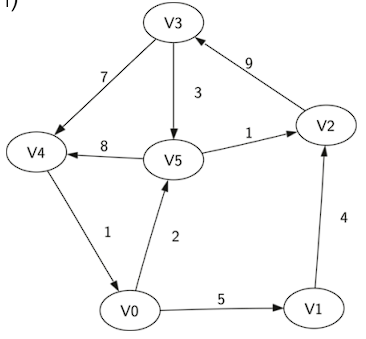
**Graph**

• กราฟ (Graph) เป็นโครงสร้างข้อมูลแบบไม่เชิงเส้น (Non-linear Data Structure)

เช่นเดียวกันกับ Tree

• Graph ถูกนําไปประยุกต์ใช้ในการจัดเก็บข้อมูล และในงานด้านต่างๆ เช่น

• การแสดงความสัมพันธ์ระหว่างบุคคลในเครือข่ายสังคม เช่น Facebookซึ่ง Nodes แทน

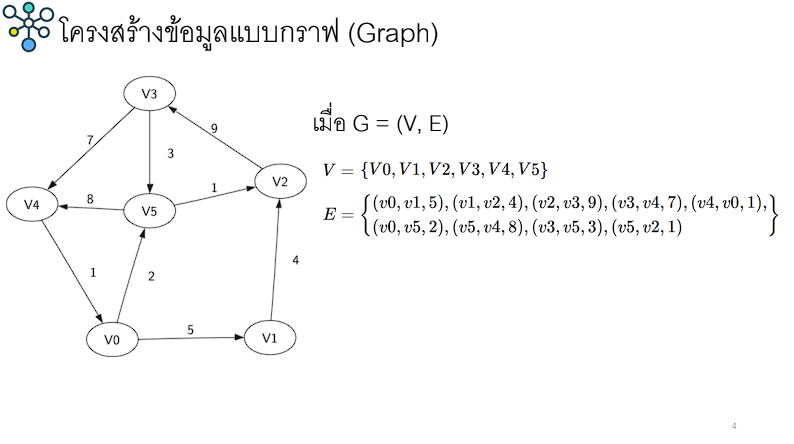
บุคคลและ Edges แทนความสัมพันธ์

• การวางแผนเส้นทางที่เร็วที่สุด เช่น การวางแผนทางในระบบโทรคมนาคม, ระบบการ

ขนส่ง, หรือระบบเครือข่ายคอมพิวเตอร์.

โครงสร้างข้อมูลแบบกราฟ(Graph)

• Graph เป็นโครงสร้างข้อมูล ที่มีรายละเอียดดังนี้ • เซตของ Nodeและเซตของ Edge • ค่า Weight เช่น ค่าระยะทาง ความเร็ว • Pathจะใช้เรียกลําดับของ Node ที่เชื่อมต่อกันจาก

จุดหนึ่งไปยังอีกจุดหนึ่ง ตัวอย่างเช่น จาก

V3 to V1path คือ (v3, v4, v0, v1) และ edges คือ{(v3,v4,7),(v4,v0,1),(v0,v1,5)}

• Cycleจะใช้เรียก Path ที่มีจุดเริ่มต้นและสิ้นสุดที่

Node เดียวกัน เช่น path (v5, v2, v3, v5)

• Adjacent Node คือ Node ที่มีการเชื่อมโยงกัน

ประเภทกราฟตามลักษณะทิศทางของ Edge

Directed Graph (Digraph) • เป็นกราฟ ที่ Edge ที่เชื่อมระหว่าง

Node มีทิศทางกํากับความสัมพันธ์

• ตัวอย่างเช่น เส้นทางในการบินจะระบุทิศทางการบินระหว่างเมืองต่างๆ ของ• การค้นหาแบบแนวลึกเป็นการเข้าถึงหรือค้นหาข้อมูลในกราฟตามความลึกของเส้นทาง

•จะเริ่มจาก Node หนึ่งในกราฟไปยัง Node อื่นๆ จนกว่าจะไม่สามารถไปในเส้นทางนั้น

ได้อีกแล้ว

•จากนั้นจึงเข้าถึง Node ในกราฟที่เส้นทางอื่นต่อไปจนกว่าจะสามารถเข้าถึงทุก Node

ในกราฟ

สายการบินหนึ่ง

• ความสัมพันธ์ระหว่างสองจุดยอดใดๆ

จะมีลักษณะเป็นแบบลําดับ เช่น

(U,V)

