# วิธีการดำเนินโครงงาน

วิธีการดำเนินโครงงานโปรแกรมแสดงผลข้อมูลประเภทกราฟออกมาในรูปแบบกราฟิกมีขั้นตอนต่างๆ ดังนี้คือ

1. ศึกษาปัญหาของการแสดงผลข้อมูลประเภทกราฟ
2. วิเคราะห์และออกแบบการทำงานของโปรแกรม
3. สร้างส่วนแสดงผลกราฟ
4. สร้างส่วนดำเนินการกับไฟล์
5. สร้างส่วนขั้นตอนวิธี

ซึ่งรายละเอียดในแต่ล่ะขั้นตอนจะอธิบายแยกออกเป็นข้อๆ ต่อไปนี้

## ศึกษาปัญหาของการแสดงผลข้อมูลประเภทกราฟ

การแสดงผลข้อมูลประเภทกราฟออกมาในรูปแบบกราฟิกนั้น จำเป็นต้องมีการใช้เครื่องมือในการสร้างส่วนกราฟิกขึ้นมา โดยจะต้องมีการแสดงส่วนประกอบต่างๆ ของกราฟได้ ทั้งในส่วนของเส้นเชื่อม และจุดตัด การแสดงผลเพิ่มเติมอื่นๆ เช่น ชื่อของแต่ละจุดตัด และน้ำหนักบนเส้นเชื่อม ซึ่งในการพัฒนาครั้งนี้ส่วนที่จะแสดงผลออกมาในรูปแบบกราฟิก มีดังนี้

1. การแสดงผลเส้นเชื่อมและจุดตัด โดยที่จะมีการวาดรูปจุดตัดและระบายสีมี มีการวาดเส้นเชื่อมโยงระหว่างจุดตัด
2. การเลื่อนจุดตัดและเส้นเชื่อม การแสดงผลข้อมูลออกมาอาจจะทำให้เกิดเส้นเชื่อมที่ไขว้กันได้ การแก้ปัญหา คือการที่ทำให้จุดตัดสามารถเปลี่ยนที่ได้ การเลื่อนที่ง่ายต่อผู้ใช้แบบหนึ่งคือ การให้ผู้ใช้เมาส์เลื่อนเอง
3. การแสดงผลชื่อของจุดตัดและน้ำหนักบนเส้นเชื่อม ซึ่งการแสดงผลน้ำหนัก บนเส้นเชื่อมจะต้องมีการเปลี่ยนตำแหน่งตามการเคลื่อนที่ของเส้นเชื่อมตลอดเวลา โดยการวางจะต้องวางตำแหน่งของน้ำหนัก ไว้ที่ตรงกลางเส้นเชื่อมเสมอเพื่อให้ดูง่าย
4. การแสดงผลอาจจะมีขีดจำกัดการมองเห็นอยู่ที่ ขนาดของกราฟและพื้นที่แสดงผล ซึ่งปัญหานี้แก้ได้โดยการทำการย่อขยายกราฟ เมื่อกราฟมีขนาดใหญ่ก็สามารถที่จะย่อกราฟให้เล็กลงเพื่อดูกราฟทั้งหมดได้
5. การบันทึกพิกัดตำแหน่งของกราฟในแต่ละรูปแบบการเก็บ ของแต่ละโปรแกรมอาจจะไม่เหมือนกัน เช่นการอ้างอิงตำแหน่งพิกัดอาจจะไม่เหมือนกันและ นำมาใช้ด้วยกันไม่ได้ โปรแกรมนี้ก็จึงไม่พิจารณาพิกัดเหล่านั้น โดยจะพิจารณาเพียงแค่โครงสร้าง แล้วนำมาจัดวางเองเลย

## วิเคราะห์และออกแบบการทำงานของโปรแกรม

การวิเคราะห์และออกแบบการทำงานของโปรแกรม การออกแบบโปรแกรมจะแบ่งออกเป็นส่วนๆ ดังนี้คือ

1. การออกแบบโครงสร้างการเก็บข้อมูล
2. การออกแบบโครงสร้างแฟ้มข้อมูล
3. การออกแบบส่วนติดต่อผู้ใช้

