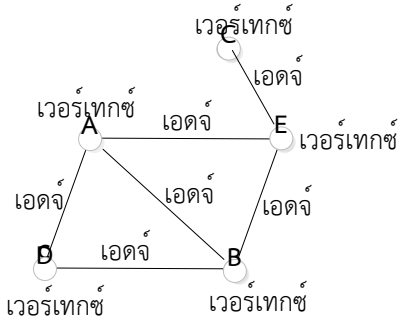
บทที่ 8
กราฟ

กราฟ (Graph) เป็นโครงสร้างข้อมูลแบบไม่ใช่เชิงเส้น (Non-Linear List) ที่สมาชิกแต่ละตัวสามารถมีสมาชิกตัวถัดไปได้มากกว่าหนึ่งตัว เช่นเดียวกับโครงสร้างข้อมูลแบบต้นไม้ (Tree) โดยที่โครงสร้างข้อมูลแบบกราฟมีความแตกต่างจากโครงสร้างข้อมูลแบบต้นไม้ คือ สมาชิกแต่ละตัวสามารถมีตัวก่อนหน้าได้มากกว่าหนึ่งตัว โครงสร้างข้อมูลแบบกราฟนำมาใช้ประโยชน์ในด้านต่าง ๆ อาทิ การสร้างแบบจำลองวงจรไฟฟ้า แผนที่ทางหลวง การวิเคราะห์วงจรไฟฟ้า และการค้นหาเส้นทางที่สั้นที่สุด (2010) การทำแผนที่ การขนส่ง วิศวกรรมไฟฟ้า เครือข่ายคอมพิวเตอร์ เส้นทางการบิน (2011) เป็นต้น ในบทนี้จะอธิบายถึงนิยามกราฟ อินดีกรีและเอาท์ดีกรี กราฟสมบูรณ์ เส้นทาง กราฟมีน้ำหนัก การแทนที่กราฟด้วยเมทริกซ์ การประกาศโครงสร้างกราฟ การดำเนินการกับกราฟ การท่องเข้าไปในกราฟ การค้นหาในแนวลึก การค้นหาในแนวกว้าง อัลกอลิทึมในการค้นหาเส้นทางที่สั้นที่สุด ต้นไม้ทอดข้าม และการแทนที่กราฟด้วยวิธีอื่น ๆ ตามรายละเอียดของเนื้อหาต่อไปนี้

นิยามกราฟ

โครงสร้างของกราฟประกอบด้วยกลุ่มของโหนดที่เรียกว่า เวอร์เทกซ์ (Vertex) และกลุ่มของเส้นที่ใช้เชื่อมโยงเวอร์เทกซ์เรียกว่า เอดจ์ (Edge) ดังภาพที่ 8.1



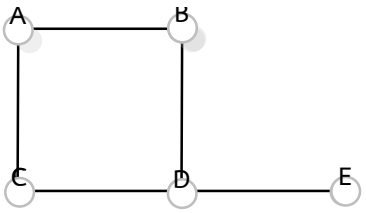
ภาพประกอบที่ 8.1 โครงสร้างกราฟ
ที่มา : ดัดแปลงจาก Mehta & Sahni. 2018 : 50

หากกำหนดให้กราฟแทนด้วยสัญลักษณ์ G เวอร์เทกซ์แทนด้วย V และเอดจ์แทนด้วย E สามารถนิยามกราฟได้ดังนี้ (สมจิตต์ ลิขิตถาวร, 2550, หน้า 305-306)

$$G = (V,E)$$

ซึ่ง V(G) คือ เซตของเวอร์เทกซ์ที่ไม่ใช่เซตว่าง (Nonempty) และมีจำนวนจำกัด (Finite)
E(G) คือ เซตของเอดจ์ เขียนด้วยคู่ของเวอร์เทกซ์ โครงสร้างข้อมูลโครงสร้างข้อม

พิจารณากราฟจากภาพประกอบที่ 8.1



จะได้ เวอร์เทกซ์ คือ A B C D E
เอดจ์ คือ เส้นที่เชื่อมโยงจาก A ไป B
เส้นที่เชื่อมโยงจาก A ไป C
เส้นที่เชื่อมโยงจาก B ไป D
เส้นที่เชื่อมโยงจาก C ไป D
เส้นที่เชื่อมโยงจาก D ไป E