จะหมายความว่า U เป็น Node เริ่มต้น

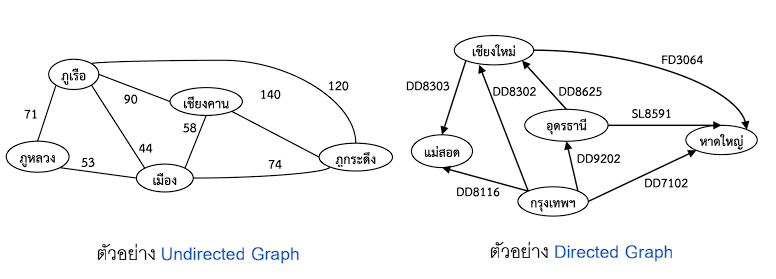
และ V คือ Node ปลายทาง เช่น

Undirected Graph

• กราฟที่ไม่ระบุทิศทางของความสัมพันธ์ระหว่าง node

• ตัวอย่างเช่น กราฟแสดงเส้นทางระหว่างอําเภอต่างๆ ในจังหวัดเลย มี edge เชื่อม

ระหว่างอําเภอเป็นกิโลเมตร เช่น ระยะทางของ “เชียงคาน – ภูกระดึง” คือ 140

กิโลเมตร เป็นต้น

**Graph Traversal** (การท่องไปในกราฟ)• กระบวนการในการท่องไปในกราฟมีด้วยกัน 2วิธี คือ

• Breadth-first Search : BFS การท่องไปในกราฟแบบแนวกว้าง • Depth-first Search : DFS การท่องไปในกราฟแบบแนวลึก

• การค้นหาแบบแนวลึกเป็นการเข้าถึงหรือค้นหาข้อมูลในกราฟตามความลึกของเส้นทาง

•จะเริ่มจาก Node หนึ่งในกราฟไปยัง Node อื่นๆ จนกว่าจะไม่สามารถไปในเส้นทางนั้น

ได้อีกแล้ว

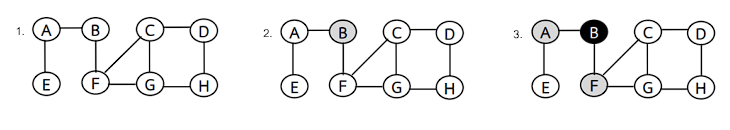
•จากนั้นจึงเข้าถึง Node ในกราฟที่เส้นทางอื่นต่อไปจนกว่าจะสามารถเข้าถึงทุก Node

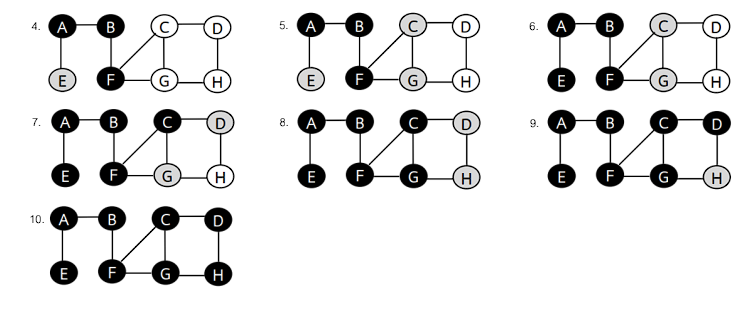
ในกราฟ

• การค้นหาแบบแนวกว้าง เป็นเทคนิคในการแวะผ่านกราฟแบบเข้าถึง Node หนึ่งๆ

•จะไปยังทุก Node ที่เป็น Node ใกล้เคียงทั้งหมดของ Node นั้นก่อน

•แล้วจึงไปยัง Node ลูกของ Node นั้นต่อไป

Breadth-first Search : BFS (B, A, F, E, C, G, D, H)

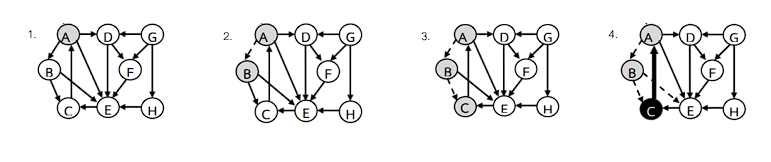
ตัวอย่างการค้นหาแบบแนวกว้าง เมื่อกําหนดให้

• Node สีขาว เมื่อ Node นั้นยังไม่เคยถูกค้นพบมาก่อน

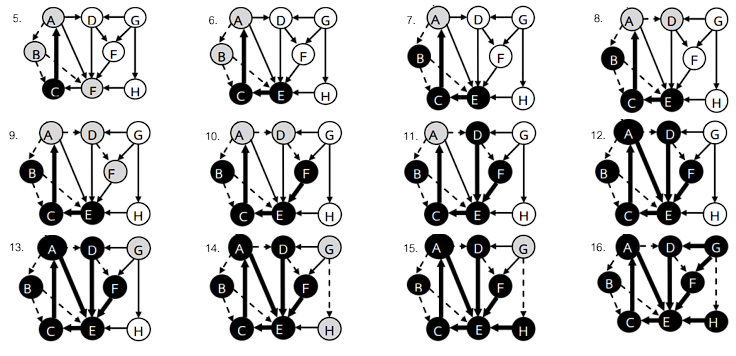
• Node สีเทา เมื่อ Node นั้นถูกค้นพบแล้วแต่ว่ายังสํารวจไม่เสร็จ