โดยจะอธิบายการออกแบบในแต่ละส่วนโดยละเอียดในส่วนต่อไปนี้

* + 1. **การออกแบบโครงสร้างการเก็บข้อมูล**

โครงสร้างข้อมูลที่สามารถนำมาใช้ในการเก็บข้อมูลประเภทกราฟนั้น อาจจะเป็นไปได้ทั้ง adjacency matrix หรือ adjacency list ผู้พัฒนาได้เลือก adjacency list มาใช้ในการพัฒนาเพราะว่าโครงสร้างข้อมูลแบบนี้ทำให้ สามารถหาเส้นเชื่อมของแต่ละ node ได้ง่ายขึ้น รูปต่อไปนี้จะแสดงการเก็บข้อมูลแบบ adjacency list และกราฟที่ได้จากการเก็บข้อมูลแบบ adjacency list ตามลำดับ

รูปที่ 3‑1 แสดงการเก็บข้อมูลแบบ Adjacency list

รูปที่ 3‑2 แสดงกราฟที่ได้จากข้อมูลกราฟของ Adjacency list

* + 1. **การออกแบบโครงสร้างแฟ้มข้อมูล**

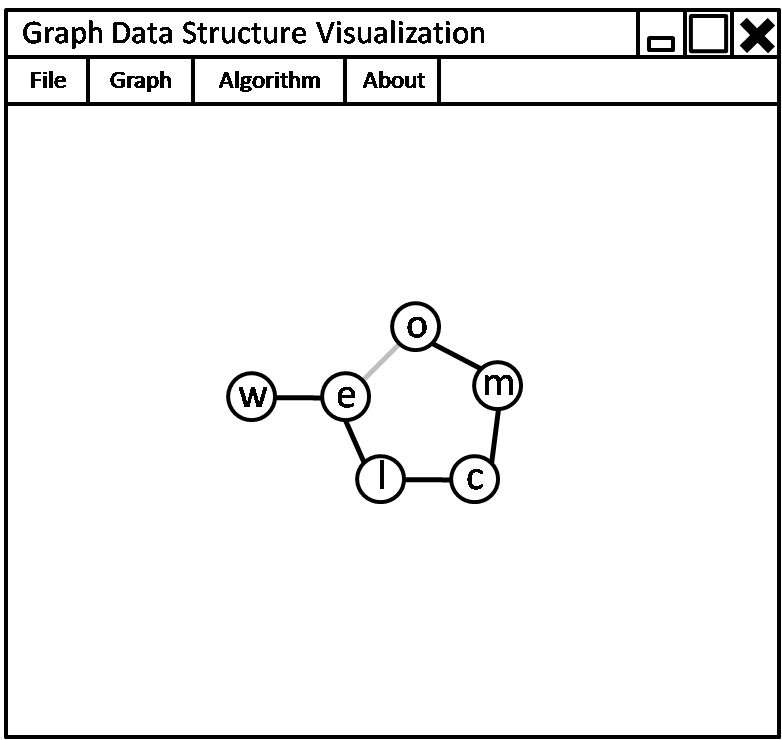
โปรแกรมนี้ต้องการให้มีการเพิ่มขั้นตอนวิธีในการจัดการกับกราฟเข้าไปได้ ซึ่งในส่วนของการเพิ่มขึ้นตอนวิธีในการจัดการกับกราฟนั้น อาจจะเพิ่มเข้าไปได้มากมายหลายวิธี ซึ่งการทำงานในส่วนนี้แทบจะไม่มีส่วนเกี่ยวข้องกับการแสดงผลเลย แต่เน้นไปที่การคำนวณกับข้อมูลของกราฟมากกว่า เพื่อความสะดวกในการจัดการแก้ไขและเข้าใจง่ายจึงได้ทำการแยกแฟ้มข้อมูลส่วนของ ขั้นตอนวิธีออกมาจากส่วนอื่น ซึ่งลักษณะของแฟ้มข้อมูลของโปรแกรมนี้จะมีลักษณะดังรูปนี้

