

หนังสือสรุปความรู้: ทฤษฎีกราฟและอัลกอริทึมเพื่อโครงการเครือข่าย

เอกสารนี้จะสรุปความรู้ที่จำเป็นทั้งหมดสำหรับโครงการ

"การสร้างแบบจำลองเครือข่ายเพื่อเพิ่มประสิทธิภาพการส่งต่อผู้ป่วย"

โดยเน้นที่ความเข้าใจง่ายและสามารถนำไปประยุกต์ใช้ได้จริง

ส่วนที่ 1: ความรู้พื้นฐานเรื่อง "ทฤษฎีกราฟ" (Graph Theory)

ก่อนอื่นเราต้องเข้าใจก่อนว่า "เครือข่าย" ที่เรากำลังจะสร้างนั้น ในทางคณิตศาสตร์เรียกว่า **กราฟ (Graph)** ซึ่งไม่ได้หมายถึงกราฟแท่งหรือกราฟเส้นที่เราคุ้นเคย

แต่หมายถึงแบบจำลองที่ประกอบด้วย 2 สิ่งหลักๆ คือ

- จุดยอด (Vertex หรือ Node):** คือสิ่งของหรือสถานที่ในเครือข่ายของเรา
 - ในโครงการนี้: โรงพยาบาล แต่ละแห่งคือ "จุดยอด"
- เส้นเชื่อม (Edge):** คือเส้นทางที่เชื่อมต่อระหว่างจุดยอด
 - ในโครงการนี้: ถนน ที่เชื่อมระหว่างโรงพยาบาลคือ "เส้นเชื่อม"

และสิ่งที่สำคัญมากอีกอย่างคือ **ค่าน้ำหนัก (Weight)** ซึ่งเป็น "ค่า" ที่กำกับอยู่บนเส้นเชื่อมแต่ละเส้น เพื่อบอกคุณสมบัติของเส้นทางนั้นๆ

- ในโครงการนี้: เวลาที่ใช้ในการเดินทาง (นาที) หรือ ระยะทาง (กม.) คือ "ค่าน้ำหนัก"

ตัวอย่างที่ 1: การแปลงปัญหาให้เป็นกราฟ

สมมติในพื้นที่ศึกษามีโรงพยาบาล 4 แห่ง (A, B, C, D) และมีเวลาเดินทางระหว่างกัน (หน่วยเป็นนาที) ดังนี้

- A ไป B: 15 นาที
- A ไป C: 20 นาที
- B ไป C: 5 นาที
- B ไป D: 30 นาที
- C ไป D: 10 นาที

1. วาดเป็นกราฟ:

เราสามารถวาดภาพแทนสถานการณ์นี้ได้ดังนี้

- จุดวงกลม 4 จุด แทน รพ. A, B, C, D
- ลากเส้นเชื่อมระหว่าง รพ. ที่มีถนนถึงกัน
- เขียนตัวเลข (ค่าน้ำหนัก) กำกับบนเส้นเชื่อม

2. ตารางเมทริกซ์ (Adjacency Matrix):

ในการนำข้อมูลไปใช้ในคอมพิวเตอร์ เรามักเก็บในรูปแบบตาราง โดยแถวและคอลัมน์คือโรงพยาบาล และค่าในตารางคือเวลาเดินทาง ถ้าไม่มีเส้นทางตรง จะใส่ค่าเป็นอนันต์ (∞) หรือค่าที่สูงมากๆ

จาก \ ไป	A	B	C	D
A	0	15	20	∞
B	15	0	5	30
C	20	5	0	10
D	∞	30	10	0

ส่วนที่ 2: อัลกอริทึมของไดคส์ตรา (Dijkstra's Algorithm)

นี่คือเครื่องมือพระเอกของโครงงานเรา! หน้าที่ของมันคือ "การหาเส้นทางที่สั้นที่สุด (ใช้เวลาน้อยที่สุด) จากจุดเริ่มต้นหนึ่งจุด ไปยังจุดยอดอื่นๆ ทั้งหมดในเครือข่าย"

หลักการทำงานแบบง่ายๆ:

อัลกอริทึมจะทำงานเหมือนเราวางแผนที่แล้วค่อยๆ สำรวจไปที่ละนิด โดยยึดหลักว่า "จะไปทางที่ใกล้ที่สุดก่อนเสมอ"

- กำหนดจุดเริ่มต้น:** เลือกโรงพยาบาลที่เราจะเริ่มส่งผู้ป่วย
- ตั้งค่าเริ่มต้น:**
 - ระยะทางจากจุดเริ่มต้นไปหาตัวเอง = 0
 - ระยะทางจากจุดเริ่มต้นไปหาจุดอื่นๆ ทั้งหมด = ∞ (เพราะยังไม่มีรู้เส้นทาง)
 - สร้างลิสต์ของจุดที่ "ยังไม่ได้ไป" (Unvisited)
- เริ่มสำรวจ:**
 - ดึงจุดที่ "ใกล้ที่สุด" และ "ยังไม่ได้ไป" ออกมา (รอบแรกก็คือจุดเริ่มต้นนั่นเอง)
 - มองไปยังจุดเพื่อนบ้าน (Neighbor) ของมัน แล้วคำนวณว่า "ถ้าเดินทางผ่านจุดที่เราอยู่ตอนนี้ไปยังเพื่อนบ้าน จะใช้เวลาน้อยกว่าเดิมที่เคยบันทึกไว้ไหม?"
 - ถ้าใช่ ให้อัปเดตระยะทางที่สั้นที่สุดของเพื่อนบ้านนั้น
- ทำซ้ำ:** ทำขั้นตอนที่ 3 ไปเรื่อยๆ จนกว่าจะสำรวจครบทุกจุดที่ต้องการ

