**รายงาน**

**เรื่อง การหาดีกรีของจุดยอดในกราฟ โดยการใช้โปรแกรมในการวาดกราฟ**

**จัดทำโดย**

นายธนกฤต มีแย้ม 620710166

นายคฑากาญจน์ ทรัพย์เสริม 620710650

นายณัฏฐวุฒิ อภินันทกุล 620710657

นางสาวปัทมพร เอื้อมณีรัตนกุล 620710728

**เสนอ**

อาจารย์ ผู้ช่วยศาสตราจารย์ ดร.คทา ประดิษฐวงศ์

**รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา**

โครงสร้างเชิงการคำนวณแบบไม่ต่อเนื่อง รหัสวิชา 517213-2560

มหาวิทยาลัยศิลปากร วิทยาเขตสนามจันทร์

ภาคเรียนที่ 1 ปีการศึกษา 2563

**คำนำ**

รายงานฉบับนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชาโครงสร้างเชิงการคำนวณแบบไม่ต่อเนื่อง รหัสวิชา 517213-2560 ของนักศึกษาปีการศึกษา 2563 เพื่อให้ได้ศึกษาหาความรู้ในเรื่องการหาดีกรีของจุดยอดในกราฟ โดยการใช้โปรแกรมในการวาดกราฟ และได้ศึกษาอย่างเข้าใจเพื่อเป็นประโยชน์กับการเรียนมากที่สุด

รายงานฉบับนี้ จะเป็นประโยชน์ต่อผู้ที่สนใจจะศึกษา ที่กำลังหาข้อมูลอยู่ หากมีข้อผิดพลาดประการใด คณะผู้จัดทำขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

**สารบัญ**

ส่วนเพิ่มเติมของโปรแกรม...........................................................................................................................1

การเก็บกราฟ ข้อมูลในตารางและเซตจุดยอด และเส้นเชื่อม......................................................................9

การ compile หรือ run โปรแกรม..........................................................................................................10

คุณสมบัติหรือความสามารถของโปรแกรม...............................................................................................18วิธีการใช้งานของโปรแกรมและการแสดงผล............................................................................................19ตัวอย่างทดสอบแบบที่ 1..........................................................................................................................31ตัวอย่างทดสอบแบบที่ 2..........................................................................................................................40

อ้างอิง.......................................................................................................................................................43

**ส่วนเพิ่มเติมของโปรแกรม**

Class Vertex

|  |  |  |
| --- | --- | --- |
| Field / Method | Type / Return type | Description |
| x | int | พิกัดแกน x ของจุดศูนย์กลางของจุดยอด |
| y | int | พิกัดแกน y ของจุดศูนย์กลางของจุดยอด |
| name | String | ชื่อของจุดยอด |
| r | int | รัศมีถึงขอบนอกของจุดยอด |
| shift | int | รัศมีขอบในของจุดยอด |
| IsSelect | boolean | จุดยอดนี้ถูกเมาส์คลิ๊กเลือกไว้หรือไม่ |
| Id | int | เก็บ id เฉพาะของ Vertex |
| idGen | int | สร้าง id เฉพาะ Vertex |
| Vertex(x:int,y:int) | - | r = 36 , shift = 30 , name = “” , isSelect = false |
| inCircle(x0:int,y0:int) | boolean | ตวรจสอบว่า พิกัด (x,y) ที่รับเข้ามา อยู่ภายในจุดยอดนี้ไหม |
| draw(g:Graphics) | void | วาดรูปจุดยอดพร้อมชื่อจุดยอดลงบน Canvas |

Class Edge\_

|  |  |  |
| --- | --- | --- |
| Field / Method | Type / Return type | Description |
| VertexA | Vertex | เก็บตัวแปรอ้างอิงวัตถุของจุดยอด ที่เป็นจุดปลายเส้นเชื่อม |
| VertexB | Vertex | เก็บตัวแปรอ้างอิงวัตถุของจุดยอด ที่เป็นจุดปลายเส้นเชื่อม |
| weight | String | เก็บค่าน้ำหนักของเส้นเชื่อม เป็นข้อความ (ตัวเลข หรือตัวแปร) |
| isSelect | boolean | เส้นเชื่อมนี้ถูกเมาส์คลิ๊กเลือกไว้หรือไม่ |
| x\_center | int | พิกัดแกน x ของ weight [ จุดปรับความโค้ง หรือ จุดหมุนวงวน ] |
| y\_center | int | พิกัดแกน y ของ weight [ จุดปรับความโค้ง หรือ จุดหมุนวงวน ] |
| r\_center | int | รัศมีรอบข้อความ weight ที่สามารถคลิ๊กเพื่อเลือกเส้นได้ |
| Edge\_(a:Vertex,b:Vertex) | - | r\_center = 50 , weight = “1” , isSelect = false |
| inLine(x0:int,y0:int) | boolean | ตวรจสอบว่า พิกัด (x,y) ที่รับเข้ามา อยู่ภายในเส้นไหม (ภายในรัศมี r\_center รอบ weight ) |
| draw(g:Graphics2D) | void | วาดเส้นเชื่อม มี 2 กรณี คือเส้นเชื่อมระหว่าง 2 จุดยอด กับ เส้นเชื่อมวงวน |

Class TempEdge

|  |  |  |
| --- | --- | --- |
| Field / Method | Type / Return type | Description |
| VertexA | Vertex | เก็บตัวแปรอ้างอิงวัตถุของจุดยอด ที่เป็นจุดปลายเส้นเชื่อม |
| x1 | int | พิกัดแกน x ของปลายเส้นเชื่อมปัจจุบันที่เมาส์ชี้อยู่ |
| y1 | int | พิกัดแกน y ของปลายเส้นเชื่อมปัจจุบันที่เมาส์ชี้อยู่ |
| TempEdge(x:int,y:int) | - | - |
| setA(a:Vertex) | void | กำหนดค่า จุดยอดเริ่มต้น |
| line(g:Graphics2D) | void | วาดเส้นเชื่อม คือเส้นเชื่อมระหว่าง จุดยอดกับจุดปัจจุบันที่เมาส์ชี้ อาจะเป็นเส้นเชื่อมงวนได้ |

Class GraphDrawing

|  |  |  |
| --- | --- | --- |
| Field / Method | Type /  Return type | Description |
| Vertexs | ArrayList<Vertex> | V(G) เอาไว้เก็บจุดยอดของกราฟ |
| Edge\_s | ArrayList  <Edge\_s> | E(G) เอาไว้เก็บเส้นเชื่อมของกราฟ |
| Selected | Object | เอาไว้อ้างอิง จุดยอดหรือเส้มเชื่อมที่จะทำการแก้ไข โดยใช้เมาส์คลิ๊กเลือก |
| TempEdge | TempEdge | เส้นเชื่อมที่อยู่ในขั้นตอนการวาด ยังไม่ถูกเพิ่มลงใน E(G) |
| c | Canvas | พื้นที่สำหรับวาดกราฟ |
| mode | String | มีค่าระหว่าง “Vertex” กับ “Edge\_” ใช้เพื่อแบ่งโหมดการวาดกราฟ ระหว่างการจัดการเส้นเชื่อม และการจัดการจุดยอด ออกจากกัน กด space bar ค้างเพื่อให้อยู่ใน mode “Edge\_” นอกจากนั้นจะอยู่ใน mode “Vertex” |
| sanSeriFont | Font | รูปแบบตัวอักษรที่แสดงผล (“SanSerif”,Font.PLAIN,24) |
| ScreenSize | Dimension | ขนาดของหน้าจอปัจจุบัน |
| FrameHelp | Jframe | Frame สำหรับแสดงข้อความช่วยเหลือ |
| FrameDegree | Jframe | Frame สำหรับแสดงดีกรีทุกจุดยอด |
| BoxSave | JPanel | Panel สำหรับการบันทึกไฟล์ |
| BoxOpen | JPanel | Panel สำหรับการเปิดไฟล์ |
| BoxHelp | JPanel | Panel สำหรับการแสดงข้อความช่วยเหลือ |
| BoxDegree | JPanel | Panel สำหรับการแสดงดีกรีทุกจุดยอด |
| SaveButt | JButton | ปุ่มกดเพื่อบันทึกกราฟการทำงานจะอยู่ใน method saveButtAction(e:ActionEvent) |
| OpenButt | JButton | ปุ่มกดเพื่ออ่านกราฟการทำงานจะอยู่ใน method openButtAction(e:ActionEvent) |
| HelpButt | JButton | ปุ่มกดเพื่อแสดงข้อความช่วยเหลือ การทำงานอยู่ใน method helpButtAction(e:ActionEvent ) |
| DegreeButt | JButton | ปุ่มกดเพื่อแสดงดีกรีทุกจุดยอด ทำงานอยู่ใน method DegreeButtAction(e:ActionEvent ) |
| PathSave | JFileChooser | ใช้สำหรับกำหนดที่อยู่ไฟล์ json ที่ต้องการจะบันทึกข้อมูลกราฟ |
| PahtOpen | JFileChooser | ใช้สำหรับเลิือกไฟล์ json เพื่อนำเข้ากราฟเข้าโปรแกรม |
| HelpString | JLabel | เก็บข้อความช่วยเหลือ |
| Menubar | JPanel | แถบสำหรับวางปุ่ม save , open , help |
| Shift | int | ขนาดความกว้างของ menubar |
| Backup | class | Class สำหรับรวม Object ทั้งหมดของกราฟ ภายในมี  ArrayList<Vertex> VertexsBackup;  ArrayList<Edge\_> Edge\_sBackup; |
| Main(args[] : String) | void | Method main สำหรับรันโปรแกรม |
| helpButtAction (e:ActionEvent ) | void | การทำงานเมื่อกดปุ่ม help จะแสดง Frame ใหม่ขึ้นมาที่กลางจอภายใน frame จะประกอบไปด้วยข้อความอธิบายการใช้งานโปรแกรม  เบื้องต้น |
| saveButtAction (e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม save แสดงหน้าต่างให้กำหนดตำแหน่งไฟล์ที่จะบันทึกกราฟ จากนั้นจะเรียกใช้ method save(path) เพื่อบันทึกกราฟ |
| degreeButtAction  (e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม Degree จะแสดง Frame ใหม่ขึ้นมาที่กลางจอภายใน frame จะประกอบไปด้วยข้อความอธิบายดีกรีแต่ละจุดยอด  ทุกจุดยอด |
| openButtAction  (e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม open จะแสดงหน้าต่างให้เลือกตำแหน่งไฟล์ที่กราฟที่จะนำเข้ามาเปิดในโปรแกรม จากนั้นจะเรียกใช้ method open(path) เพื่ออ่านไฟล์กราฟ |
| save(path:String) | void | ใช้ Gson ในการแปลง object Backup เป็น json และบันทึกไฟล์ลงในเครื่องตามตำแหน่งทีกำหนด |
| open(path:String) | void | อ่านไฟล์กราฟ แล้วใช้ Gson ในการแปลง json  เป็น object Backup และแสดงกราฟบนหน้าจอ |
| Clear() | void | ทำให้ canvas กลายเป็นสีขาวทั้งหมด |
| Selected(x:int,y:int) | void | รับพิกัด x,y ที่เมาส์คลิ๊ก ถ้าตำแหน่งที่คลิ๊กคือ vertex แล้ว vertex นั้นจะเป็นวัตถุที่ถูกเลือก ถ้าไม่มี vertex ที่ตำแหน่งนั้นก็จะไปตรวจสอบเส้นเชื่อมในแบบเดียวกัน ถ้าตำแหน่งนั้นไม่ใช่ทั้งเส้นเชื่อมและจุดยอด วัตถุที่ถูกเลือกจะถูกยกเลิกการเลือก |
| Draw() | void | วาดจุดยอดและเส้นเชื่อมทั้งหมดลงบน canvas  ถ้าเส้นเชื่อมหรือจุดยอดนั้นถูกเลือกจะแสดงสีน้ำเงิน ถ้าไม่ใช่จะเป็นสีดำ  (วาดเส้นเชื่อม ก่อน แล้วค่อยวาดจุดยอดทับ) |
| keyTyped(ke:KeyEvent) | void | เมื่อกดแป้นพิมพ์ เมธอดนี้จะทำงานโดยจะตรวจว่ากดอะไรเข้ามา  ถ้ากด ctrl+s จะทำการบันทึกกราฟลงไฟล์ backup.json  ctrl+o จะเปิดกราฟจากไฟล์ backup.json  ถ้า vertex ถูกเลือกอยู่ การพิมพ์จะเป็นการพิมพ์ชื่อให้แก่ vertex  ถ้า edge\_ ถูกเลือกอยู่ การพิมพ์จะเป็นการพิมพ์ค่าให้แก่ weight ของเส้น |
| keyPressed(ke:KeyEvent) | void | ถ้ากด spacebar ค้างในระขณะที่กดค้างโปรแกรมจะอยู่ใน mode “Edge\_” |
| keyReleased(ke:KeyEvent) | void | ถ้าเลิกกด keyboard โปรแกรมจะอยู่ใน mode “Vertex” |
| mouseMoved  (me MoouseEvent) | void | - |
| mouseClicked  (e:MouseEvent) | void | เมื่อคลิ๊กเมาส์ลงไป 1 ครั้งจะเป็นการเลือกวัตถุในกราฟ (vertex , edge)  ถ้า double click จะเป็นการสร้าง Vertex ใหม่ทีตำแหน่งที่ double click |
| mouseDragged  (e:MouseEvent) | void | การกดเมาส์ค้าง  ถ้าอยู่ใน mode “Vertex”  จะเป็นการเปลี่ยนตำแหน่งของเส้นเชื่อมหรือจุดยอดที่  ถูกเลือกอยู่  ถ้าอยู่ใน mode “Edge\_” จะเป็นการวาดเส้นเชื่อม แต่จะวาดได้เมื่อเริ่มกดจากภายในจุดยอดไปสู่จุดยอดเท่านั้น |
| mouseReleased  (e:MouseEvent) | void | การออกจากการคลิ๊ก (เมื่อคลิ๊กเสร็จแล้ว) ถ้าอยู่ใน mode “Vertex”จะยกเลิกการวาดเส้นเชื่อมที่ยังวาดไม่เสร็จ  ถ้าอยู่ใน mode “Edge\_” จะทำการสร้างเส้นเชื่อมขึ้นมา โดยมี 2 แบบ คือเส้นเชื่อมระหว่าง 2 จุดยอดกับ เส้นเชื่อมวงวน |
| mousePressed  (e:MouseEvent) | void | เมื่อเริ่มกดคลิ๊กเมาส์ และอยู่ใน mode “Edge\_” จะสร้างเส้นเชื่อมชั่วคราวขึ้นมาโดยจุดยอดเริ่มต้นคือจุดยอดในตำแห่นงที่เมาส์เริ่มคลิ๊ก ถ้าตำแหน่งนั้นไม่มีจุดยอด จะไม่สร้างเส้นเชื่อมขึ้นมาให้ |
| mouseExited  (e:MouseEvent) | void | - |
| mouseEntered  (e:MouseEvent) | void | - |