รูปที่ 3‑3 การจัดเก็บแฟ้มข้อมูลของโปรแกรม

ภายได้ไดเรกทอรี Program จะมีส่วนประกอบสำคัญต่างๆ ดังนี้

* ไดเรกทอรี mylib สำหรับเก็บขั้นตอนวิธี
* ไดเรกทอรี images สำหรับเก็บรูปที่ใช้ในโปรแกรม
* ไดเรกทอรี input สำหรับเก็บแฟ้มข้อมูลเข้า
* แฟ้ม DrawWidget สำหรับสร้างส่วนการแสดงผล
* แฟ้ม GraphViewr เป็นแฟ้มของโปรแกรมหลัก
  + 1. **การออกแบบส่วนติดต่อผู้ใช้**

การออกแบบส่วนติดต่อผู้ใช้เบื้องต้นสำหรับโปรแกรมนี้ เพื่อเป็นพื้นฐานในส่วนการพัฒนาต่อไป โดยส่วนประกอบหลักๆ ของส่วนแสดงผลของโปรแกรมนี้ได้แก่ ชื่อโปรแกรม ส่วนของเมนูต่างๆ และส่วนแสดงผลกราฟ โดยสามารถดูภาพรวมของโปรแกรมได้จากรูปต่อไปนี้



รูปที่ 3‑4 ส่วนติดต่อผู้ใช้เบื้องต้น

ส่วนของเมนูของโปรแกรมจะมีอยู่ 4 เมนูคือ

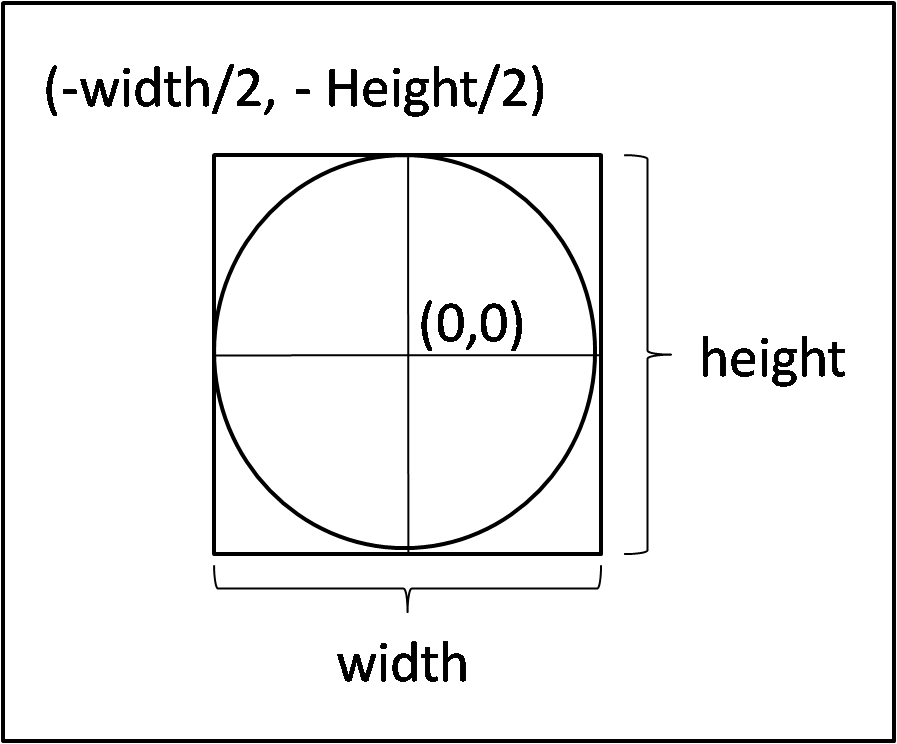
* File เป็นเมนูที่เกี่ยวกับการเปิดและบันทึกแฟ้มต่างๆ
* Graph เป็นส่วนดำเนินการกับกราฟเช่นการเพิ่มลด เส้นเชื่อมและจุดตัด
* Algorithm สำหรับเลือกขั้นตอนวิธีต่างๆ เพื่อใช้กับกราฟ
* About ข้อมูลเกี่ยวกับโปรแกรม

## สร้างส่วนแสดงผลกราฟ

ส่วนแสดงผลกราฟคือส่วนที่จะวาดกราฟในส่วนต่างๆ ออกมาไม่ว่าจะเป็นส่วนของจุดตัด เส้นเชื่อม โดยจะอธิบายการสร้างแต่ล่ะส่วนดังนี้

