มหาวิทยาลัยพระจอมเกล้าธนบุรี

รายงานโครงงาน

รายวิชา CSS121

จัดทำโดย

นาย ธัญพิสิษฐ์ บัวประคอง 64090500404

นางสาว ปุณณภา เทียนชัย 64090500405

นาย วรินทร สิทธิสินธุ์ 64090500407

นาย ภูมิพัฒน์ กรเจริญพิสุทธิ์ 64090500443

ภาคเรียนที่ 1

ประจำปีการศึกษา 2564

คำนำ

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชา CSC121 เพื่อให้ได้ศึกษาหาความรู้และได้ศึกษาอย่างเข้าใจเพื่อเป็นประโยชน์กับการศึกษา

ผู้จัดทำหวังว่ารายงานเล่มนี้เป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษา ที่กำลังหาข้อมูลเรื่องนี้อยู่ หากมีข้อผิดพลาดประการใด ผู้จัดทำขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

25/11/2564

CSS121 Project Assignment (12%)

กลุ่มละ 3 คน

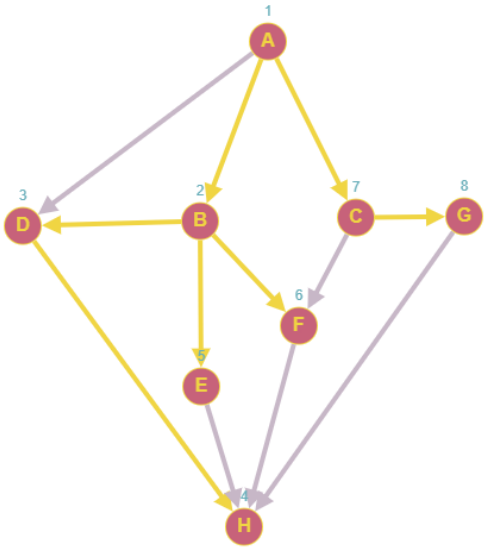
จงเขียนโปรแกรมในการรับข้อมูลกราฟขนาด N ใดๆ แล้ว

1. หาเส้นทางที่เป็นไปได้ทั้งหมด เมื่อกำหนดจุดยอดต้นทางและจุดยอดปลายทาง โดยวิธี Depth-first search (DFS) หรือ Breadth-first search (BFS)
2. หาเส้นทางที่สั้นที่สุด จากจุดยอดต้นทางที่กำหนดไปยังจุดยอดปลายทางที่กำหนด โดยวิธี Djikstra's Shortest Path
3. **Extra credit** หาต้นไม้แผ่ทั่วที่มีน้ำหนักต่ำสุดโดยวิธี Prim's algorithm หรือ Kruskal's algorithm

4. วิธีส่ง

   Code Python ที่สามารถ run ได้ ไม่มี bugs หรือ errors **พร้อมรายงาน** กำหนดส่ง เลือก วันที่ 26 พ.ย. 2564 หรือ วันที่ 10 ธ.ค. 2564 โดยเป็นการมา

1.Depth-first search (DFS) Or Breadth-first search (BFS)



"การค้นหาตามแนวลึก" วิธีการคือพุ่งตรงดิ่งเข้าไปก่อนเลย สุดทางแล้วค่อยถอยกลับมาแล้วหาทางใหม่ลองไปเรื่อย ๆ

หลักการ กำหนดกราฟ V = {v1,v2,… …, vn}

1. เลือก vi เป็น root มา 1 จุด
2. จาก vi หาจุดยอดที่อยู่ระดับต่อไป (1 จุด)
3. ทำซ้ำขั้นตอนที่ 2 ไปเรื่อย ๆจนถึงใบ
4. ให้ ย้อนรอย(backtracking) ถอยกลับมาทีละระดับเมื่อพบจุดยอด(ที่ยังไม่เลือก)ใดก็ดูว่าจะเชื่อมต่อระดับได้หรือไม่ (ต้องไม่เกิดวงจร)

ก) ถ้ามี ให้เพิ่มจุดยอดนั้นและทำซ้ำขั้นตอนที่ 2-4

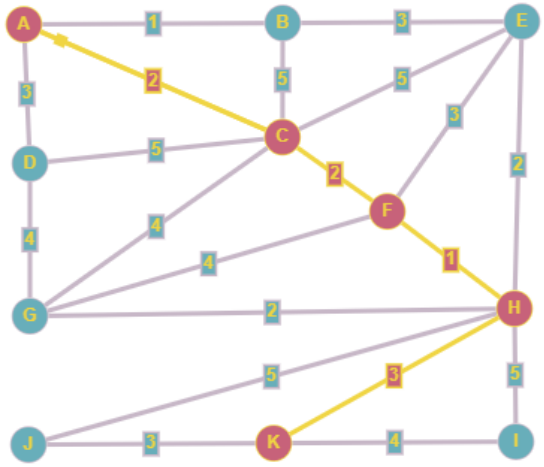
ข) ถ้าไม่มี ให้ย้อนรอยต่อไป จนกว่า

ข.1) ถึงจุดรากแล้ว

หรือ ข.2) ครบ n จุดยอดแล้ว

หรือ ข.3) ครบ (n –1) เส้นเชื่อมแล้ว

2. Dijkstra’s Shortest Path



กำหนดให้ปมหนึ่งเป็นปมเริ่มต้น (initial node) และกำหนดให้ "ระยะทางของปม Y" (distance of node Y) หมายถึงระยะทางจากปมเริ่มต้นไปยังปม Y ขั้นตอนวิธีของไดก์สตราจะกำหนดค่าระยะทางเริ่มต้นไว้บางปมและจะเพิ่มค่าไปทีละขั้นตอน