Class ShowSolution

|  |  |  |
| --- | --- | --- |
| Field / Method | Type / Return type | Description |
| Model | class | เก็บข้อมูลที่ได้จาก Prim’s Algorithm |
| round | int | รอบการทำงานของ Prim’s Algorithm |
| sumWeight | double | ผลรวมของค่าน้ำหนักของเส้นทางที่สั้นที่สุด |
| start | Vertex | จุดยอดเริ่มต้น จากการเลือกโดยผู้ใช้ |
| answerLabel | JLabel | เก็บคำตอบของโปรแกรม |
| scroll | JScrollPane | สร้าง scroll ของตาราง |
| primTable | JTable | นำข้อมูลจาก Model มาสร้างเป็นตาราง |
| graph | JButton | ปุ่มกดเพื่อแสดงกราฟคำตอบ การทำงานจะอยู่ใน method graphButtAction(e:ActionEvent) |
| table | JButton | ปุ่มกดเพื่อแสดงตารางคำตอบ การทำงานจะอยู่ใน method tableButtAction(e:ActionEvent) |
| next | JButton | ปุ่มกดเพื่อคำตอบหรือเส้นเชื่อมถัดไปของคำตอบ การทำงานจะอยู่ใน method nextButtAction(e:ActionEvent) |
| home | JButton | ปุ่มกดเพื่อแสดงกราฟตั้งต้น การทำงานจะอยู่ใน method homeButtAction(e:ActionEvent) |
| Vertexs | ArrayList<Vertex> | V(G) เอาไว้เก็บจุดยอดของกราฟ |
| Edge\_s | ArrayList<Edge\_s> | E(G) เอาไว้เก็บเส้นเชื่อมของกราฟ |
| T | LinkedList<Edge\_> | เซตของเส้นเชื่อมที่เป็นเซตคำตอบของต้นไม้ทอดข้าม |
| A | LinkedList<Vertex> | เซตของจุดยอดที่ไม่ได้อยู่ต้นไม้ทอดข้าม |
| N | LinkedList<Vertex> | เซตของจุดยอดที่อยู่ต้นไม้ทอดข้าม |
| u | Vertex | จุดยอดเริ่มต้น |
| c | Canvas | พื้นที่สำหรับวาดกราฟ |
| sanSerifFont | Font | รูปแบบตัวอักษรที่แสดงผล (“SanSerif”,Font.PLAIN,24) |
| menubar | JPanel | Panel สำหรับเมนูบาร์ |
| shift | int | ขนาดความกว้างของ menubar |
| homeButtAction(e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม home จะแสดง frame ของรูป graph ตั้งต้นเดิม |
| primAlgorithm() | void | ส่วนของการทำงานของ Prim’s Algorithm ในการหาต้นไม้แบบทอดข้าม |
| graphButtAction(e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม graph จะแสดง frame ของรูป graph ที่ได้ผ่านการประมวลผลของ Prim’s Algorithm |
| tableButtAction(e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม table จะแสดง frame ของตารางคำตอบ ที่ได้ผ่านการประมวลผลของ Prim’s Algorithm |
| nextButtAction(e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม next จะแสดง frame ของตารางคำตอบถัดไป |
| draw() | void | วาดกราฟลง canvas |

**การเก็บกราฟ ข้อมูลในตารางและเซตจุดยอด และเส้นเชื่อม**

* 1. **กราฟ**

โปรแกรมจะเก็บจุดยอดในลักษณะของ ArrayList <Vertexs> และ เก็บเส้นเชื่อมในลักษณะของ ArrayList <Edge\_s>

* 1. **ตาราง**

2.1 หากจุดยอด w ที่ถูกเลือกอยู่ใน LinkedList<Vertex> N ให้เก็บ - ลงในตาราง

2.2 ถ้าจุดยอด w ประชิดกับจุดยอด u ให้ทำการเปรียบเทียบน้ำหนักของ w ก่อนหน้าในตารางกับ เส้นเชื่อมที่ตกกระทบ หาก u < w ให้ทำการปรับค่าโดยใส่ น้ำหนัก และจุดยอด u คั่นด้วย , เพื่อแยกจุดยอดและน้ำหนัก

2.3 หาก w ไม่ประชิดกับ u จะไม่ปรับค่าลงตาราง

* 1. **เซตจุดยอด**

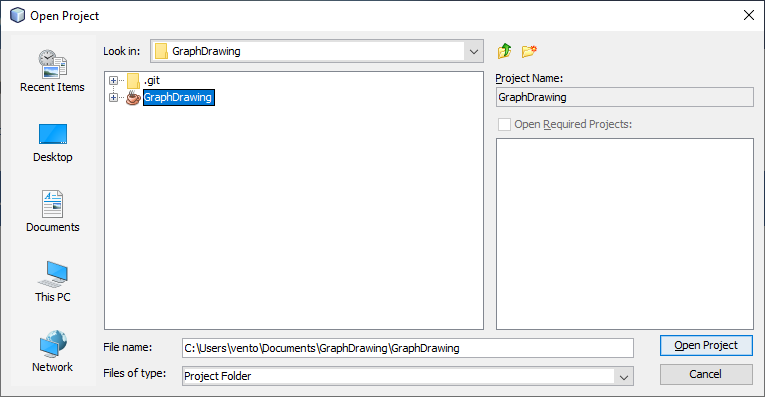
โปรแกรมจะเก็บเซตของจุดยอดในลักษณะของ LinkedList โดยจะมี LinkedList ที่เก็บจุดยอดที่ยังไม่ได้อยู่ในต้นไม้แบบทอดข้าม คือ LinkedList<Vertex> A และ LinkedList ที่เก็บจุดยอดที่อยู่ในต้นไม้แบบทอดข้าม คือ LinkedList<Vertex> N เมื่อ compile หรือ run โปรแกรมนั้น เมื่อเข้าสู่กระบวนการของ Prim’s Algorithm ผู้ใช้จะทำการเลือกจุดเริ่มต้น ทำให้จุดยอดที่เลือกนั้นอยู่ใน LinkedList<Vertex> N และถูกลบออกจาก LinkedList<Vertex> A จากนั้นหาจุดยอด u ใน LinkedList<Vertex> A ที่ประชิดกับจุดยอด w ใน LinkedList<Vertex> N และเส้นเชื่อม uw เป็นเส้นเชื่อมที่มีค่าน้ำหนักน้อยที่สุด จากนั้นเพิ่ม u ใน LinkedList<Vertex> N และลบ u ออกจาก LinkedList<Vertex> A ไปเรื่อยๆจนกว่า LinkedList<Vertex> A จะเป็นเซตว่างหรือไม่เหลือสมาชิกแล้ว

* 1. **เซตเส้นเชื่อม**

โปรแกรมจะเก็บเซตของเส้นเชื่อมในลักษณะ LinkedList โดยจะมี LinkedList ที่เก็บเส้นเชื่อมที่ คือLinkedList<Edge\_> T เมื่อ compile หรือ run โปรแกรมนั้น เมื่อเข้าสู่กระบวนการของ Prim’s Algorithm โดยแรกเริ่ม LinkedList<Edge\_> T จะมีค่าเป็นเซตว่าง จากนั้นหาจุดยอด u ใน LinkedList<Vertex> A ที่ประชิดกับจุดยอด w ใน LinkedList<Vertex> N และเส้นเชื่อม uw เป็นเส้นเชื่อมที่มีค่าน้ำหนักน้อยที่สุด จากนั้นเพิ่ม uw ใน LinkedList<Edge\_> T เพิ่ม u ใน LinkedList<Vertex> N และลบ u ออกจาก LinkedList<Vertex> A ไปเรื่อยๆจนกว่า LinkedList<Vertex> A จะเป็นเซตว่างหรือไม่เหลือสมาชิกแล้ว

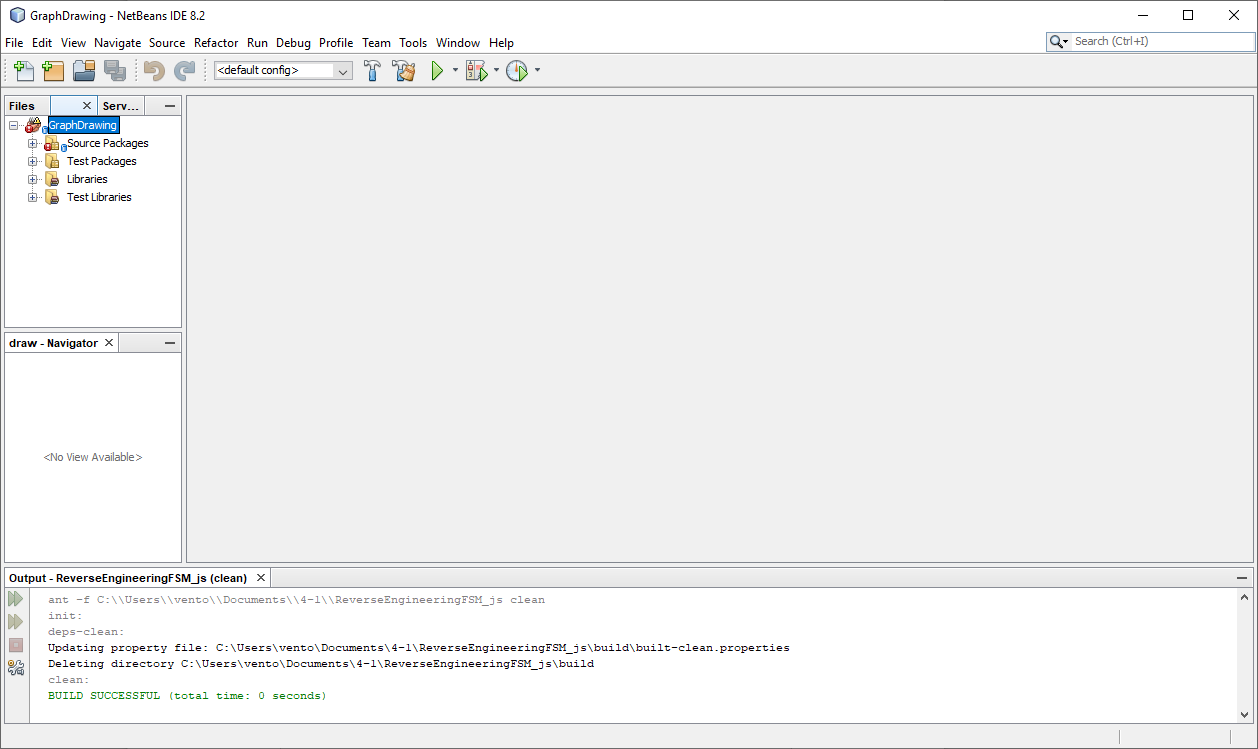
**การ compile หรือ run โปรแกรม**

1. compile โปรแกรมภาษา java ผ่านโปรแกรม IDE Netbean8.2 โดย open Project ชื่อ GraphDrawing ซึ่งภายในมี source code อยู่ใน Folder src



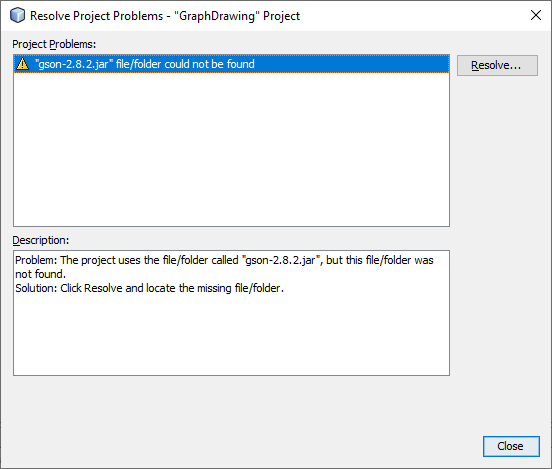
รูปที่ 1 แสดงการ open Project

2. ให้กดคลิ๊กขวาที่ ชื่อโปรเจคที่เปิดแล้วจากนั้นกด Resolve Project Problems ..เพื่อเพิ่ม GSON เข้ามาในโปรเจค



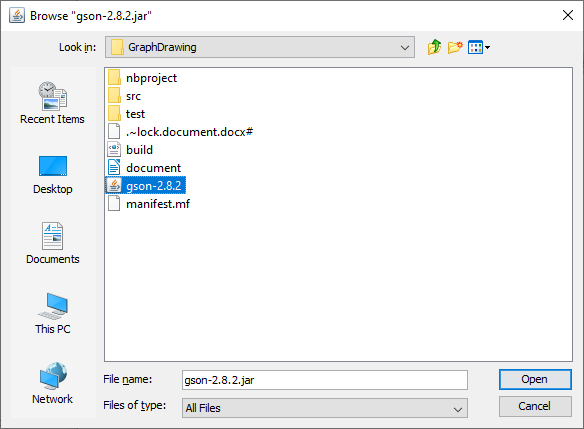
รูปที่ 2 แสดงการเพิ่ม GSON เข้ามาในโปรเจค