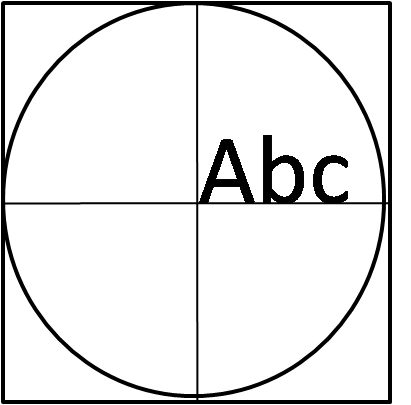
* + 1. **ส่วนวาดจุดตัด**

จุดตัดมีส่วนประกอบสองส่วนคือ ส่วนของรูปวงกลมและส่วนของชื่อจุดตัด โดยการวาดจุดออกแบบเป็นทรงกลม โดยการสร้างรูปของ Qt จะคิดโดยมีพิกัดของวัตถุเป็น (0, 0) และต้องกำหนดคุณลักษณะการวาด 4 คุณลักษณะ ดังนี้คือ จุดเริ่มวาด x, จุดเริ่มวาด y, ความสูง และความกว้าง โดยลักษณะของการวาดจะมีลักษณะเหมือนรูปต่อไป



รูปที่ 3‑5 รูปการวาดจุดตัดเป็นทรงกลม

ส่วนของการวาดชื่อของจุดตัดโดยปกติจะวาดที่พิกัด (0,0) ของวัตถุแต่ว่าเมื่อวาดออกมาแล้วจะออกมาไม่สวยเพราะว่าตัวอักษรจะมีลักษณะดังรูป



รูปที่ 3‑6 แสดงการวาดชื่อของจุดตัด

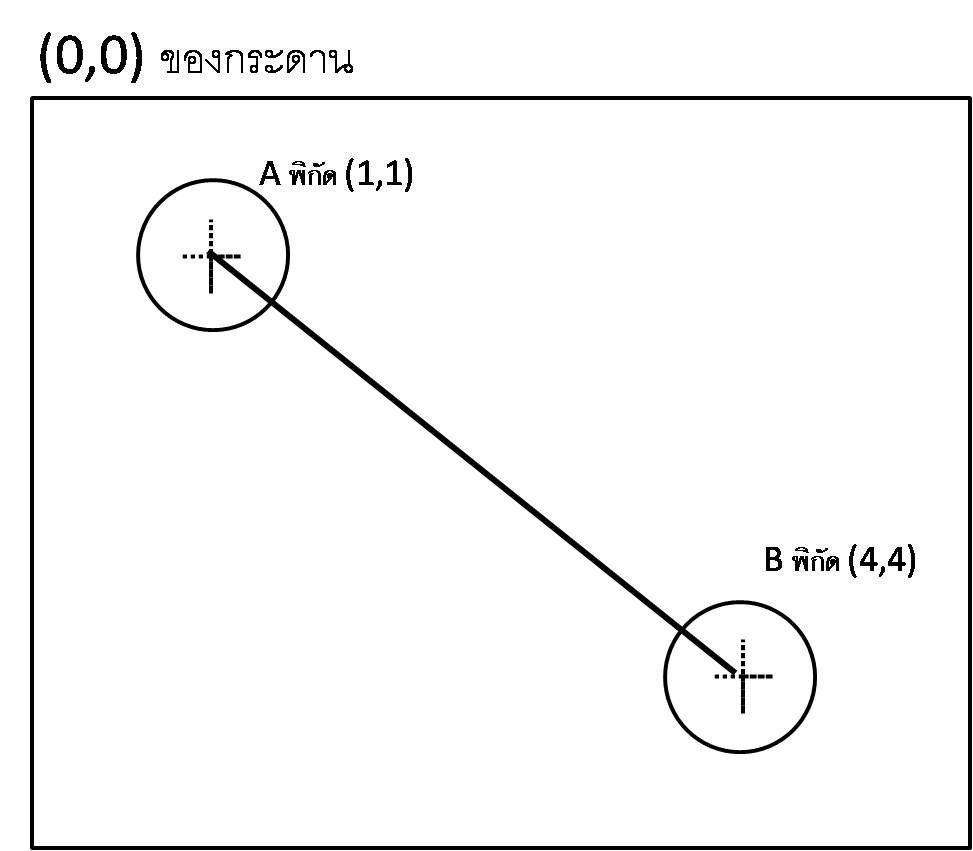
การจะแก้ปัญหาการวางชื่อของจุดตัดนี้จะต้องทำการเลื่อนพิกัดในการวาดชื่อจุดตัดไปทางด้านซ้ายล่าง เพื่อที่จะให้วาดชื่อออกมาแล้วอยู่ตรงกลางของจุดตัด เมื่อวาดออกมาก็จะได้ออกมาดังรูปต่อไป



