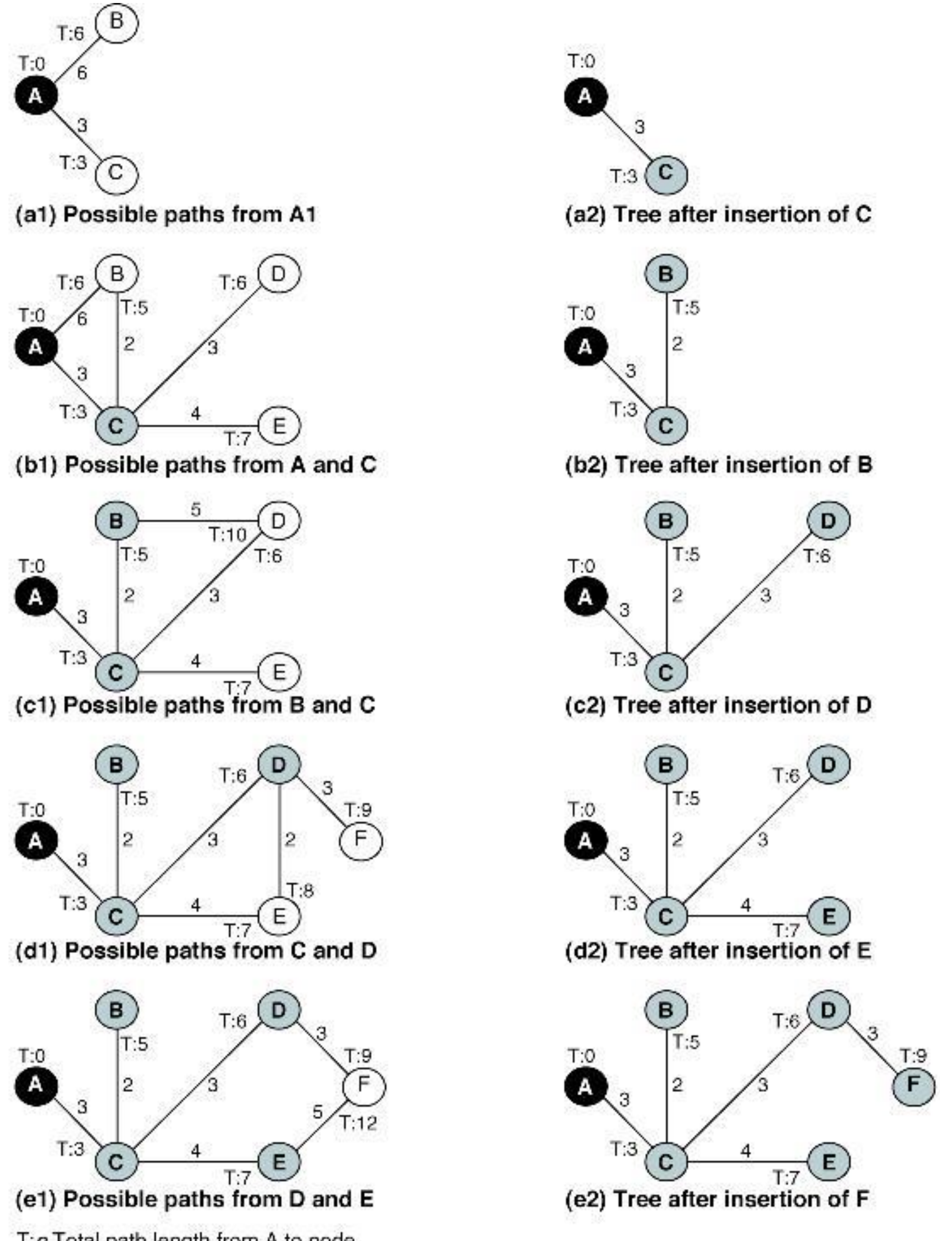
(g) Final tree in the graph

Shortest Path Algorithm

- สามารถค้นหาเส้นทางสั้นที่สุดระหว่างเวอร์เทกซ์ใดๆ ได้โดยใช้ Dijkstra algorithm ซึ่งมีหลักการดังนี้
  - กำหนดเวอร์เทกซ์เริ่มต้น และแทรกเข้าไปในต้นไม้
  - พิจารณาเวอร์เทกซ์ถัดไปที่เป็นไปได้ทั้งหมด
  - ให้เลือกเวอร์เทกซ์ที่มีค่าผลรวมระยะทางจากเวอร์เทกซ์เริ่มต้นน้อยที่สุด (ค่าน้ำหนักน้อยที่สุด)
  - แทรกเวอร์เทกซ์นั้นเข้าไปในต้นไม้
  - พิจารณาเวอร์เทกซ์ถัดไปของโหนดทั้งหมดในต้นไม้ หาเวอร์เทกซ์ที่ทำให้ค่าผลรวมน้อยที่สุด ทำไปเรื่อยๆ จนกว่าจะครบทุกเวอร์เทกซ์

Shortest Path Algorithm





Quiz 1

1) จงเขียนกราฟแสดงความสัมพันธ์ของคนต่างๆ

People = {George, Jim, Jean, Frank, Fred, John, Susan}

Friendship = { (George, Jean), (Frank, Fred), (George, John),
(Jim, Fred), (Jim, Frank), (Jim, Susan),
(Susan, Frank) }

2) จากกราฟข้างต้น จงตอบคำถามต่อไปนี้

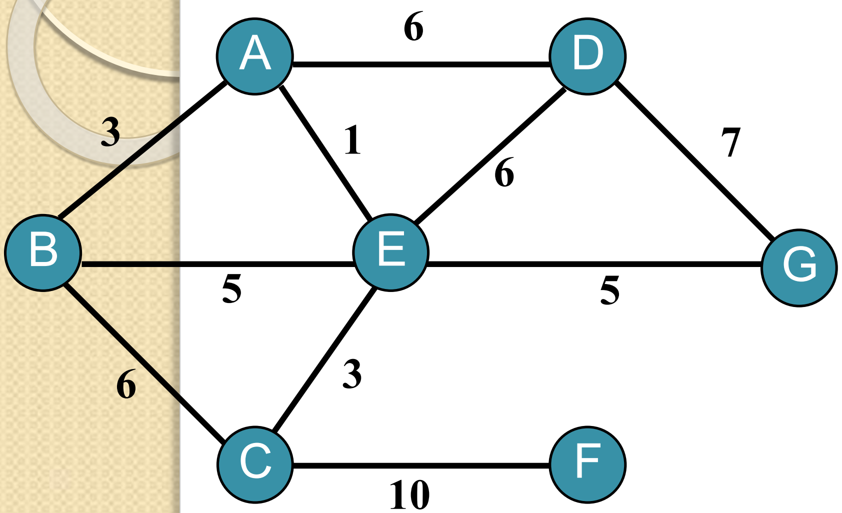
2.1) จงหาเพื่อนทั้งหมดของ John

2.2) จงหาเพื่อนทั้งหมดของ Susan

2.3) จงหาเพื่อนทั้งหมดของเพื่อนของ Jean

2.4) จงหาเพื่อนทั้งหมดของเพื่อนของ Jim

Quiz 2



1) จงเขียน Adjacency Matrix แสดงกราฟนี้
2) จงเขียน Adjacency List แสดงกราฟนี้
3) จงเขียน Minimum Spanning Tree ของกราฟนี้
4) จงเขียน Shortest Path จาก Vertex B ไปยัง Vertex อื่นๆ