ตัวอย่างที่ 2: การใช้ Dijkstra's Algorithm

จากกราฟในตัวอย่างที่ 1 เราต้องการหาเส้นทางที่ใช้เวลาน้อยที่สุดจาก รพ. A ไปยังโรงพยาบาลอื่นๆ ทั้งหมด

ตารางติดตามการทำงาน:

ขั้นตอน	จุดที่กำลังพิจารณา	ระยะทางไป A	ระยะทางไป B	ระยะทางไป C	ระยะทางไป D	จุดที่ยังไม่ได้ไป
เริ่มต้น	-	0	∞	∞	∞	{A, B, C, D}
1	A	0	15 (ผ่าน A)	20 (ผ่าน A)	∞	{B, C, D}
2	B (ใกล้สุด)	0	15	20 (ไม่เปลี่ยน)	45 (15+30)	{C, D}
3	C (ใกล้สุด)	0	15	20	30 (20+10)	{D}
4	D (สุดท้าย)	0	15	20	30	{}

สรุปผลลัพธ์จากตาราง:

- เส้นทางที่เร็วที่สุดจาก A ไป B คือ: $A \rightarrow B$ (15 นาที)
- เส้นทางที่เร็วที่สุดจาก A ไป C คือ: $A \rightarrow C$ (20 นาที)
- เส้นทางที่เร็วที่สุดจาก A ไป D คือ: $A \rightarrow C \rightarrow D$ (30 นาที) <--
สังเกตว่าอัลกอริทึมไม่เลือกทางตรง $A \rightarrow B \rightarrow D$ เพราะใช้เวลาถึง 45 นาที!

ส่วนที่ 3: แบบฝึกหัดเพื่อทบทวน

ลองใช้ความรู้ทั้งหมดแก้ปัญหาต่อไปนี้

สถานการณ์: จังหวัดสมมติแห่งหนึ่งมีโรงพยาบาล 5 แห่ง (P, Q, R, S, T) และมีเวลาเดินทางระหว่างกันดังนี้

- P - Q: 10 นาที
- P - R: 50 นาที
- Q - R: 20 นาที
- Q - S: 40 นาที
- R - S: 15 นาที
- R - T: 25 นาที
- S - T: 5 นาที

โจทย์ข้อที่ 1:

1.1) จงวาดกราฟแทนเครือข่ายโรงพยาบาลนี้ พร้อมระบุค่าน้ำหนักบนเส้นเชื่อม

1.2) จงสร้างตารางเมทริกซ์ (Adjacency Matrix) ของเครือข่ายนี้

โจทย์ข้อที่ 2:

ผู้ป่วยอยู่ที่โรงพยาบาล P และต้องการส่งต่อไปยังโรงพยาบาล T ซึ่งเป็นโรงพยาบาลศูนย์ที่มีศักยภาพสูงสุด

2.1) จงใช้อัลกอริทึมของไดคัสตรา เพื่อหาเส้นทางที่ใช้เวลาน้อยที่สุดจาก P ไปยังโรงพยาบาลอื่นๆ ทั้งหมด (แสดงตารางติดตามการทำงาน)

2.2) เส้นทางที่เร็วที่สุดจาก P ไป T คือเส้นทางใด และใช้เวลาทั้งหมดกี่นาที?

ส่วนที่ 4: ก้าวต่อไปสู่การเขียนโค้ด

เมื่อเข้าใจหลักการทั้งหมดนี้แล้ว การนำไปเขียนโค้ดในภาษา Python จะไม่ยากเกินไป เพราะมีไลบรารีที่ช่วยเราได้มาก

- **Pandas:** ใช้สำหรับโหลดและจัดการข้อมูลจากไฟล์ Excel ที่เราเก็บข้อมูลโรงพยาบาลและเส้นทาง
- **NetworkX:** คือไลบรารีสำหรับทฤษฎีกราฟโดยเฉพาะ เราสามารถสร้างกราฟจากข้อมูลของเราได้ง่ายๆ และที่สำคัญคือ มีฟังก์ชันสำหรับรัน **Dijkstra's Algorithm ให้เราเลย!** เราแค่ต้องรู้วิธีเรียกใช้และอ่านผลลัพธ์เท่านั้น

การเข้าใจหลักการทำงานเบื้องหลังจะช่วยให้เราสามารถใช้เครื่องมือเหล่านี้ได้อย่างถูกต้องและสามารถตรวจสอบแก้ไขได้เมื่อเกิดข้อผิดพลาด