รูปที่ 3‑7 แสดงการวาดชื่อของจุดตัดหลังจากการเลื่อนพิกัด

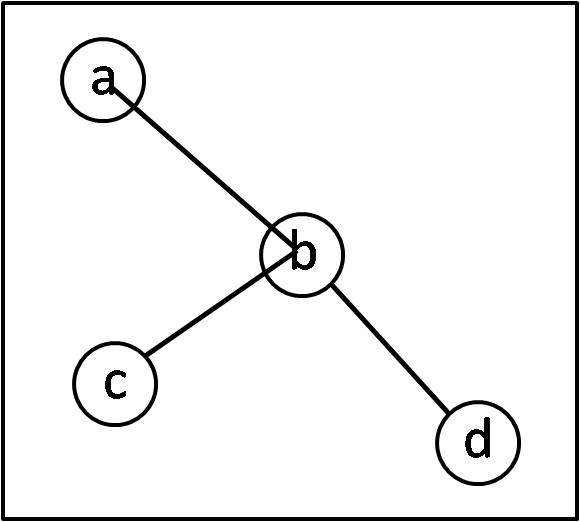
* + 1. **ส่วนการวาดเส้นเชื่อม**

เส้นเชื่อมคือเส้นที่เชื่อมระหว่างจุดตัด 2 จุดตัดโดยที่ตัวมันเองก็มีน้ำหนักด้วย ซึ่งต้องแสดงทั้งน้ำหนักแล้วส่วนของเส้น ถ้ามีเส้นเชื่อมระหว่างจุดตัด A, B จะมีการวางจุดปลายของเส้นเชื่อมไว้ที่จุด (0, 0) ของแต่ละจุดปลาย ซึ่งคือพิกัดของแต่ละจุดตัดบนกระดานวาดนั่นเอง รูปต่อไปจะแสดงการวาดเส้นเชื่อมระหว่างจุดตัด A, B โดยพิกัดเป็นพิกัดสมมติเพื่อให้ดูง่าย



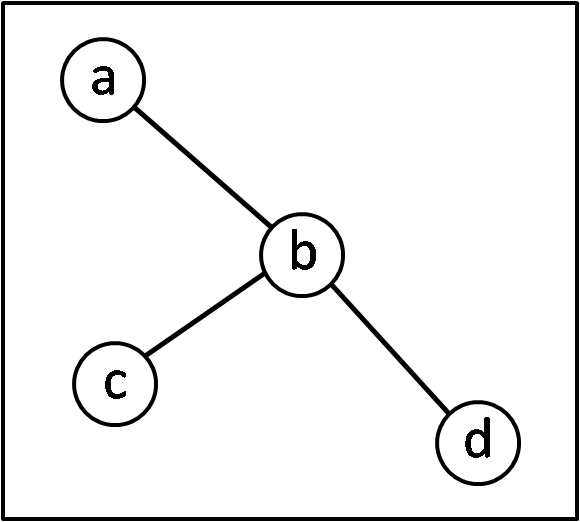
รูปที่ 3‑8 แสดงการวาดเส้นเชื่อม

การวาดภาพลงไปบนกระดานใน Qt นั้นจะวาดสิ่งที่วาดลงไปทีหลังไว้ข้างบนสุดเสมอเมื่อมีการวาดเส้นเชื่อมลงไปหลังจากที่วาดจุดตัดจะทำให้เส้นเชื่อมพาดอยู่บนจุดตัด ซึ่งเมื่อดูแล้วจะทำให้ดูไม่เหมาะสมดังรูป