3. กด Resolve



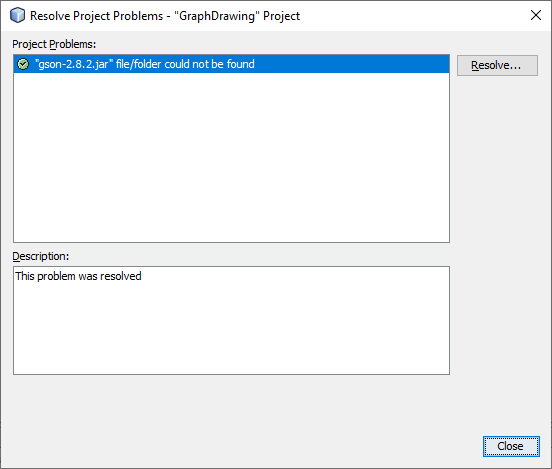
รูปที่ 3 แสดงหน้าต่าง Resolve

4. เข้าไปสู่ Directory ของ Project >> กดเลือกไฟล์ gson-2.8.2.jar >> กด open

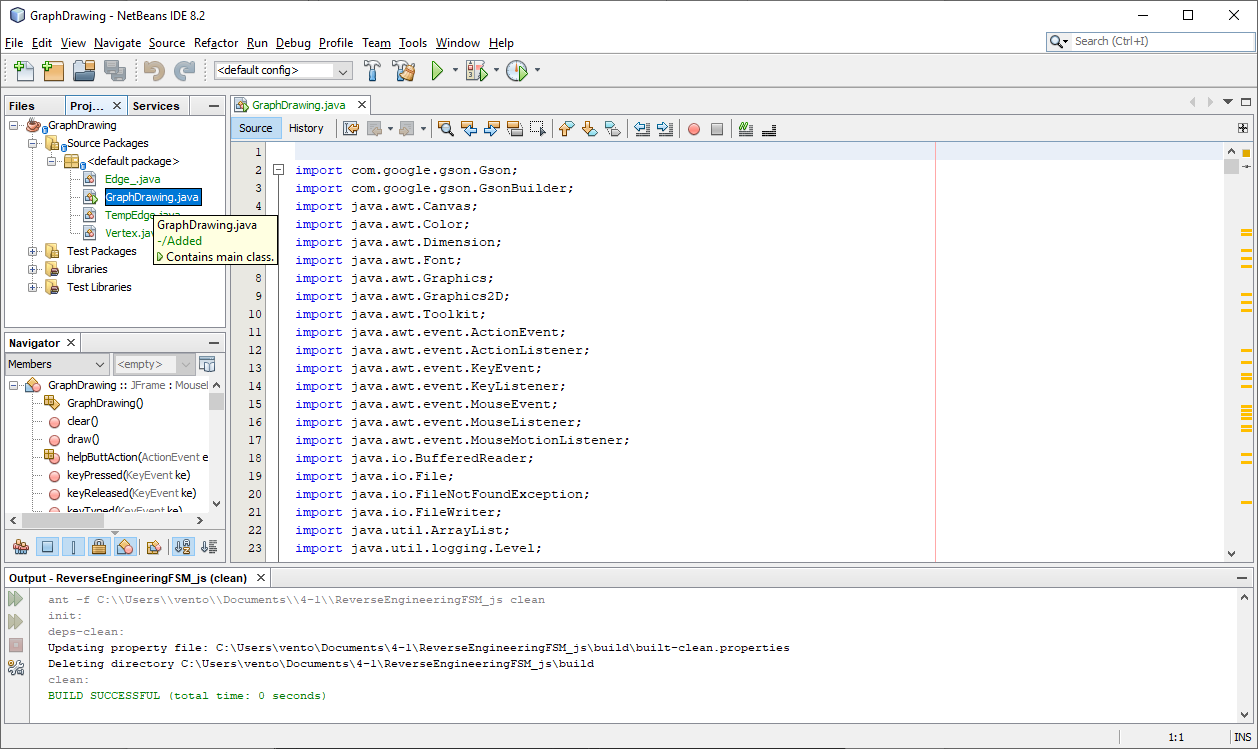


รูปที่ 4 แสดงการเปิดไฟล์ gson-2.8.2.jar

1. กด Close

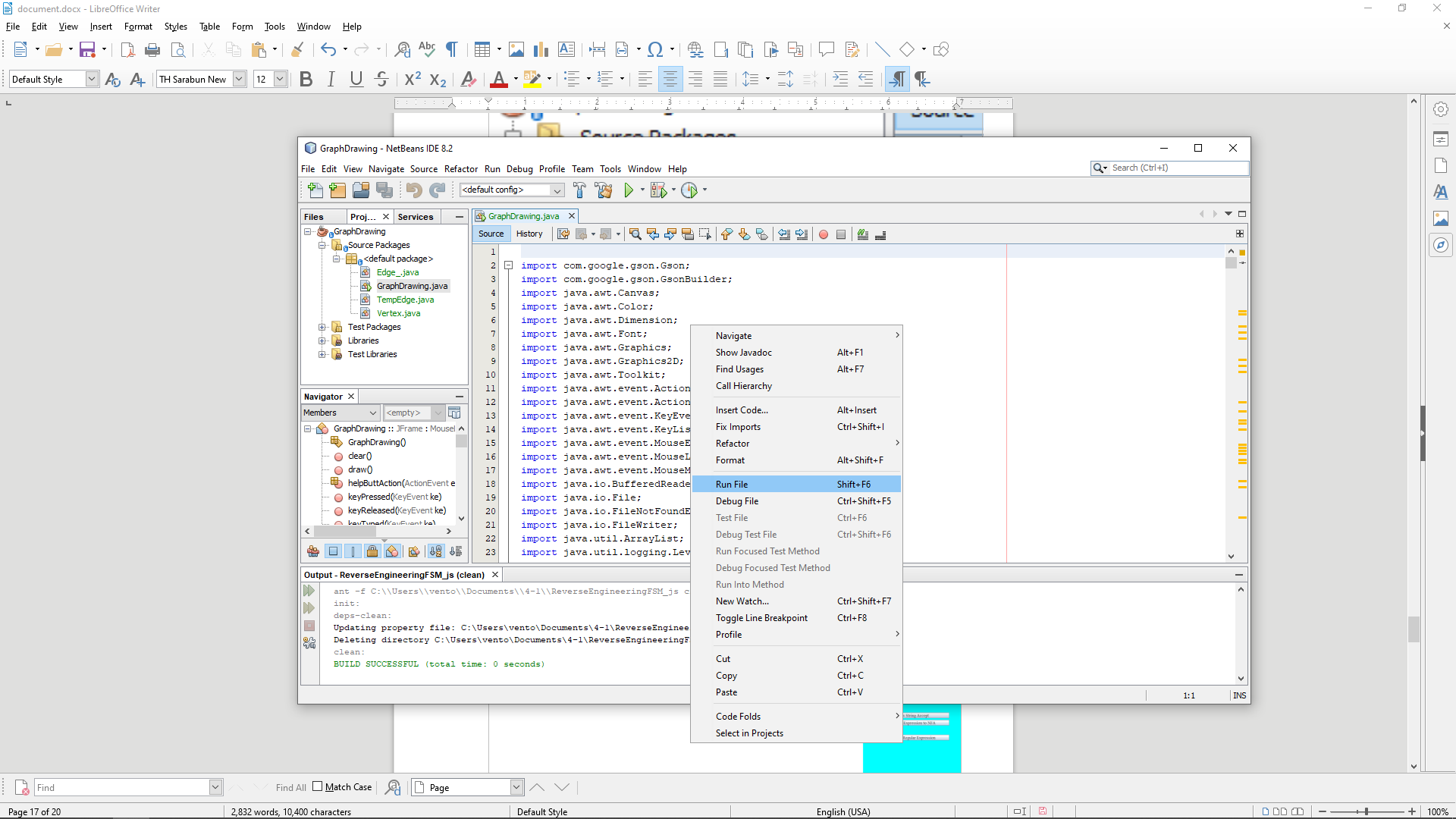


รูปที่ 1 แสดงหน้าต่างหลังจากเพิ่ม GSON

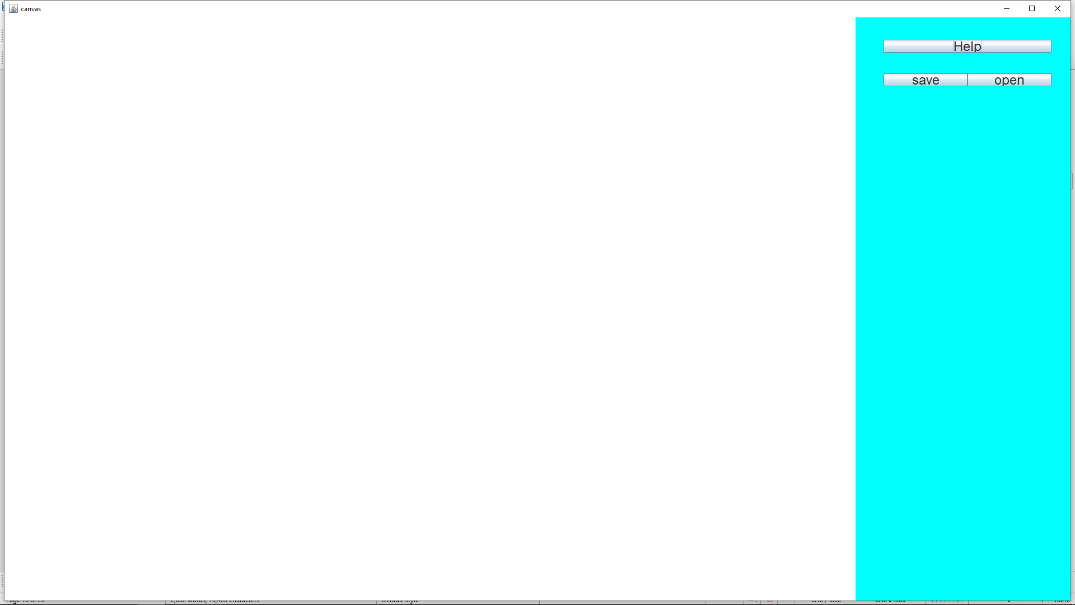
 6. เข้าสู่ Project GraphDrawingกดเข้าไปสู่ java ไฟล์ ดังนี้ GraphDrawing >> Source Packages >> <default Packages> >> GraphDrawing.java

รูปที่ 6 แสดงการเปิดไฟล์ GraphDrawing.java

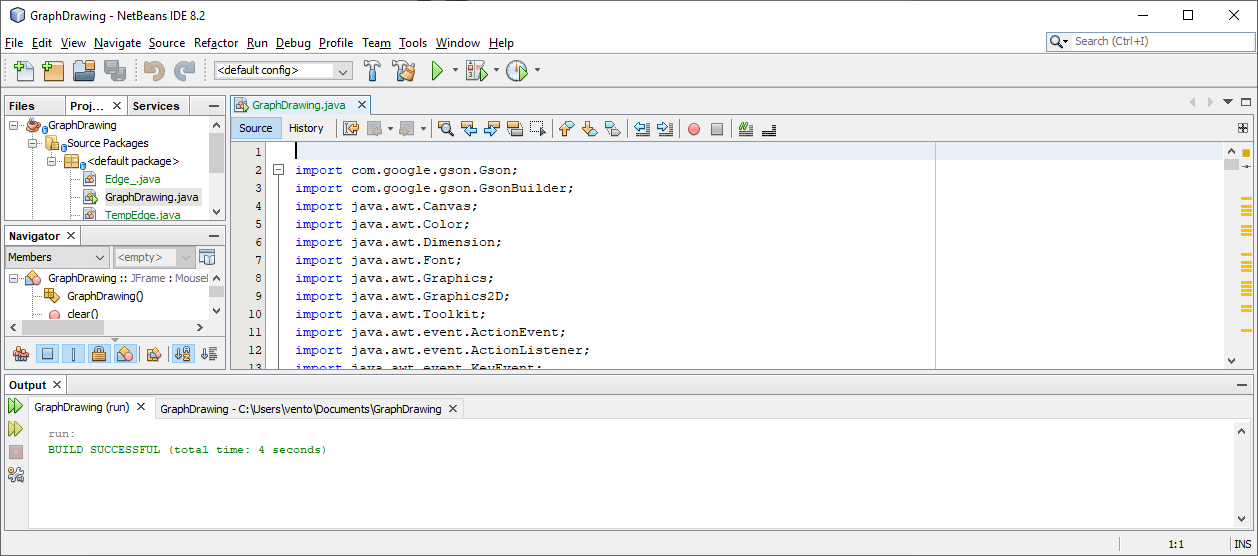
7. คลิกขวาที่ส่วนของโค้ดแล้วกด Run File (Shift + F6)