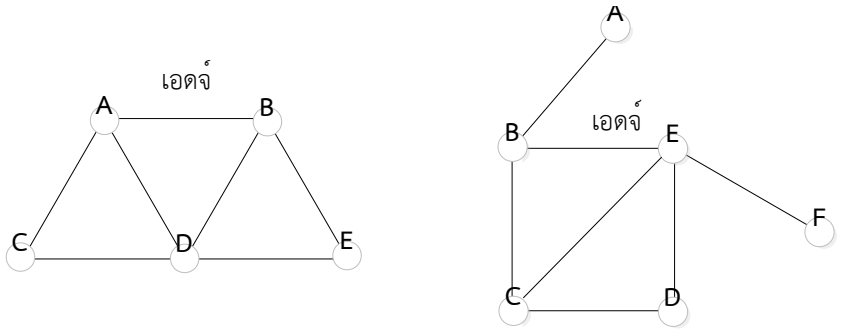
V = {A, B, C, D, E}
E = {(AB),(AC),(BD),(CD),(DE)}

ภาพประกอบที่ 8.2 แสดงการเชื่อมโยงของกราฟ
ที่มา : ดัดแปลงจาก Das. 2006 : 173

จากภาพประกอบที่ 8.2 เครื่องหมาย { } ซึ่งใช้แสดงองค์ประกอบของ V(G) และ E(G) เป็นเครื่องหมายเซตทางคณิตศาสตร์ ไม่ใช่เครื่องหมายเพื่อแสดงกลุ่มหรือบล็อก (Block) ในภาษาซี และแต่ละเอดจ์สามารถแจกแจงด้วยคู่ของเวอร์เทกซ์ที่เอดจ์นั้นเชื่อมโยง โดยมีจุลภาค (Comma) คั่นระหว่างเวอร์เทกซ์คู่นั้น (สมจิตต์ ลิขิตถาวร, 2550, หน้า 307)

โครงสร้างกราฟ แบ่งตามลักษณะการเชื่อมโยง แบ่งเป็น 2 ประเภท คือ กราฟแบบไม่มีทิศทาง (Undirected Graph) และกราฟแบบมีทิศทาง (Directed Graph) มีรายละเอียดดังนี้

1. กราฟแบบไม่มีทิศทาง เป็นกราฟที่ไม่แสดงทิศทางหรือเส้นเชื่อมโยงระหว่างเวอร์เทกซ์ไม่มีหัวลูกศร จะเรียกว่า เอดจ์ (Edge) จากเวอร์เทกซ์ต้นทางไปยังเวอร์เทกซ์ปลายทางและเกิดการผันกลับของเวอร์เทกซ์ปลายทางไปยังเวอร์เทกซ์ต้นทางได้ ดังภาพที่ 8.3



(ก) (ข)

ภาพประกอบที่ 8.3 กราฟแบบไม่มีทิศทางแบบต่าง ๆ
ที่มา : ดัดแปลงจาก McMilan. 2014 : 140 และGilberg & Forouzan. 2005 : 482

จากภาพที่ 8.2 (ก) จะได้ เวอร์เทกซ์ คือ {A B C D E} มีเอดจ์ทั้งหมด 7 ชุด คือ เส้นที่เชื่อมโยงจาก A ไป B เส้นที่เชื่อมโยงจาก A ไป C เส้นที่เชื่อมโยงจาก A ไป D เส้นที่เชื่อมโยงจาก B ไป D เส้นที่เชื่อมโยงจาก B ไป E เส้นที่เชื่อมโยงจาก C ไป D เส้นที่เชื่อมโยงจาก D ไป E ดังนี้