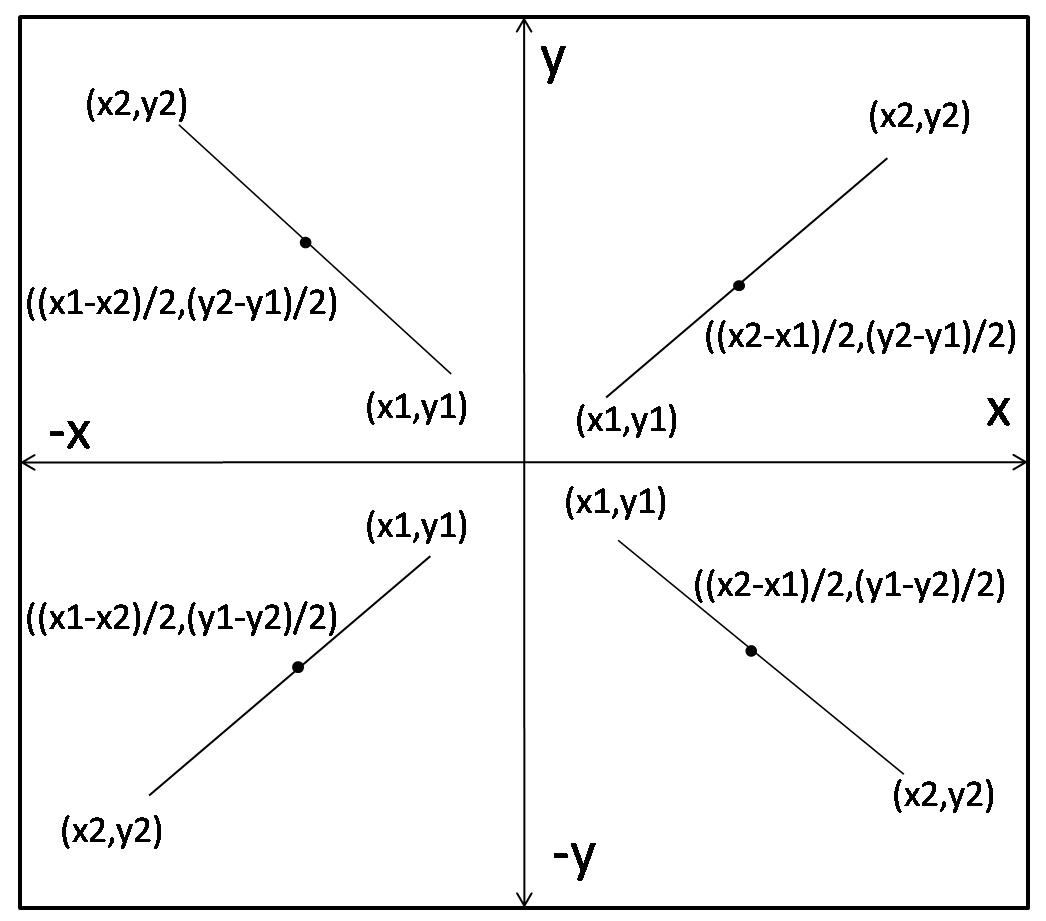
รูปที่ 3‑9 ปัญหาการซ้อนทับของเส้นเชื่อม

ปัญหาการซ้อนทับของเส้นเชื่อมสามารถแก้ได้โดยการกำหนดค่าของวัตถุที่เป็นจุดตัดให้วาดอยู่บนวัตถุที่เป็นเส้นเชื่อมเสมอเมื่อแก้ได้แล้ว การวาดเส้นเชื่อมก็จะออกมามีลักษณะเหมือนดังรูปต่อไป