รูปที่ 7 แสดงวิธีการกดรัน

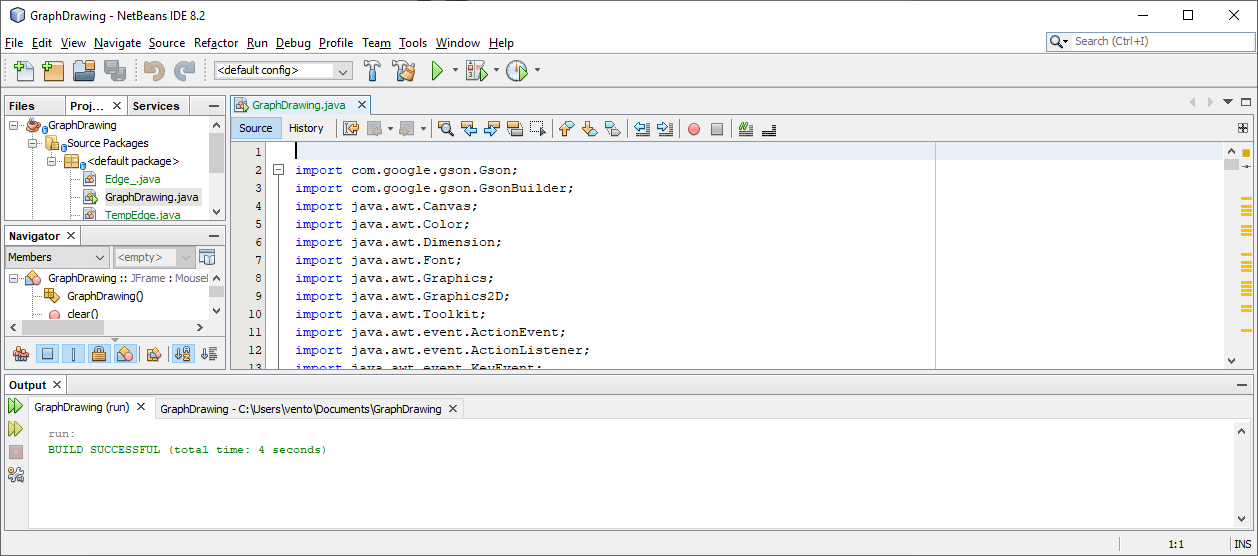
8. โปรแกรมรันหน้า GUI ขึ้นมาดังนี้

รูปที่ 8 แสดงตัวอย่างหน้า GUI

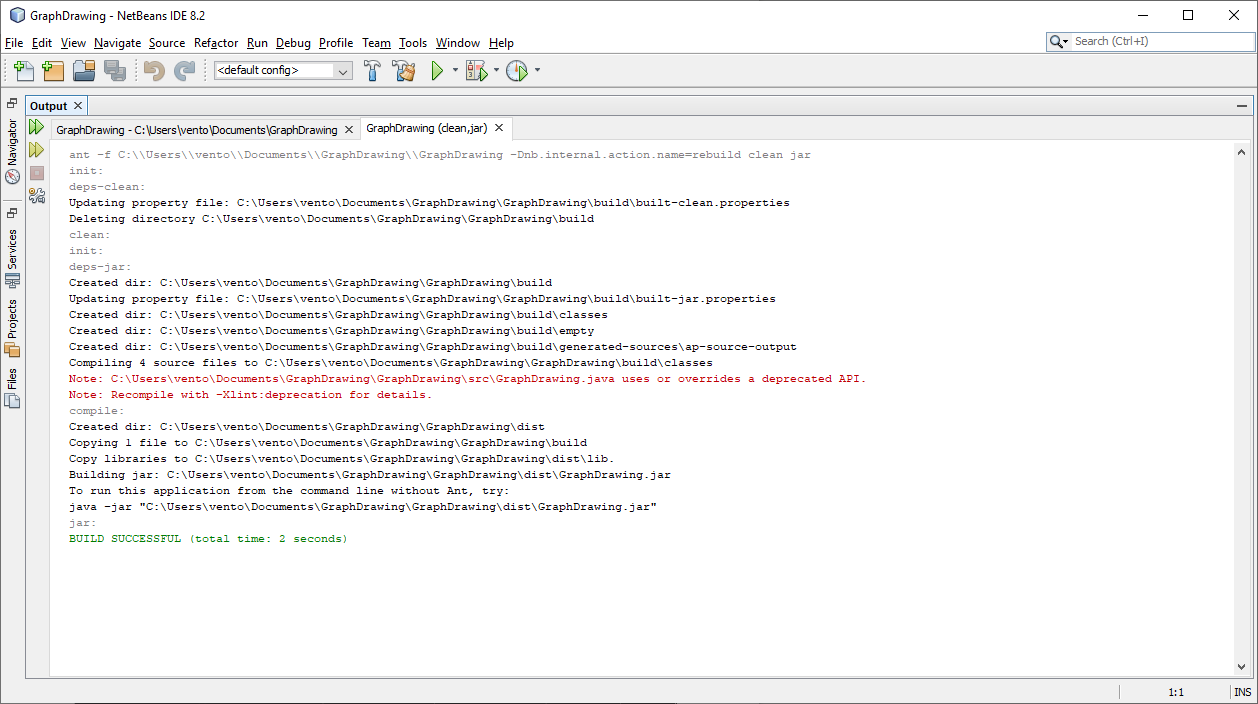


9. หากต้องการสร้างไฟล์ .jar ให้กด

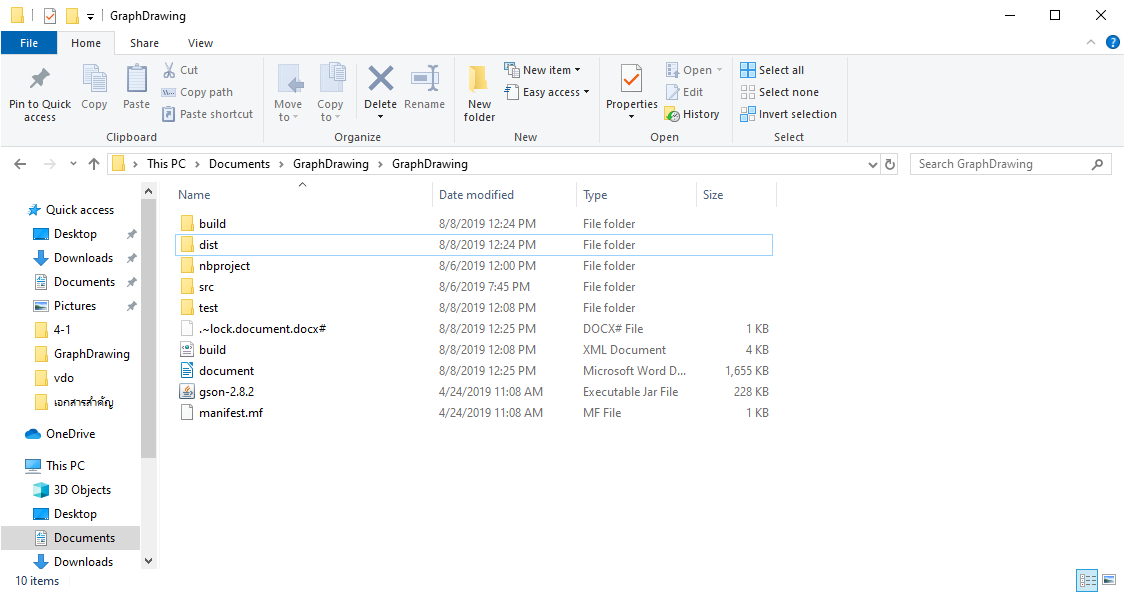
Clean and Build Project (Shift + F11)



รูปที่ 9แสดงวิธีการสร้างไฟล์ .jar

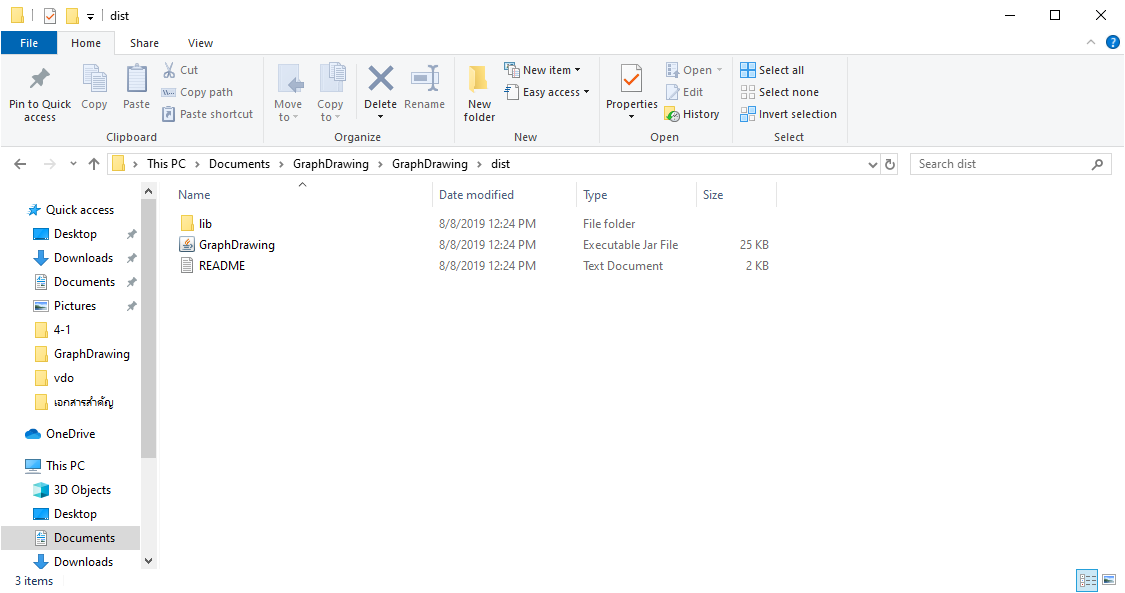
10. แสดงสถานะการ Build Project

รูปที่ 10 แสดงสถานะการสร้าง

11. หลังจากขั้นตอน 10 จะมีโฟลเดอร์ dist ปรากฏขึ้นมา

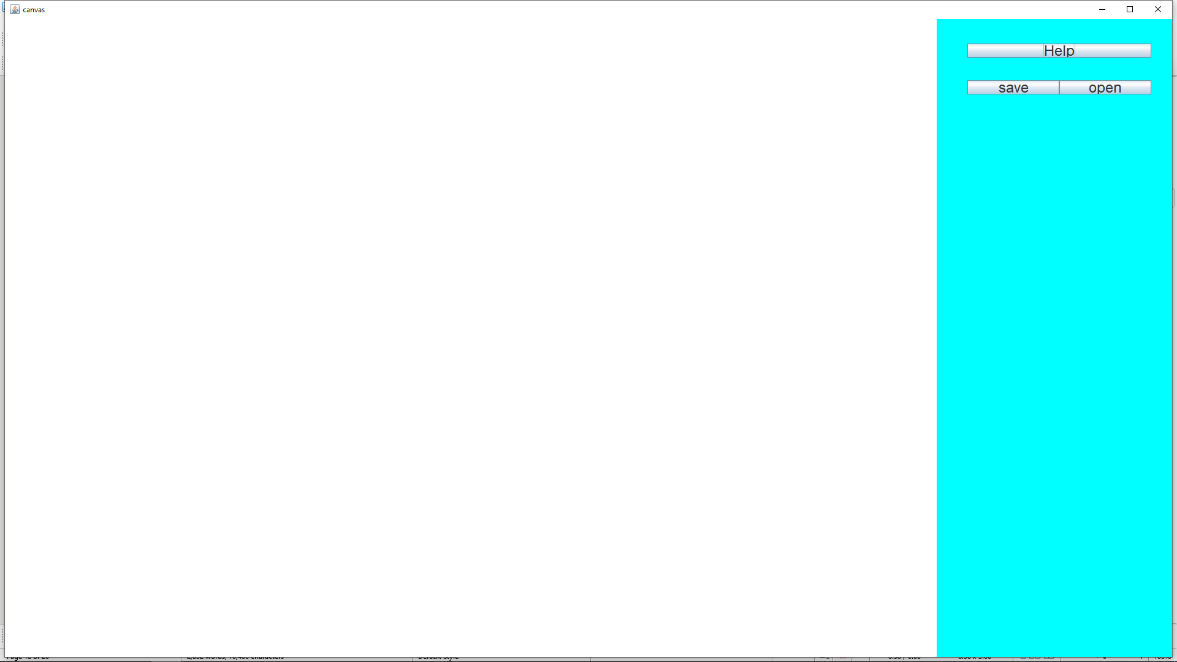
รูปที่ 11 แสดงโฟลเดอร์เมื่อสร้างสำเร็จ

12. เมื่อเข้าไปใน Folder dist ก็จะพบการโปรแกรม GraphDrawing.jar และ Folder lib ที่เก็บ library GSON สำหรับใช้ save และ open ภายในโปรแกรม



รูปที่ 12 แสดงส่วนประกอบที่ถูกสร้างในโฟลเดอร์

13. สามารถ double click ไฟล์ GraphDrawing.jar เพื่อเปิดโปรแกรมได้ (ต้องลง JDK ก่อนถึงจะสามารถ run  ไฟล์ .jar ได้)

รูปที่ 13 แสดงการเปิดโปรแกรมเพื่อรันไฟล์ .jar

**คุณสมบัติหรือความสามารถของโปรแกรม**

1. ผู้ใช้สามารถกรอกน้ำหนักแต่ละเส้นเชื่อมได้ โดยน้ำหนักที่กรอกไปนั้นจะกรอกได้เพียงแค่ตัวเลขเท่านั้น หากกรอก อักษร อักขระพิเศษ ช่องว่าง หรือ จำนวนเต็มลบ ระบบจะให้ผู้ใช้กรอกน้ำหนักใหม่ทันที
2. หากผู้ใช้เกิดความผิดพลาดในการกรอกน้ำหนัก ผู้ใช้สามารถเลือกเส้นเชื่อมดังกล่าวและกด Enter เพื่อกรอกน้ำหนักใหม่อีกครั้ง หากเส้นเชื่อมใดที่ผู้ใช้ละเลย ระบบจะกำหนดค่าน้ำหนักของเส้นเชื่อมนั้นเป็นศูนย์ทันที
3. ผู้ใช้สามารถเลือกจุดเริ่มต้นได้
4. ระบบจะแจ้งผลการคำนวณแบบทันทีในรูปแบบตารางและกราฟ

**ข้อเสียของโปรแกรม**

1. หากผู้ใช้ต้องการย้อนดูเส้นทางของกราฟก่อนหน้า หรือ คำตอบภายในตารางก่อนมีการเปลี่ยนแปลงถัดไปจะไม่สามารถทำได้ เนื่องจากตัวโปรแกรมถูกออกแบบมาในรูปของ one – way
2. หากผู้ใช้สร้างกราฟเชิงเดี่ยว,กราฟเทียม,กราฟหลายเชิง หรือมีหลาย component โปรแกรมจะไม่สามารถ run ได้