• Node สีดํา เมื่อ Node นั้นถูกค้นพบและสํารวจเสร็จแล้ว

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

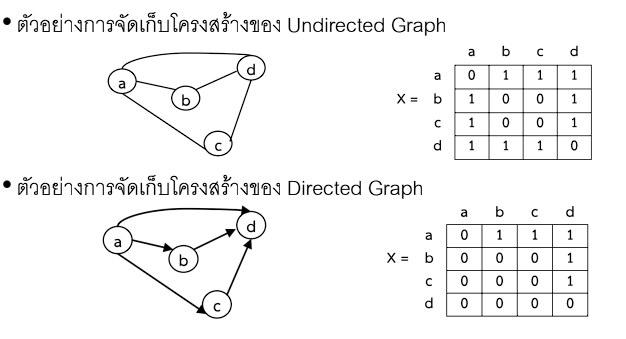
Depth-first Search : DFS (C, E, B, F, D, A, G, H)

ตัวอย่างการค้นหาแบบแนวกว้าง เมื่อกําหนดให้

• Node สีขาว เมื่อ Node นั้นยังไม่เคยถูกค้นพบมาก่อน

• Node สีเทา เมื่อ Node นั้นถูกค้นพบแล้ว

• Node สีดํา เมื่อ Node นั้นถูกสํารวจหรือเข้าถึงข้อมูลเสร็จแล้ว

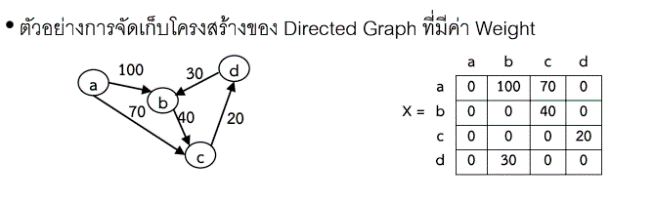
การนําโครงสร้างกราฟไปใช้ในการจัดเก็บข้อมูล

• การนําโครงสร้างข้อมูลกราฟไปใช้ในการจัดเก็บข้อมูล จะขึ้นอยู่กับลักษณะการ

จัดเก็บข้อมูล ซึ่งจะมีลักษณะการจัดเก็บอยู่ 2รูปแบบ ดังนี้

**Adjacency Matrix**: ใช้ Array 2 มิติสําหรับเก็บข้อมูลความสัมพันธ์ระหว่างNode ใกล้เคียงในกราฟ

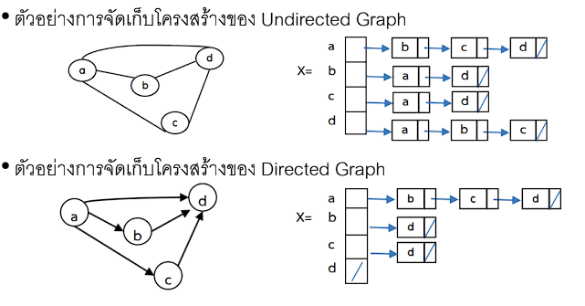
•เป็นโครงสร้างที่ประกอบไปด้วย Node และ Edge ที่บอกถึงเส้นทางของการเดินทาง

หรือความสัมพันธ์ในทิศทางซึ่งสามารถนํามาแทนความสัมพันธ์นั้นด้วยการกําหนด

เมตริกซ์ n x n โดยกําหนดให้

•0 : ไม่เป็น Adjacency กัน ไม่มีเส้นเชื่อมโยงกัน

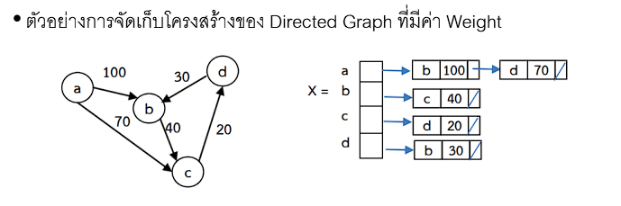
A**djacency Matrix**

•1 : เป็น Adjacency กัน มีเส้นเชื่อมโยงระหว่างกัน

**Adjacency List**: การแทนกราฟด้วย Array ของLinked Lists

•อย่างไรก็ตามบางครั้งการจัดเก็บข้อมูลกราฟขนาด n ในรูปแบบ Adjacency Matrix

ด้วย Array 2 มิติขนาด n x n

•แต่ในกรณีที่ n มีขนาดใหญ่มากๆ จะทําให้ใช้เนื้อที่หน่วยความจํามากเกินไป

•วิธีหนึ่งที่ช่วยลดการใช้งานพื้นที่หน่วยความจํา คือ การแทนกราฟด้วย Array ของLinked Lists

**Adjacency list**