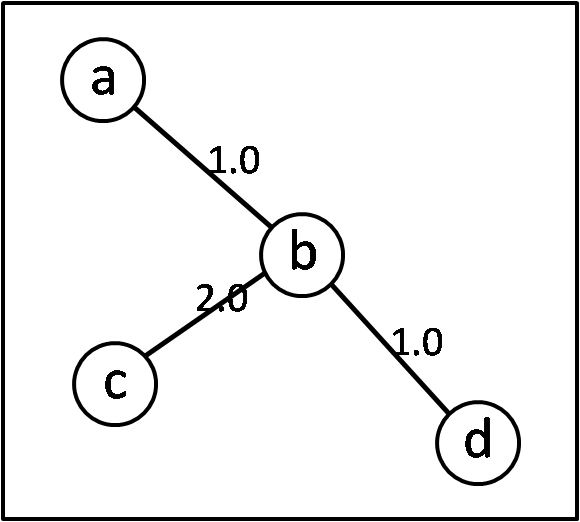
รูปที่ 3‑10 การวาดเส้นเชื่อมที่แก้ไขปัญหาการซ้อนทับแล้ว

การวาดน้ำหนักของเส้นเชื่อมให้ต้องวาดให้อยู่ที่จุดกึ่งกลางของเส้นเชื่อม เนื่องจากเส้นเชื่อมอาจเชื่อมจากจุดตัดไปได้ทุกทิศทาง ดังนั้นการวาดน้ำหนักของเส้นเชื่อมจึงต้องหาจุดกึ่งกลางของเส้นเชื่อมก่อนซึ่งแต่ละ Quadrant ของ ทิศทางเส้นเชื่อมจะมีวิธีคิดที่ต่างกันเล็กน้อยดังรูปต่อไปนี้



รูปที่ 3‑11 แสดงการหาจุดกึ่งกลางของเส้นเชื่อมในแต่ละ Quadrant

เมื่อหาจุดกึ่งกลางของเส้นเชื่อมแล้วได้แล้วจึงจะวาดน้ำหนักของเส้นเชื่อมลงไปที่จุดกึ่งกลางนั้นเมื่อวาดเส้นเชื่อมและจุดตัดเสร็จสมบูรณ์จะได้กราฟออกมาในลักษณะแบบรูปนี้



รูปที่ 3‑12 กราฟที่วาดสมบูรณ์ทั้งเส้นเชื่อม และจุดตัด

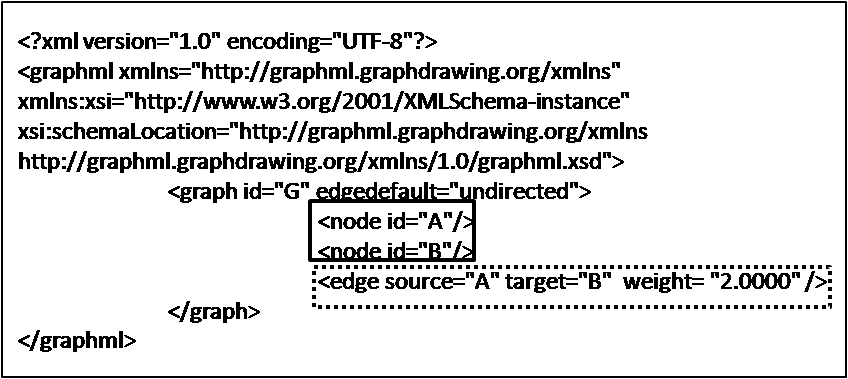
## สร้างส่วนดำเนินการกับไฟล์

เมื่อสร้างส่วนของการวาดกราฟ และส่วนของการแสดงผลกราฟได้แล้ว เพื่อให้สามารถเก็บข้อมูลที่ได้จากการสร้างกราฟ หรือว่าเพื่ออ่านข้อมูลขึ้นมาแสดงผล จำเป็นต้องมีการดำเนินการกับไฟล์เพื่อจัดการในส่วนนี้

การดำเนินการกับไฟล์จำเป็นต้องมีการกำหนดรูปแบบไฟล์ ก่อนที่จะดำเนินการซึ่งในการพัฒนาครั้งนี้ได้ใช้รูปแบบไฟล์ประเภท GraphML ในการเก็บข้อมูลซึ่งเป็นรูปแบบไฟล์ที่เป็น xml ชนิดหนึ่งที่ออกแบบมาเพื่อการเก็บข้อมูลประเภทกราฟโดยเฉพาะ ซึ่งส่วนสำคัญในการจัดทำส่วนดำเนินการกับไฟล์มีดังนี้