**วิธีการใช้งานของโปรแกรมและการแสดงผล**

* 1. เมื่อคลิกที่ Help ในขณะที่รันโปรแกรมจะมีหน้าต่างอธิบายการใช้งานโปรแกรม

Graphical user interface, text, application

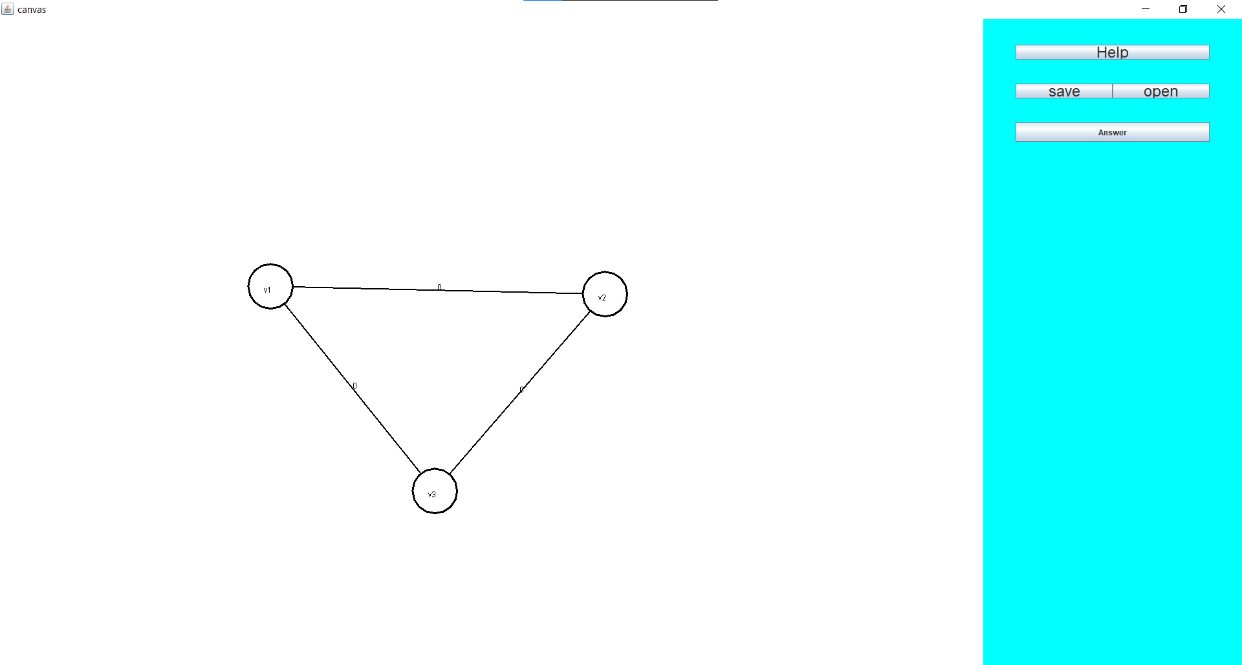
Description automatically generated

* 1. หากดับเบิ้ลคลิกจะทำการสร้างจุดยอด

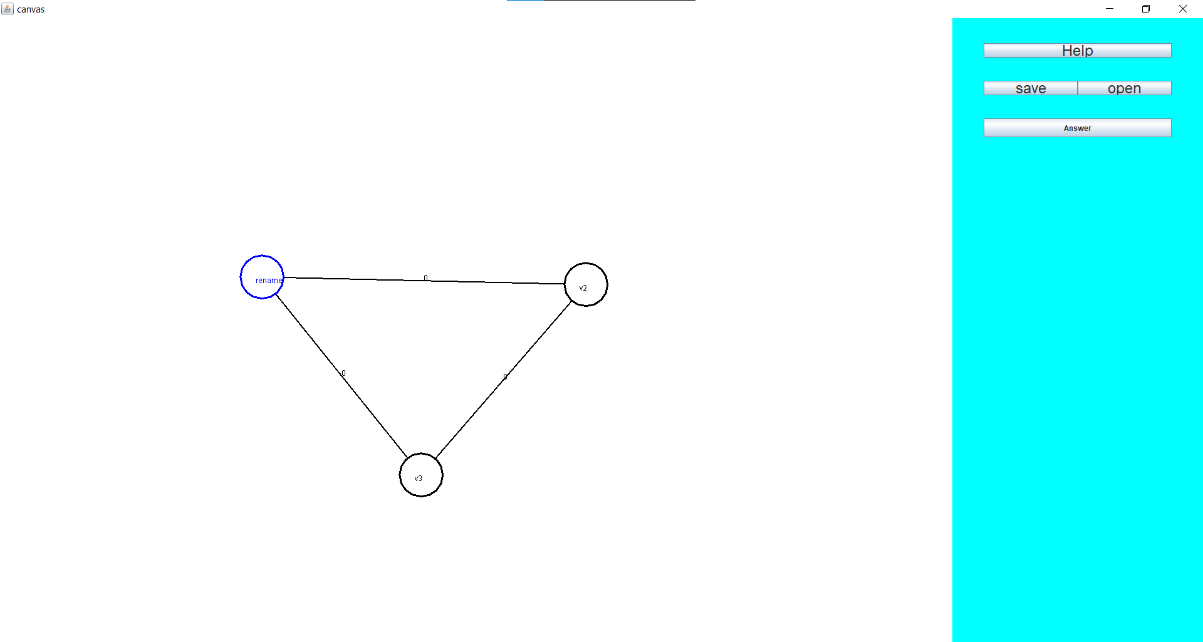
A picture containing application

Description automatically generated

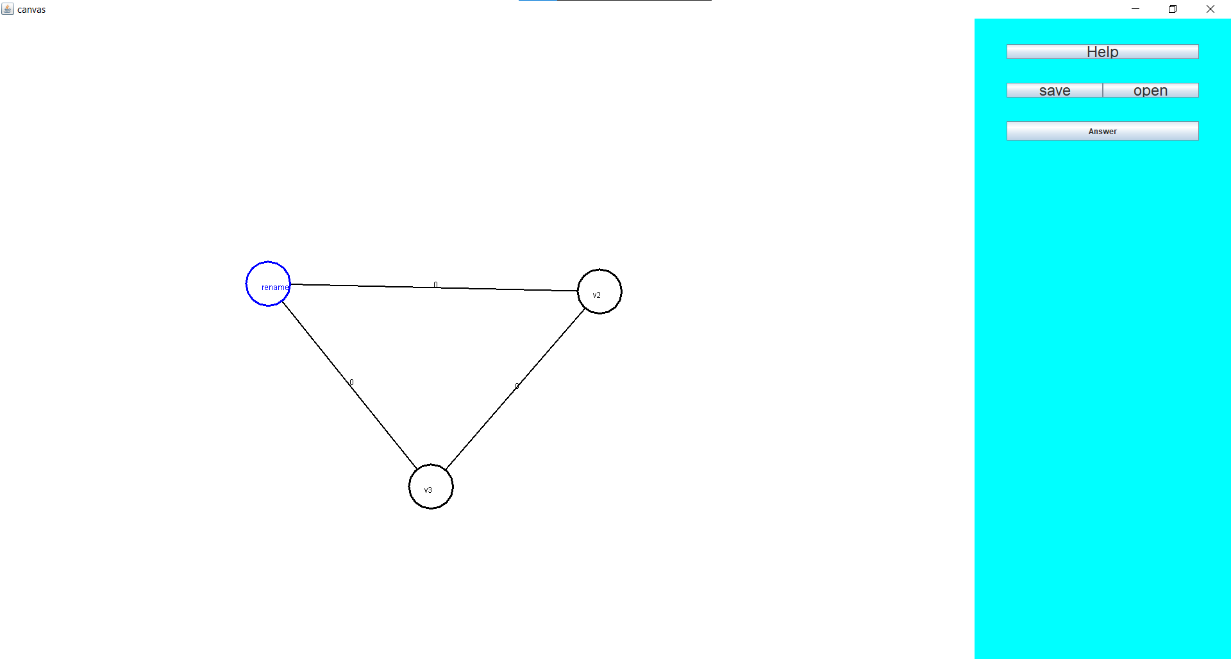
* 1. หากกดค้างที่แถบสเปซบาร์ และกดค้างที่เม้าส์จะเป็นการสร้างเส้นเชื่อม

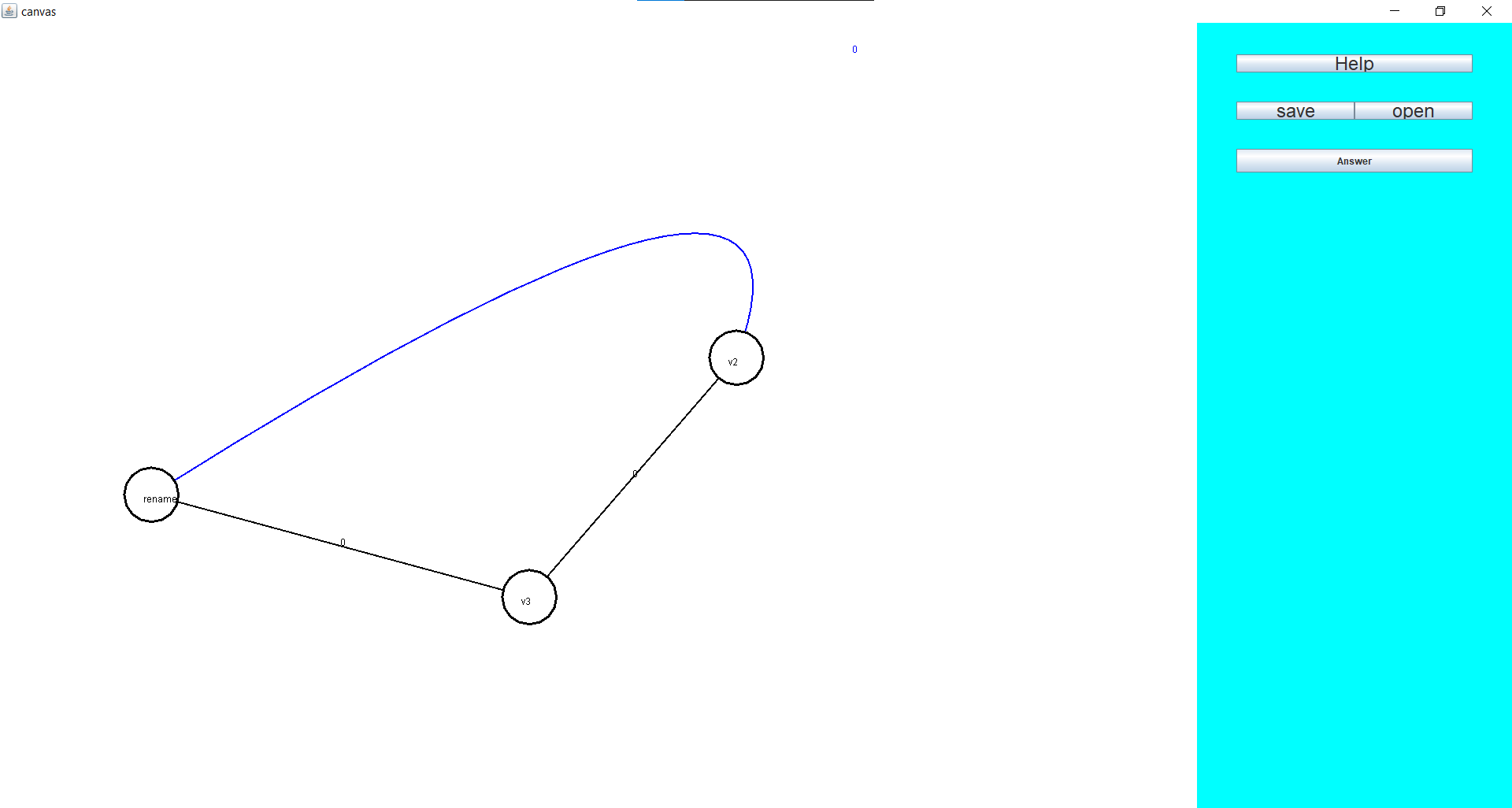


* 1. หากต้องการเปลี่ยนชื่อของจุดยอด ให้ทำการคลิกที่จุดยอดจะมีสีน้ำเงินขึ้นมา และทำการพิมพ์หรือแก้ไขชื่อที่จุดยอด

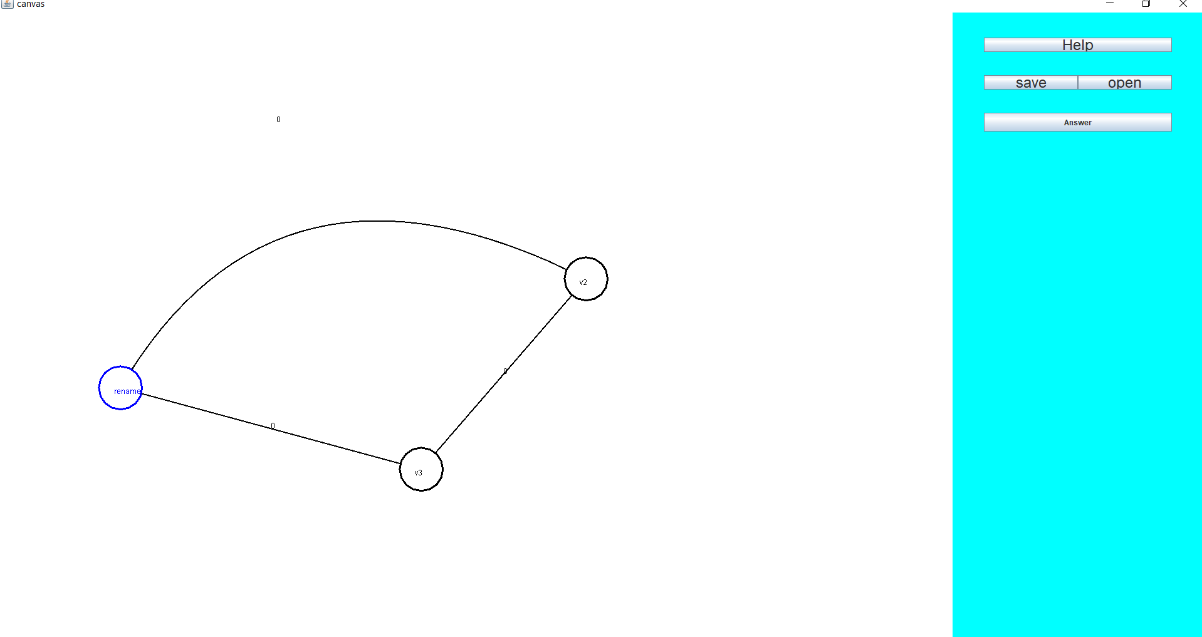


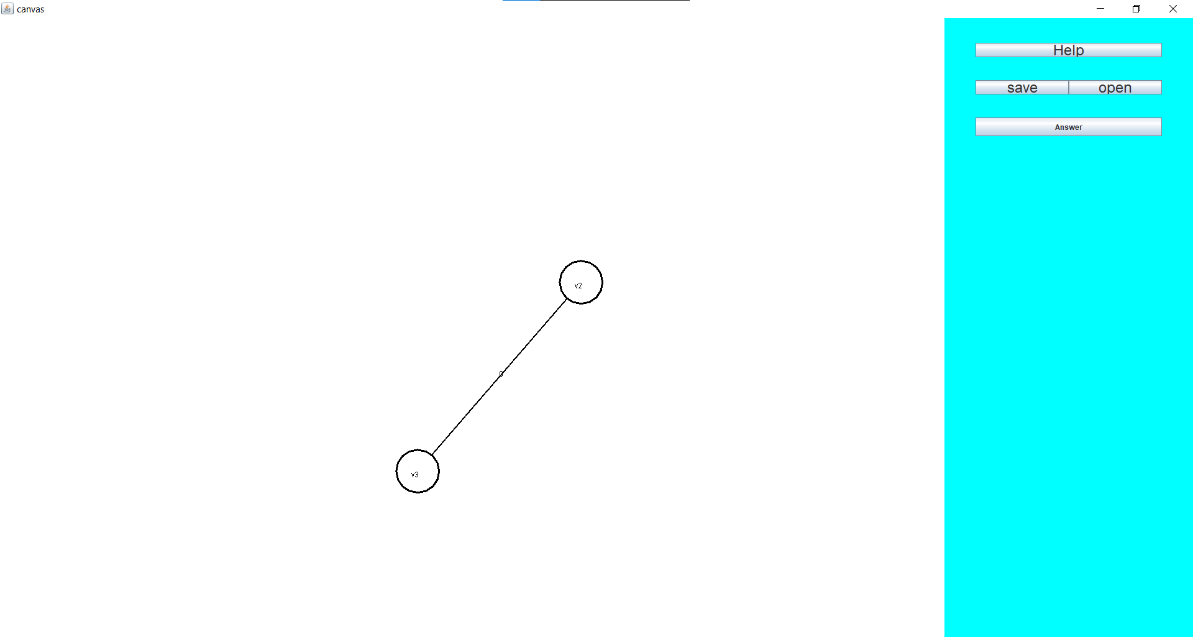
* 1. หากคลิกที่จุดยอดหรือเส้นเชื่อมและขึ้นเป็นสีน้ำเงิน จะสามารถเคลื่อนย้าย ลบ หรือแก้ไขชื่อได้