V = {A, B, C, D, E}

E = {(A, B), (A, C), (A, D), (B, D), (B, E), (C, D), (D, E)}

หรือ E = {(B, A), (C, A), (D, A), (D, B), (E, B), (D, C), (E, D)}

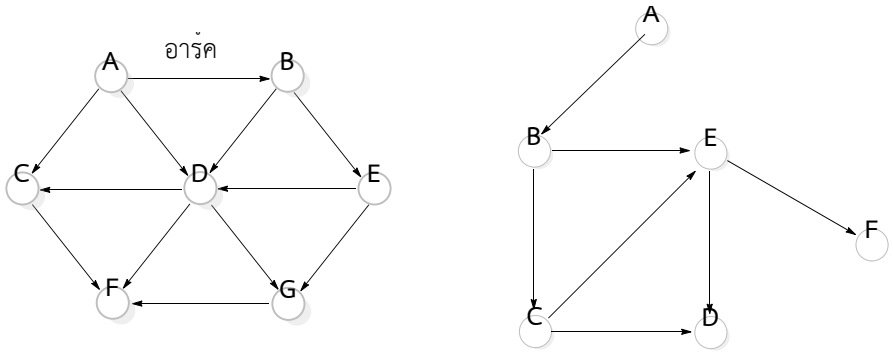
ภาพ (ข) จะได้ เวอร์เทกซ์ คือ {A B C D E F} มีเอดจ์ทั้งหมด 5 ชุด คือ เส้นที่เชื่อมโยงจาก A ไป B เส้นที่เชื่อมโยงจาก A ไป C เส้นที่เชื่อมโยงจาก B ไป D เส้นที่เชื่อมโยงจาก B ไป E เส้นที่เชื่อมโยงจาก C ไป F เขียนได้ดังนี้

V = {A, B, C, D, E, F}

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

หรือ E = {(B,A), (C, A), (D, B), (E, B), (F, C)}

2. กราฟแบบมีทิศทางหรือไดกราฟ เป็นกราฟที่แสดงทิศทางหรือเส้นเชื่อมโยงระหว่างเวอร์เทกซ์มีหัวลูกศร เรียกว่า อาร์ค (Arcs) จากเวอร์เทกซ์ต้นทางไปยังเวอร์เทกซ์ปลายทางเท่านั้น ดังภาพที่ 8.4



(ก) (ข)

ภาพประกอบที่ 8.4 กราฟแบบมีทิศทางแบบต่าง ๆ
ที่มา : ดัดแปลงจาก Weiss M' A'. 2014 : 380 และ Gilberg & Forouzan. 2005 : 482

จากภาพที่ 8.4 ภาพ (ก) จะได้ เวอร์เทกซ์ คือ {A B C D E F G} มีเอดจ์ทั้งหมด 12 ชุด คือ เส้นที่เชื่อมโยงจาก A ไป B เส้นที่เชื่อมโยงจาก A ไป C เส้นที่เชื่อมโยงจาก A ไป D เส้นที่เชื่อมโยงจาก B ไป D เส้นที่เชื่อมโยงจาก B ไป E เส้นที่เชื่อมโยงจาก C ไป F เส้นที่เชื่อมโยงจาก D ไป C เส้นที่เชื่อมโยงจาก D ไป F เส้นที่เชื่อมโยงจาก D ไป G เส้นที่เชื่อมโยงจาก E ไป D เส้นที่เชื่อมโยงจาก E ไป G และเส้นที่เชื่อมโยงจาก G ไป F เขียนได้ดังนี้

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

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

ภาพ (ข) จะได้ เวอร์เทกซ์ คือ {A B C D E} มีเอดจ์ทั้งหมด 6 ชุด คือ เส้นที่เชื่อมโยงจาก A ไป B เส้นที่เชื่อมโยงจาก A ไป D เส้นที่เชื่อมโยงจาก B ไป E เส้นที่เชื่อมโยงจาก C ไป A เส้นที่เชื่อมโยงจาก D ไป C และเส้นที่เชื่อมโยงจาก D ไป E เขียนได้ดังนี้