1.กำหนดให้ทุกปมมีค่าระยะทางตามเส้นเชื่อม โดยให้ปมเริ่มต้นมีค่าเป็นศูนย์ และปมอื่นมีค่าเป็นอนันต์

2.ทำเครื่องหมายทุกปมยกเว้นปมเริ่มต้นว่ายังไม่ไปเยือน (unvisited) ตั้งให้ปมเริ่มต้นเป็นปมปัจจุบัน สร้างเซตของปมที่ยังไม่ไปเยือนขึ้นมาเซตหนึ่งซึ่งประกอบด้วยทุกปมยกเว้นปมเริ่มต้น

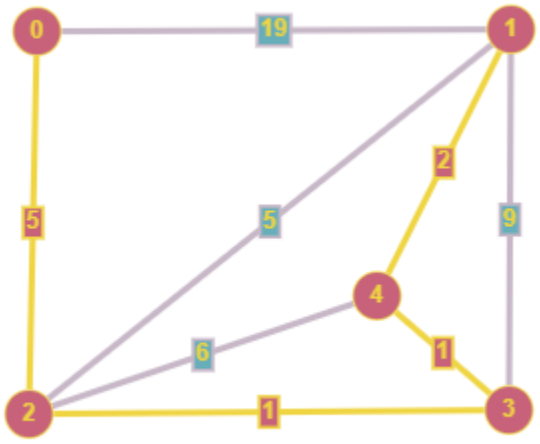
3.จากปมปัจจุบัน พิจารณาปมข้างเคียงตามเส้นเชื่อมทุกปมที่ยังไม่ไปเยือน และคำนวณระยะทางต่อเนื่องของเส้นเชื่อม ตัวอย่างเช่น ถ้าปมปัจจุบันคือ A มีระยะทางของปมเป็น 6 และเส้นเชื่อมที่ต่อจาก A ไปยังปมข้างเคียง B มีระยะทางเป็น 2 ดังนั้นระยะทางของปม B (โดยผ่าน A) จึงเท่ากับ 6+2=8 เป็นต้น ถ้าระยะทางที่คำนวณได้มีค่าน้อยกว่าค่าระยะทางที่บันทึกอยู่ของปมนั้น ให้เขียนทับค่าระยะทางของปมดังกล่าว แม้ว่าปมข้างเคียงได้ถูกพิจารณาแล้ว แต่ก็ยังไม่ทำเครื่องหมายว่าไปเยือนแล้ว (visited) ในขั้นตอนนี้ ปมข้างเคียงจะยังคงอยู่ในเซตของปมที่ยังไม่ไปเยือนเช่นเดิม

4.เมื่อพิจารณาปมข้างเคียงจากปมปัจจุบันครบทุกปมแล้ว ทำเครื่องหมายปมปัจจุบันว่าไปเยือนแล้ว และนำออกจากเซตของปมที่ยังไม่ไปเยือน ปมที่ไปเยือนแล้วนี้จะไม่ถูกนำมาตรวจสอบอีก ค่าระยะทางที่บันทึกอยู่จะสิ้นสุดและมีค่าน้อยสุด

5.ปมปัจจุบันตัวถัดไปที่ถูกเลือกจะเป็นปมที่มีค่าระยะทางน้อยสุดในเซตของปมที่ยังไม่ไปเยือน

6.ถ้าเซตของปมที่ยังไม่ไปเยือนฟว่างแล้วให้หยุดการทำงาน ขั้นตอนวิธีเสร็จสิ้น หากไม่ใช่ให้เลือกปมยังไม่ไปเยือนที่มีค่าระยะทางน้อยสุดเป็นปมปัจจุบัน แล้ววนกลับไปทำขั้นตอนที่ 3

3.Prim’s algorithm Or Kruskal’s algorithm



ให้ T เริ่มจากต้นไม้ว่าง

1. เลือกจุดราก v i ใดก็ได้

2. หาจุดยอดต่อไปเพิ่มใน T โดยดูเส้นเชื่อมทั้งหมดใน G ซึ่งเชื่อมต่อกับจุดยอดใน T และไม่เกิดวงจร – เลือกด้านน้ำหนักน้อยสุด – ถ้ามีหลายด้านมีน้ำหนักน้อยเท่ากัน อักษรแรกต่างกัน เช่น AB = 3 และ CD = 3 เลือกด้าน AB เพราะ A มาก่อน C อักษรแรกเหมือนกัน เช่น AB = 3 และ AC = 3 เลือกด้าน AB เพราะ B มาก่อน C

3. ทำซ้ำขั้นตอน 2. จนกว่าได้จำนวนจุดยอด =n จุด หรือ จำนวนเส้นเชื่อม = (n –1) เส้น