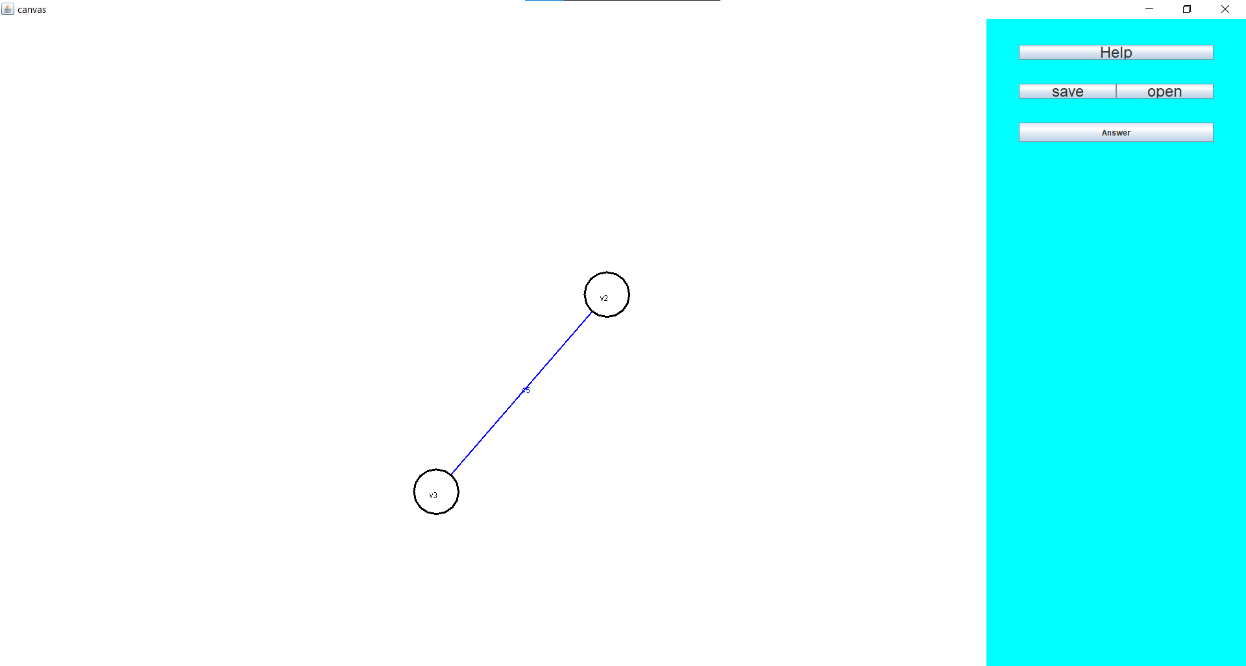


1. หากคลิกที่จุดยอด จากนั้นกดปุ่มลบ จะทำการลบจุดยอดนั้นทันที

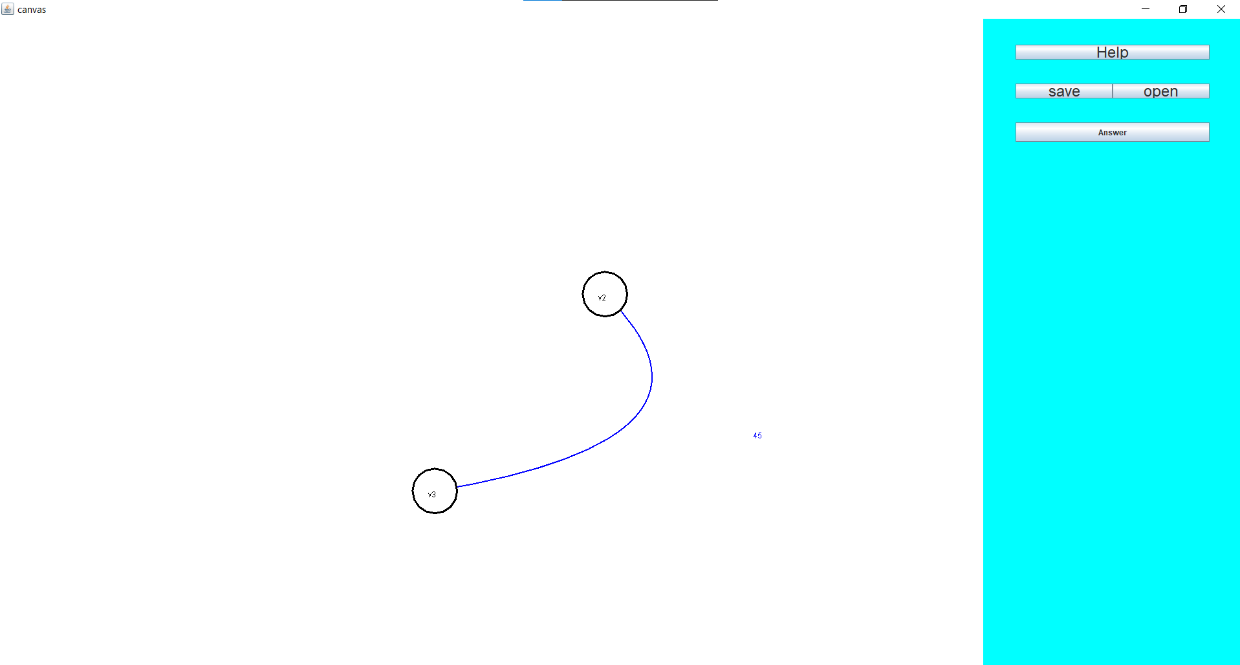




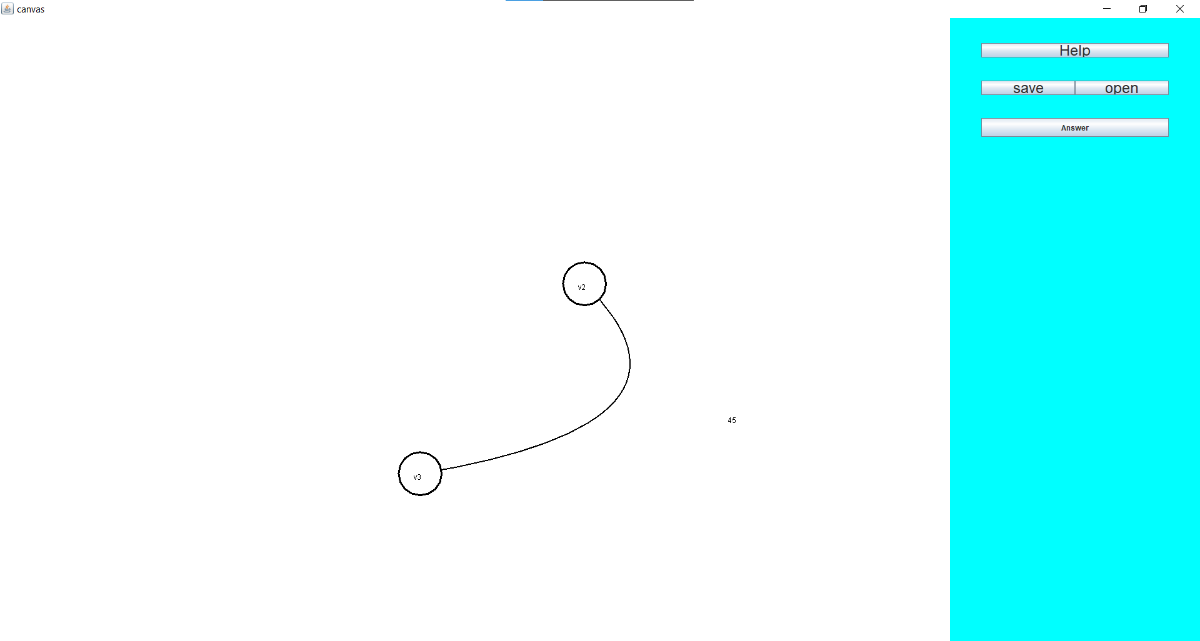
1. หากต้องการเปลี่ยนชื่อของเส้นเชื่อม ให้ทำการคลิกที่ตัวอักษรบนเส้นเชื่อมและพิมพ์เปลื่ยนชื่อของเส้นเชื่อม

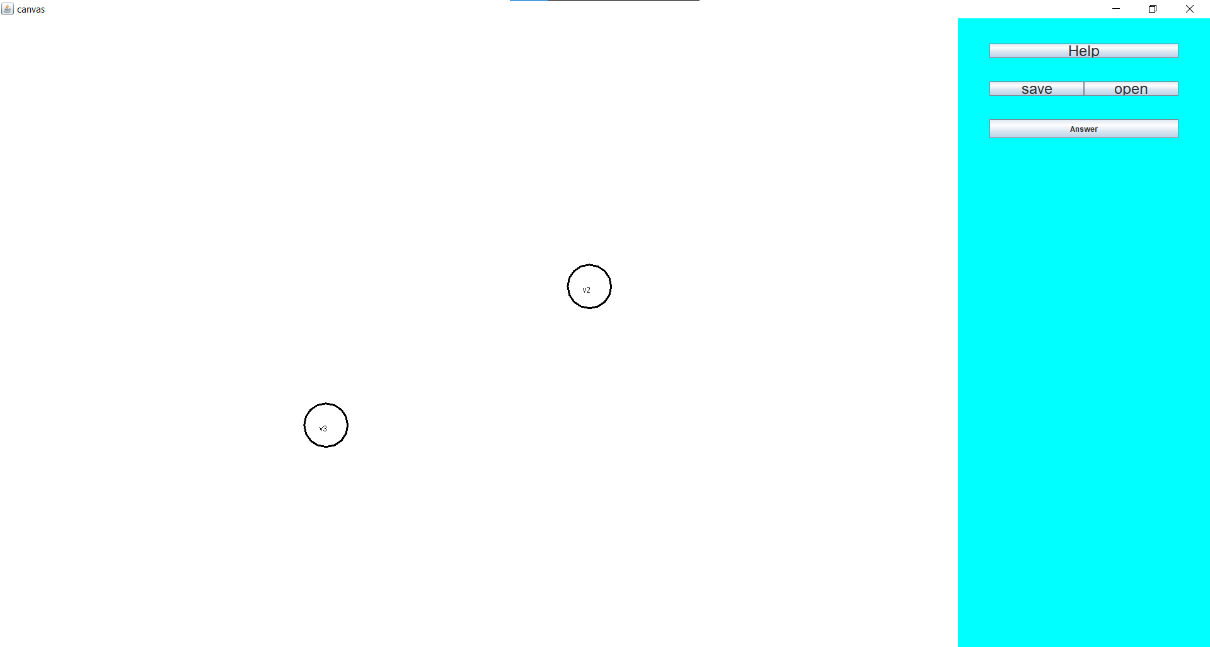


1. หากต้องการเคลื่อนย้ายเส้นเชื่อม ให้ทำการคลิกที่เส้นเชื่อมและลากเพื่อเคลื่อนย้าย

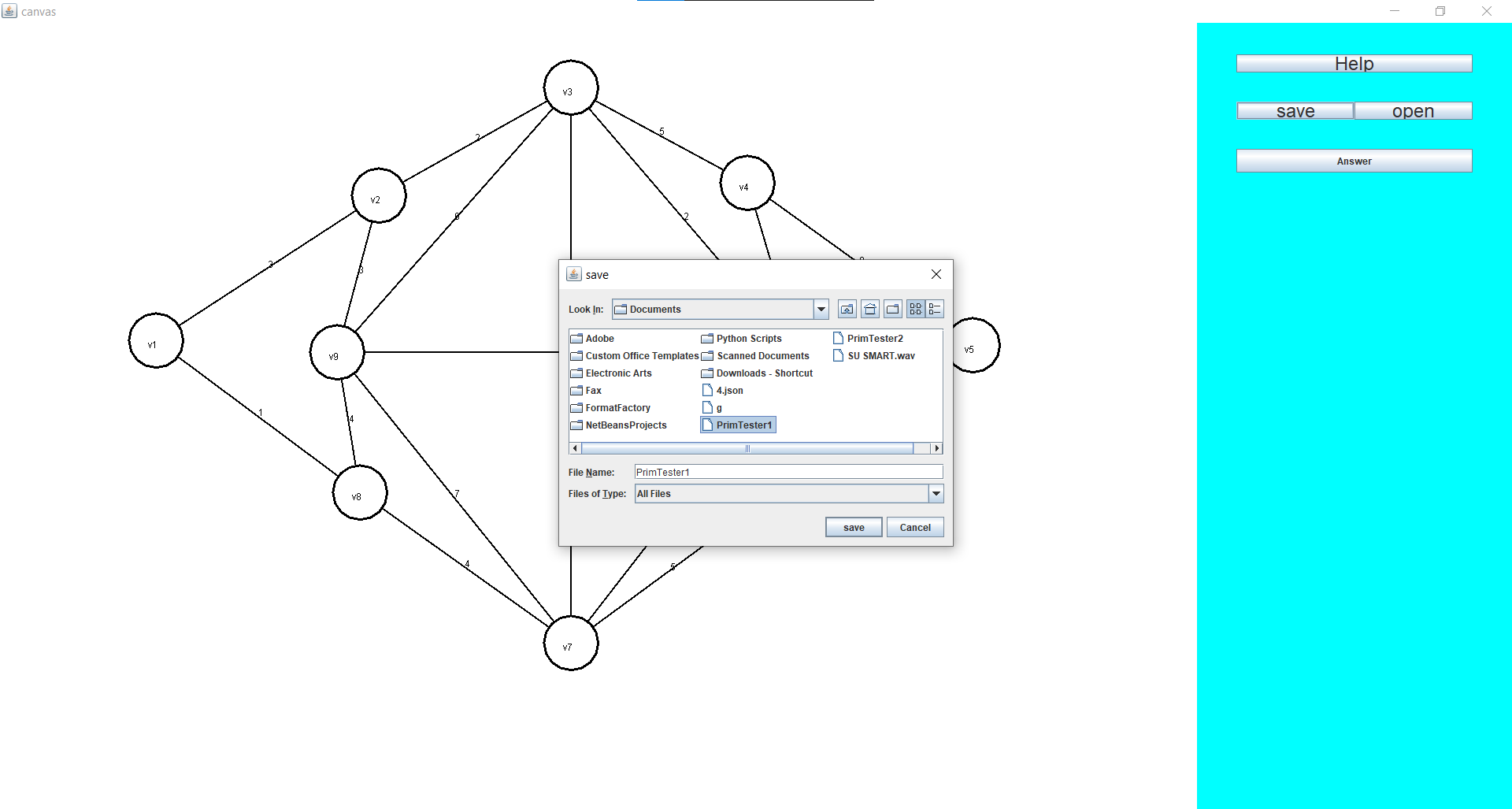


1. หากคลิกที่เส้นเชื่อมจะมีสีน้ำเงินขึ้น จากนั้นกดปุ่มลบ จะทำการลบเส้นเชื่อมนั้นทันที