1. การอ่านไฟล์ GraphML ขึ้นมาแสดงผล
2. การบันทึกไฟล์ GraphML
   * 1. **การอ่านไฟล์ GraphML ขึ้นมาแสดงผล**

การอ่านไฟล์ GraphML ขึ้นมาแสดงผลใช้ parser ของภาษาไพธอนเองที่ชื่อว่า minidom ในการอ่านไฟล์ xml ของข้อมูลเข้าโดยจะเลือกอ่านขึ้นมาเฉพาะ element ของ node และ edge เท่านั้น โดยส่วนประกอบที่เลือกอ่านขึ้นมาดูได้จากไฟล์ตัวอย่างนี้



รูปที่ 3‑13 ตัวอย่างไฟล์ GraphML ที่อ่าน

จากรูป จะเห็นว่าแต่ละ tag node จะมี attribute id เป็นตัวระบุแยกของแต่ละ node ซึ่ง id นี้ก็จะนำมาเป็นชื่อในการวาดจุดตัด มี tag edge ที่ระบุการเชื่อมโยงด้วย source และ target ของจุดปลาย และมีการน้ำหนักด้วย attribute weight เมื่ออ่านข้อมูลขึ้นมาแล้วก็จะนำข้อมูลเหล่านี้ไปเก็บไว้ที่ adjacency list ของกราฟ และนำไปแทนที่กราฟที่แสดงอยู่ในปัจจุบัน

* + 1. **การบันทึกไฟล์ GraphML**

การบันทึกข้อมูลหลังจากการ สร้างหรือแก้ไขกราฟด้วยโปรแกรมแล้ว จะใช้วิธีการเขียนไฟล์ธรรมดาซึ่งง่ายกว่าการใช้ parser ถ้าต้องการเก็บแค่ข้อมูลพื้นฐานเท่านั้น โดยจะทำงานเป็นขั้นตอนดังนี้

1. เตรียม String ข้อความส่วนหัวของไฟล์ประกอบด้วยการประกาศ ต่างๆ และ root element
2. สร้าง element ของกราฟ ในส่วนของ node และ edge จาก adjacency list ทีละแถวโดยตรวจสอบให้ไม่ซ้ำกันด้วย แล้วนำมาสร้างต่อ String ในข้อ 1.
3. เมื่ออ่านข้อมูลของ adjacency list ออกมาต่อหมดแล้ว ให้ต่อด้วยการปิด element
4. นำ String ที่ต่อไว้ มาเขียนลงไฟล์

## สร้างส่วนขั้นตอนวิธี

ส่วนขั้นตอนวิธีที่ใช้ดำเนินการกับกราฟ จะทำงานโดยการส่งข้อมูลของกราฟเข้าไป แล้วแสดงผลการทำงานออกมาโดยการเน้นสีกราฟ และแสดงเป็นข้อมูลสรุปในสำหรับบางขั้นตอนวิธี โดยการเรียกใช้ขั้นตอนวิธีในตอนนี้แบ่งออกเป็น 2 แบบคือแบบต้องการข้อมูลเข้า และแบบที่ไม่ต้องการข้อมูลเข้า โดยขั้นตอนการทำงานของการเรียกใช้ขั้นตอนวิธีจะเป็นแบบนี้

ข้อมูลเข้าเบื้องต้น : adjacency list ของกราฟปัจจุบัน

1. ถ้ากราฟต้องการข้อมูลเข้าอื่นให้สร้างกล่องข้อความเพื่อรับข้อมูลเข้าเหล่านั้น
2. ส่งข้อมูลเข้าให้กับขั้นตอนวิธีที่เรียกใช้ โดยจะได้ผลลัพธ์ออกมาเป็น adjacency list ของกราฟผลลัพธ์ และข้อมูลหลังจากการทำงานของขั้นตอนวิธี ซึ่งบางขั้นตอนวิธีอาจจะไม่มีส่วนหลัง
3. นำข้อมูลของ adjacency list ของกราฟผลลัพธ์ ไปเน้นสีให้กับกราฟที่แสดงอยู่ในปัจจุบัน
4. นำข้อมูลหลังจากการทำงานของขั้นตอนวิธี มาแสดงผล(ถ้ามี)