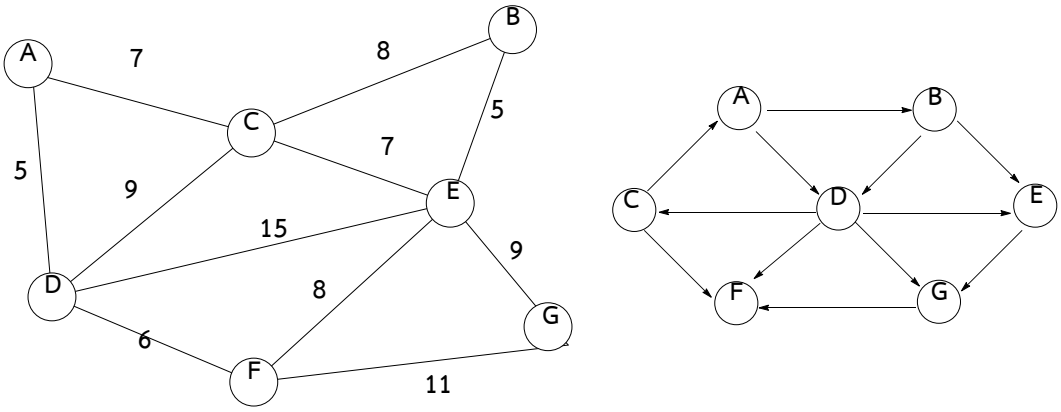
V = {A, B, C, D, E}

E = {(A, B), (A, D), (B, E), (C, A), (D, C), (D, E)}

สรุปได้ว่า ลูกศรในไดกราฟมีความสำคัญเพราะจะบ่งบอกถึงทิศทางความสัมพันธ์ของเวอร์เทกซ์ เวอร์เทกซ์ที่มีเอดจ์เชื่อมต่อกัน เรียกว่า เวอร์เทกซ์ข้างเคียง (Adjacent) หรือ เวอร์เทกซ์ประชิด (สมจิตต์ ลิขิตถาวร, 2550, หน้า 309) เช่น ภาพที่ 8.4 (ก) เวอร์เทกซ์ B เป็นเวอร์เทกซ์ข้างเคียงกับ A แต่เวอร์เทกซ์ A ไม่เป็นเวอร์เทกซ์ข้างเคียงกับ B ส่วนกราฟแบบไม่มีทิศทาง เช่น ภาพที่ 8.3 (ก) เวอร์เทกซ์ A เป็นเวอร์เทกซ์ข้างเคียงกับ B และเวอร์เทกซ์ B เป็นเวอร์เทกซ์ข้างเคียงกับ A เป็นต้น สำหรับเส้นทาง (Path) คือ ลำดับของเวอร์เทกซ์แต่ละเวอร์เทกซ์ประชิดต่อกันไปยังตัวถัดไป ทั้งกราฟแบบมีทิศทางและไม่มีทิศทาง ต่างก็มีเส้นทางด้วยกันทั้งสิ้น เพียงแต่กราฟแบบไม่มีทิศทางเส้นทางสามารถเดินทางไปมาระหว่างกันได้ (โอภาส เอี่ยมสิริวงศ์, 2559, หน้า 298) จากภาพที่ 8.4 (ก) มีหลายเส้นทาง เช่น {A, B, E, D, C} {A, B, E, G, F} {A, D, G, F} {A, C, F} เป็นต้น

ถ้าแบ่งโครงสร้างกราฟตามน้ำหนักแบ่งออกเป็น 2 ประเภท คือ กราฟที่มีน้ำหนัก (Weighted Graphs) และกราฟที่ไม่มีน้ำหนัก (Unweighted Graphs) มีรายละเอียดดังนี้ (กนกมน รุจิรกุล, 2560, หน้า 130)

1. กราฟแบบมีน้ำหนัก เป็นกราฟที่เส้นเชื่อมแสดงถึงความสัมพันธ์ของ 2 โหนด อาจเป็นกราฟแบบมีทิศทางหรือไม่มีทิศทาง โดยเส้นเชื่อมแสดงความหมายอย่างใดอย่างหนึ่ง เช่น ระยะทาง ความเร็ว เวลาเดินทาง ค่าโดยสาร ฯลฯ ดังภาพที่ 8.5 (ก)



(ก) (ข)

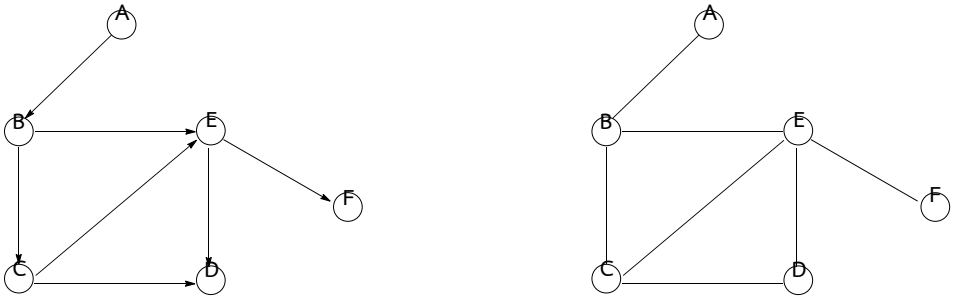
ภาพประกอบที่ 8.5 (ก) กราฟแบบมีน้ำหนัก (ข) กราฟแบบไม่มีน้ำหนัก

ที่มา : ดัดแปลงจาก Karumanchi. 2017 : 418 และ Weiss M' A'. 2014 : 388

2. กราฟที่ไม่มีน้ำหนัก เป็นกราฟที่เส้นเชื่อมแสดงถึงความสัมพันธ์ของ 2 โหนด อาจเป็นกราฟแบบมีทิศทางหรือไม่มีทิศทาง โดยเส้นเชื่อมแสดงความหมายอย่างใดอย่างหนึ่ง แต่ไม่ระบุข้อมูลหรือค่าต่าง ๆ ดังภาพที่ 8.5 (ข)