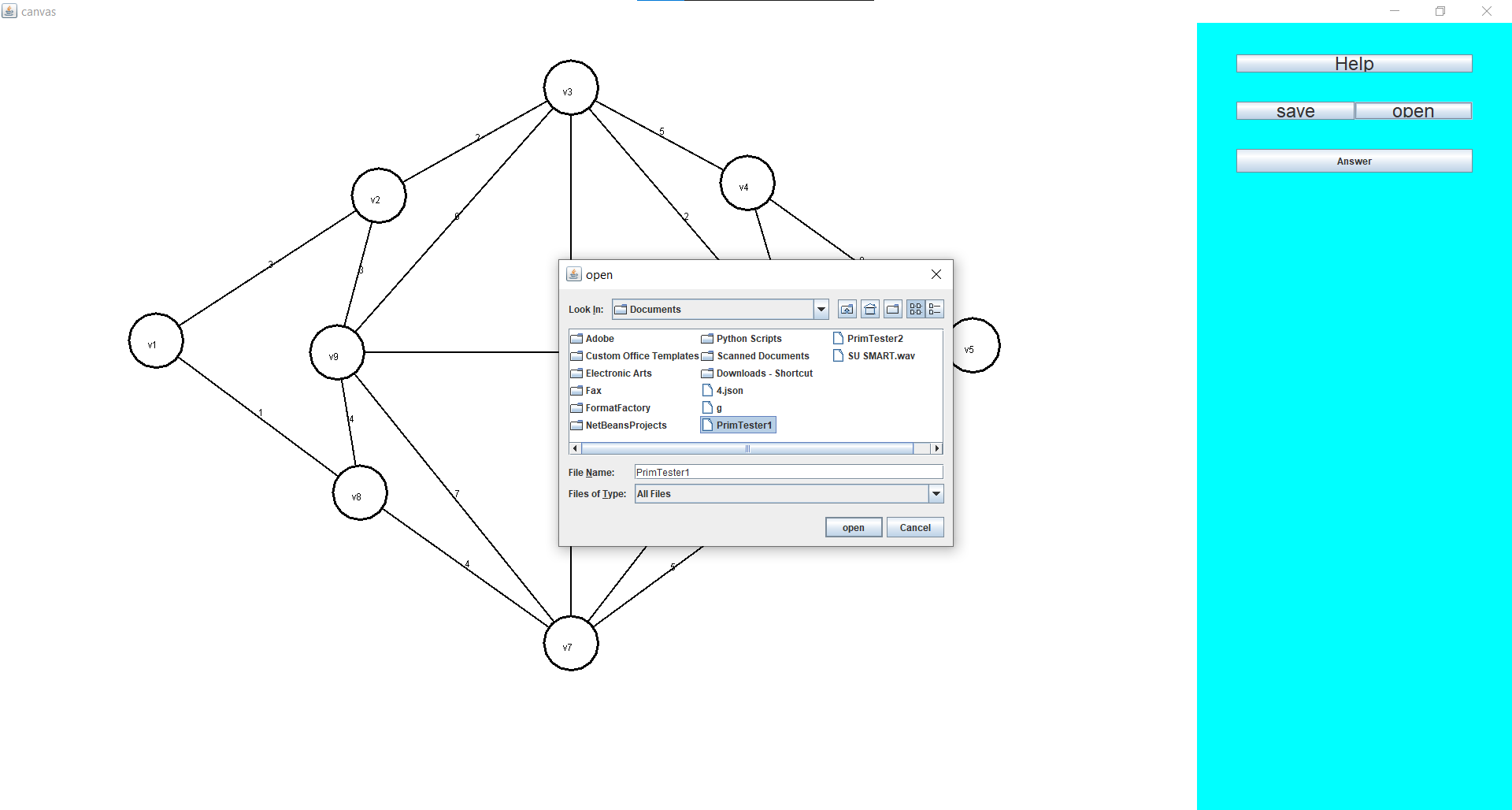




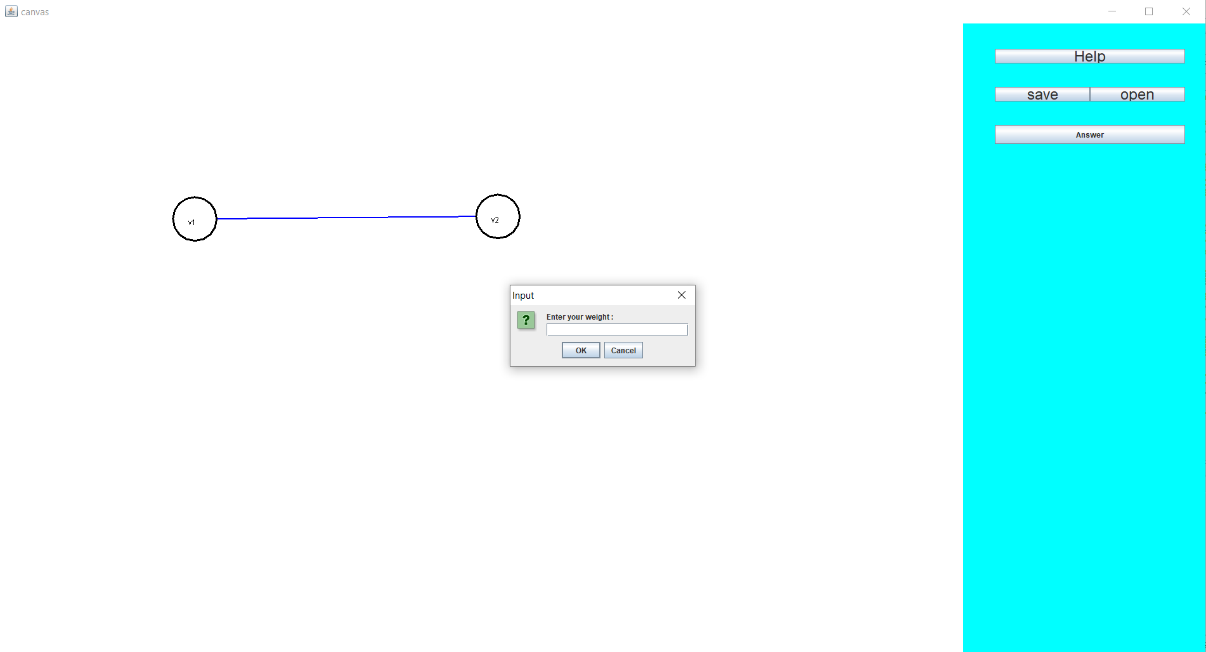
1. หากกดปุ่ม save ระบบจะทำการใช้ Gson ในการแปลง object Backup เป็น json และบันทึกไฟล์ลงในเครื่องตามตำแหน่งที่กำหนด

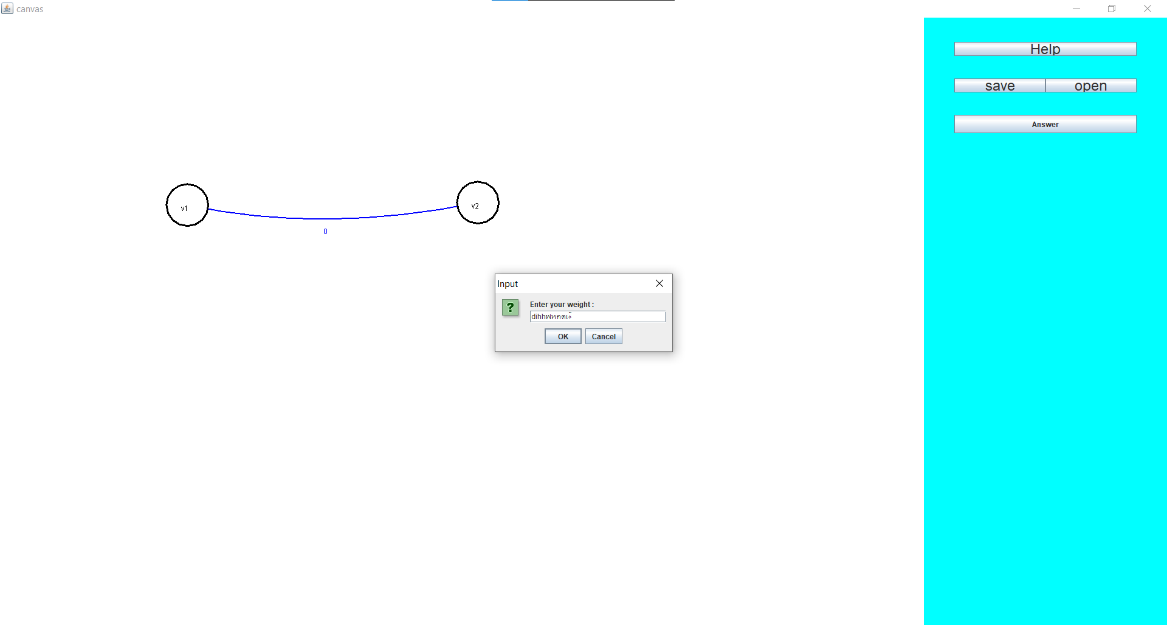


1. หากกดปุ่ม open ระบบจะทำการอ่านไฟล์กราฟ แล้วใช้ Gson ในการแปลง json เป็น object Backup และแสดงกราฟบนหน้าจอ

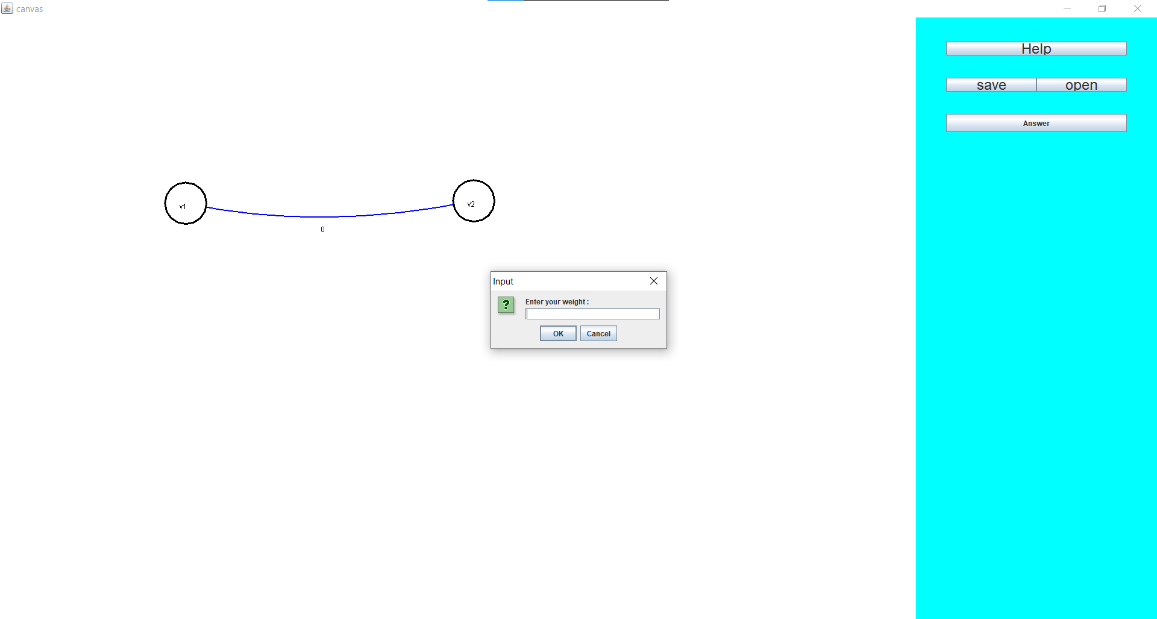


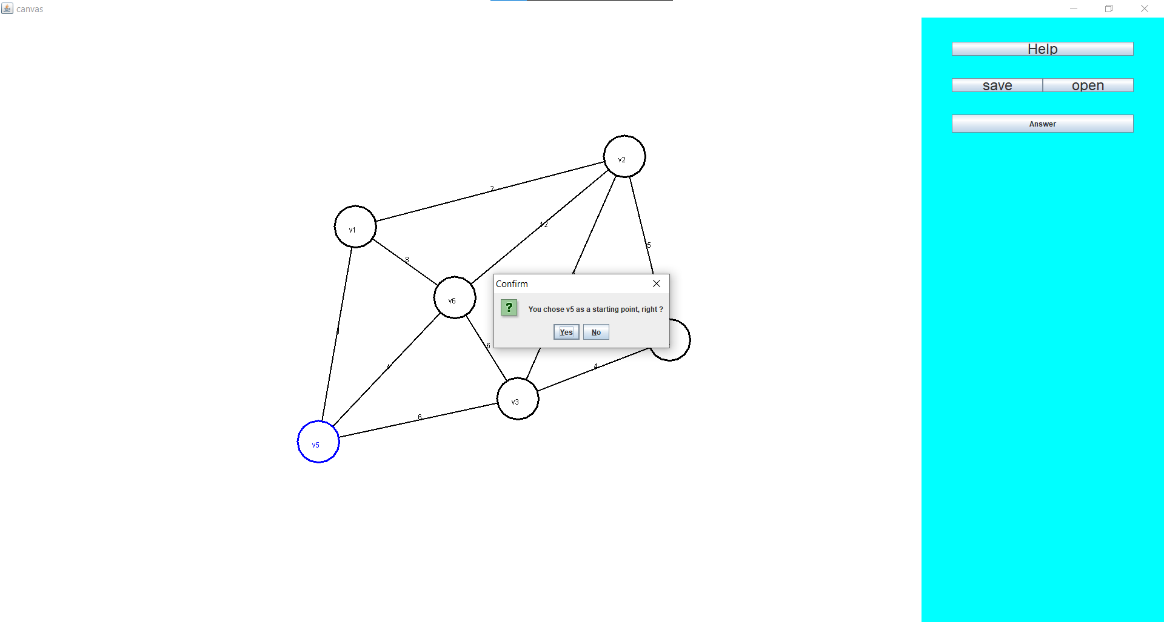
1. หากสร้างเส้นเชื่อมระหว่างจุดยอดใดๆแล้ว จะให้กรอกน้ำหนักทันที โดยสามารถกรอกได้เพียงตัวเลขเท่านั้น และขณะกรอกหากมีช่องว่าง อักขระ หรือจำนวนเต็มลบ ระบบจะให้ผู้ใช้ใส่ค่าน้ำหนักใหม่





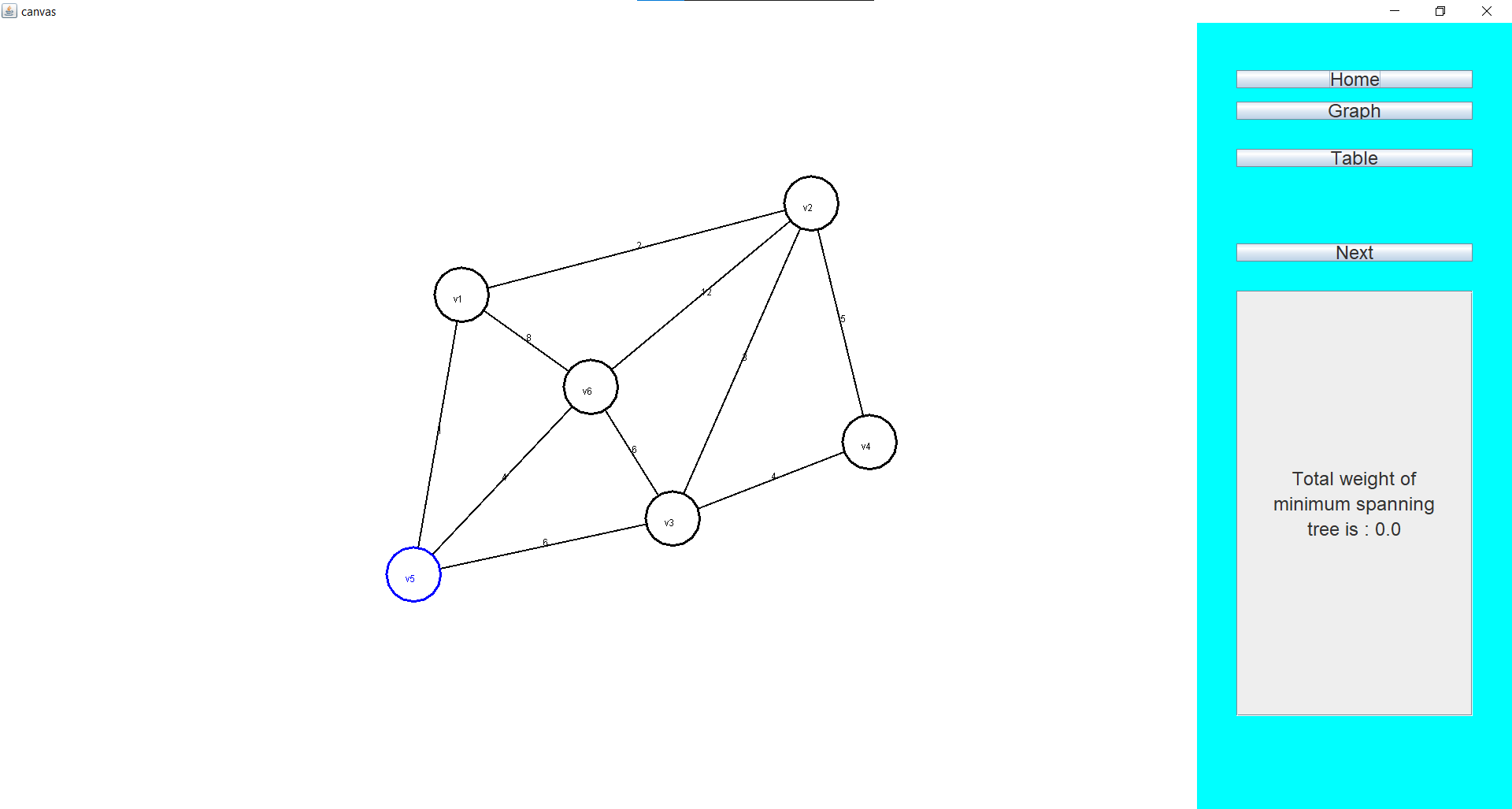
1. หากผู้ใช้ไม่ใส่ค่าน้ำหนัก ผู้ใช้สามารถกดที่เส้นเชื่อม และกด Enter ระบบจะแสดง frame ขึ้นมาใหม่ เพื่อให้ผู้ใช้กรอกน้ำหนักได้อีกครั้ง