คำศัพท์ที่เกี่ยวกับกราฟ

1. เวอร์เทกซ์ข้างเคียง (Adjacent Vertex) หรือ เวอร์เทกซ์ประชิด คือ เวอร์เทกซ์ที่มีเอดจ์เชื่อมต่อกัน ดังภาพประกอบที่ 8.6



(ก) กราฟแบบมีทิศทาง (ข) กราฟแบบไม่มีทิศทาง

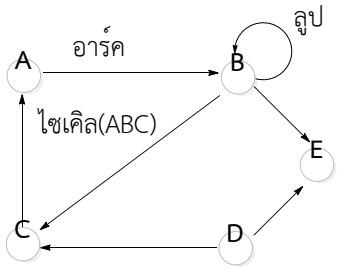
ภาพประกอบที่ 8.6 กราฟแบบมีทิศทางและแบบไม่มีทิศทาง
ที่มา : ดัดแปลงจาก Gilberg & Forouzan. 2005 : 482

จากภาพที่ 8.6 (ก) เวอร์เทกซ์ B เป็นเวอร์เทกซ์ข้างเคียงกับ A แต่เวอร์เทกซ์ A ไม่เป็นเวอร์เทกซ์ข้างเคียงกับ B โดยที่เวอร์เทกซ์ E ไม่เป็นเวอร์เทกซ์ข้างเคียงกับ D แต่เวอร์เทกซ์ D เป็นเวอร์เทกซ์ข้างเคียงกับ E ส่วนกราฟแบบไม่มีทิศทาง เช่น ภาพที่ 8.6 (ข) เวอร์เทกซ์ E เป็นเวอร์เทกซ์ข้างเคียงกับ D และเวอร์เทกซ์ D เป็นเวอร์เทกซ์ข้างเคียงกับ E สำหรับเวอร์เทกซ์ D ไม่เป็นเวอร์เทกซ์ข้างเคียงกับ F เป็นต้น

2. เส้นทาง (Path) คือ ลำดับของเวอร์เทกซ์แต่ละเวอร์เทกซ์ประชิดต่อกันไปยังตัวถัดไป ทั้งกราฟแบบมีทิศทางและไม่มีทิศทาง ต่างก็มีเส้นทางด้วยกันทั้งสิ้น เพียงแต่กราฟแบบไม่มีทิศทางเส้นทางสามารถเดินทางไปมาระหว่างกันได้ (โอภาส เอี่ยมสิริวงศ์, 2559, หน้า 298) จากภาพที่ 8.6 (ก) มีหลายเส้นทาง เช่น {A, B, C, D} {A, B, C, E, F} {A, B, E, D} {A, B, E, F} ภาพที่ 8.6 (ข) มีหลายเส้นทาง เช่น {A, B, C, D, E, F} {A, B, E, F} {A, B, E, D} {A, B, C, E, D} เป็นต้น

3. ดีกรี (Degree) คือ จำนวนเส้นเอดจ์หรืออาร์คที่เชื่อมกับเวอร์เทกซ์ ดีกรีสามารถแบ่งย่อยออกเป็นเอาต์ดีกรี (Outdegree) และอินดีกรี (Indegree) โดยเอาต์ดีกรีของเวอร์เทกซ์ในไดกราฟ คือ จำนวนอาร์คที่ออกจากเวอร์เทกซ์ ส่วนอินดีกรีคือจำนวนอาร์คที่เข้ามายังเวอร์เทกซ์ (โอภาส เอี่ยมสิริวงศ์, 2559, หน้า 299) จากภาพที่ 8.6 (ก) อินดีกรีของเวอร์เทกซ์ A เท่ากับ 0 เอาต์ดีกรีเท่ากับ 3 อินดีกรีของเวอร์เทกซ์ B เท่ากับ 1 เอาต์ดีกรีเท่ากับ 2 อินดีกรีของเวอร์เทกซ์ C เท่ากับ 2 เอาต์ดีกรีเท่ากับ 1 อินดีกรีของเวอร์เทกซ์ E เท่ากับ 1 เอาต์ดีกรีเท่ากับ 2 อินดีกรีของเวอร์เทกซ์ D เท่ากับ 3 เอาต์ดีกรีเท่ากับ 3 อินดีกรีของเวอร์เทกซ์ F เท่ากับ 3 เอาต์ดีกรีเท่ากับ 0 อินดีกรีของเวอร์เทกซ์ G เท่ากับ 2 เอาต์ดีกรีเท่ากับ 1

4. ไซเคิล (Cycle) คือ วงจรที่มีเส้นทางเชื่อมอย่างน้อย 3 เวอร์เทกซ์ ที่มีจุดเริ่มต้นและจุดสิ้นสุดอยู่บนเวอร์เทกซ์เดียวกัน ดังภาพประกอบที่ 8.7



ภาพประกอบที่ 8.7 ไซเคิล A, B, C
ที่มา : ดัดแปลงจาก Malik, D.S. 2010 : 688

จากภาพที่ 8.7 เป็นกราฟแบบมีทิศทางที่มีไซเคิล (ABC) จากเวอร์เทกซ์ A ไปเวอร์เทกซ์ B และ เวอร์เทกซ์ C สิ้นสุดที่เวอร์เทกซ์ A ส่วนภาพที่ 8.5 (ข) เป็นกราฟแบบไม่ทิศทางมี (B, C, E, B) เป็นไซเคิล

5. ลูป (Loop) คือ เวอร์เทกซ์ที่มีอาร์คเดียวที่มีจุดเริ่มต้นและจุดสิ้นสุดบนเวอร์เทกซ์เดียวกัน ดังภาพประกอบที่ 8.7

เอกสารอ้างอิง

Das, V. V. (2006). *Principles of Data Structures Using C and C++.* New Dehli: New Age International (P) Ltd. .

Drozdek, A. (2013). *Data Structures and Algorithms in C++* (4th ed.). Boston, MA: Cengage Learning.

Gilberg, R. F., & Forouzan, B. A. (2005). *Data Structures: A Pseudocode Approach with C* (2nd ed.). Boston: Thomson.

Goodrich, M. T., Tamassia, R., & Mount, D. M. (2011). *Data Structures and Algorithms in C++* (2nd ed.). Hoboken, NJ: Wiley.

Goodrich., M. T., & Tamassia, R. (2015). *Algorithm Design and Applications.* Hoboken,NJ: John Wiley & Sons, Inc.

Karumanchi, N. (2017). *Data structures and algorithms made easy in Java : data structures and algorithmic puzzles.* India: CareerMonk.

Malik, D. (2010). *Data structures using C++* (2nd ed.). Boston, MA: Course Technology/Cengage Learning.

McMilan, M. (2014). *Data structures and algorithms with JavaScript* (1st ed.). Sebastopol, CA: O'Reilly Media.

McMillan, M. (2007). *Data structures and algorithms using C#.* Cambridge ; New York: Cambridge University Press.

Robertson, L. A. (2007). *Simple program design : a step by step approach* (5th ed.). Australia: Thomson.

Shaffer, C. A. (2011). *Data Structures and Algorithm Analysis* (3rd ed.). Blacksburg, VA: http://people.cs.vt.edu/~shaffer/Book/.

Srivastava, S. a. (2004). *Data Structures Through C In Depth.* New Dehli: BPB Publications.

Weiss, M. A. (2014). *Data structures and algorithm analysis in C++* (4th ed.). Boston: Pearson.

Weiss, M. A. (2014). *Data Structures and Algorithm Analysis in C++* (4th ed.). Upper Saddle River, NJ: Pearson Education.

โอภาส เอี่ยมสิริวงศ์. (2559). *โครงสร้างข้อมูล (Data Structure) เพื่อการออกแบบโปรแกรมคอมพิวเตอร์ (ฉบับปรับปรุงเพิ่มเติม).* กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

กนกมน รุจิรกุล. (2560). *โครงสร้างข้อมูล (Data Structure).* นครราชสีมา: โคราชมาร์เก็ตติ้ง แอนด์ โปรดักชั่น.

วิษณุ ช้างเนียม. (2562). *คู่มือเรียนโครงสร้างข้อมูลและอัลกอลิทึม (Data Structure and Algorithm) ฉบับสมบูรณ์* (2 ed.). นนทบุรี: ไอดีซี พรีเมียร์.

สมจิตต์ ลิขิตถาวร. (2550). *โครงสร้างข้อมูลและการวิเคราะห์อัลกอลิทึมด้วยซี (Data Structures and Algorithm Analysis in C* (4 ed.). กรุงเทพฯ: ธนาเพรส.