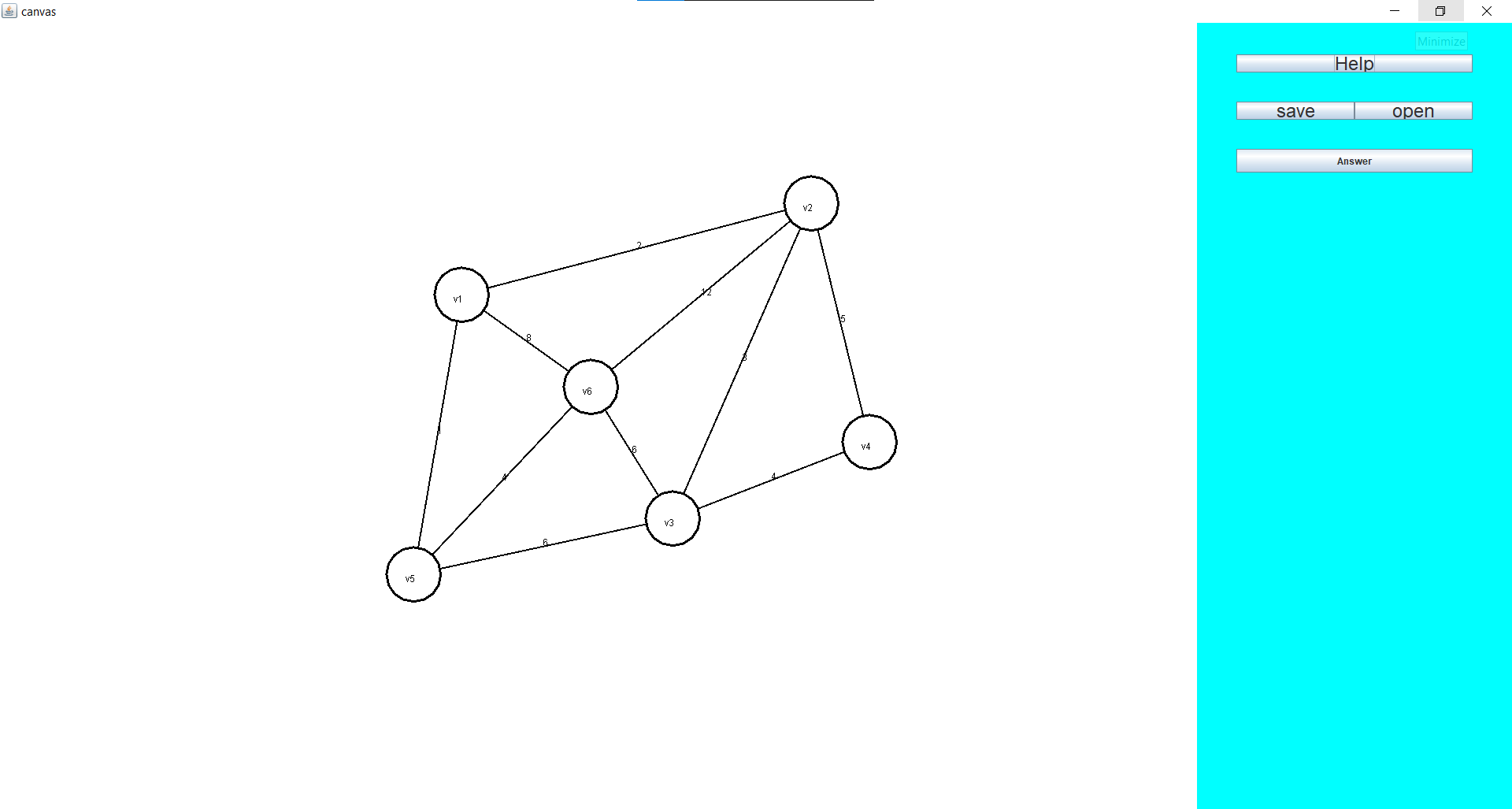
1. เมื่อผู้ใช้สร้างกราฟเสร็จแล้วนั้น ผู้เลือกสามารถเลือกจุดเริ่มต้นได้ โดยผู้ใช้สามารถเลือกจุดยอดให้เป็นสีน้ำเงิน จากนั้นกด Enter จะมี frame ขึ้นมาสอบถามผู้ใช้ว่า เลือกจุดยอดนี้เป็นจุดเริ่มต้นใช่หรือไม่

1. เมื่อกดปุ่ม answer จะแสดง frame ขึ้นมาใหม่ ซึ่งจะประกอบด้วย

ปุ่ม home , graph , table , next

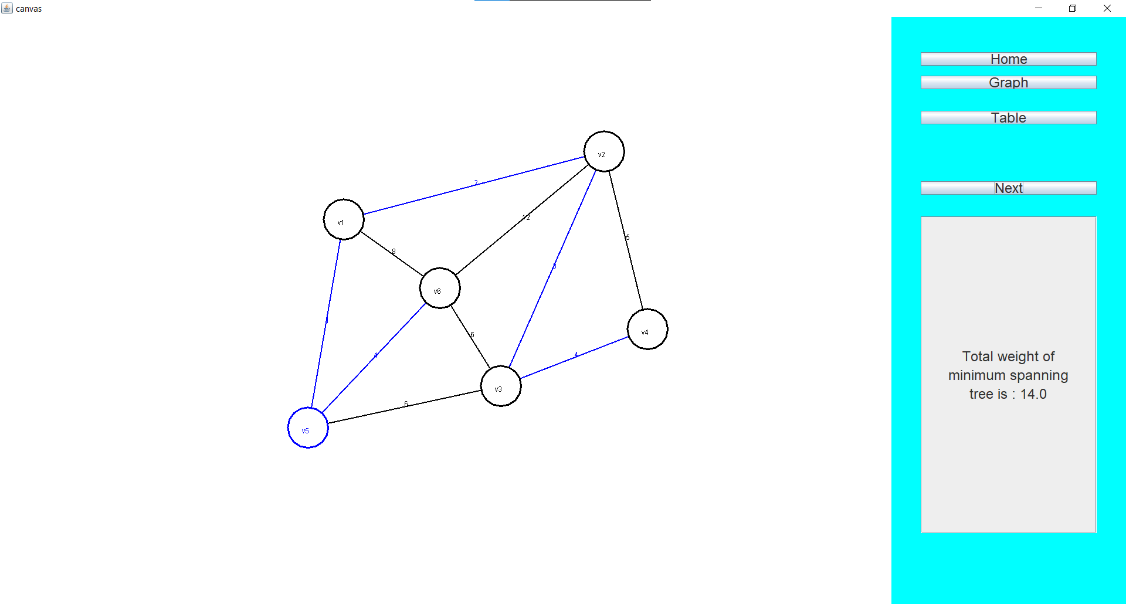


1. เมื่อกดปุ่ม home ระบบจะแสดงกราฟตั้งต้น

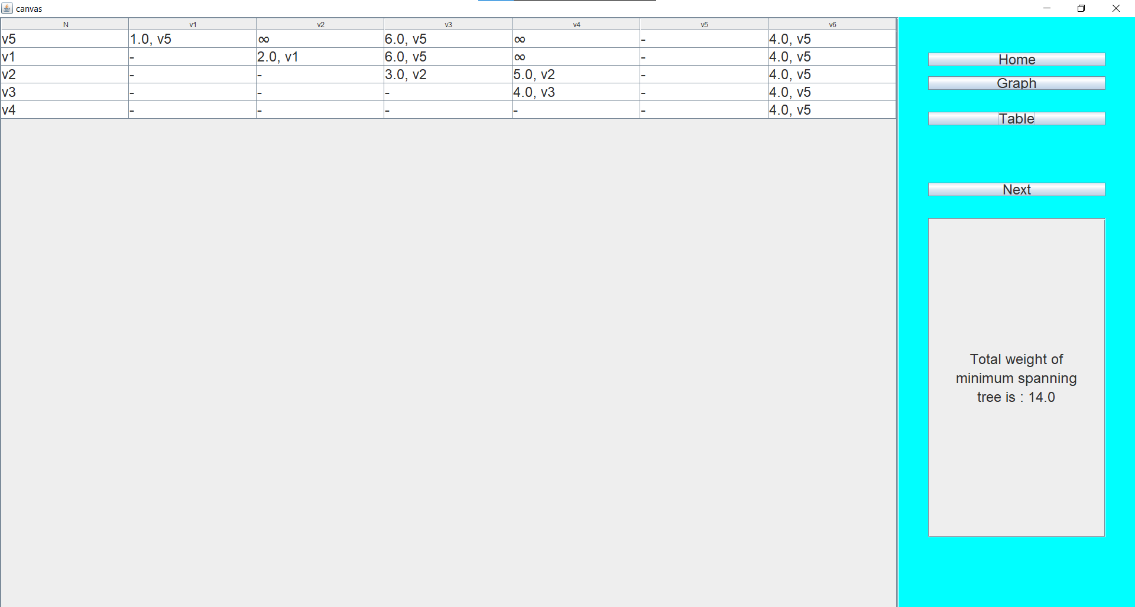


1. เมื่อกดปุ่ม graph ระบบจะแสดงผลเป็นรูปภาพของกราฟที่ผ่านการประมวลผล

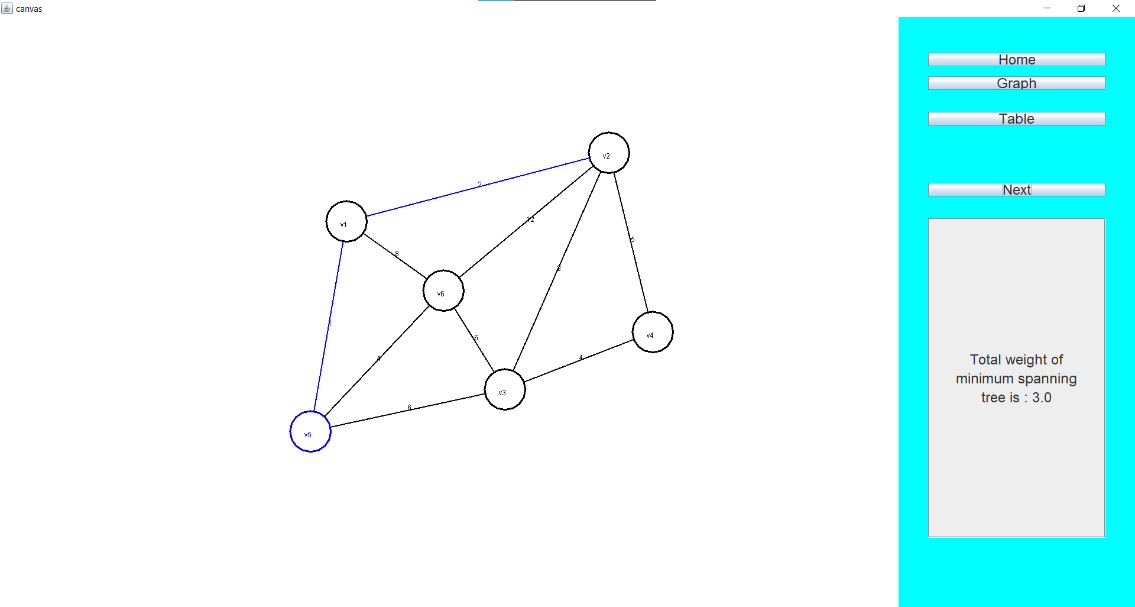
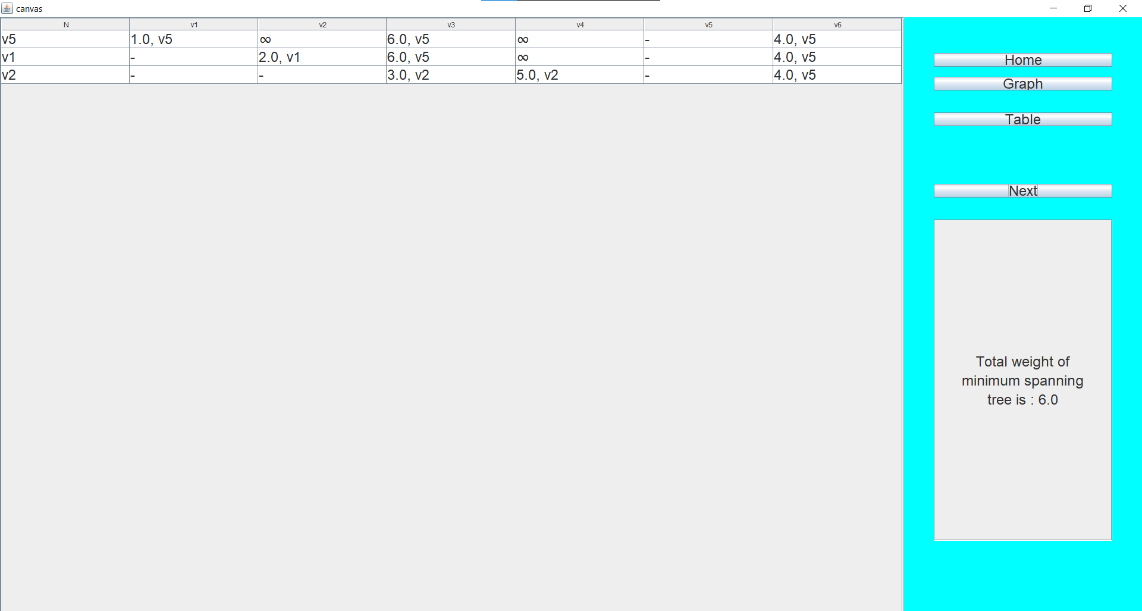
ตามวิธีการของ Prim’s Algorithm



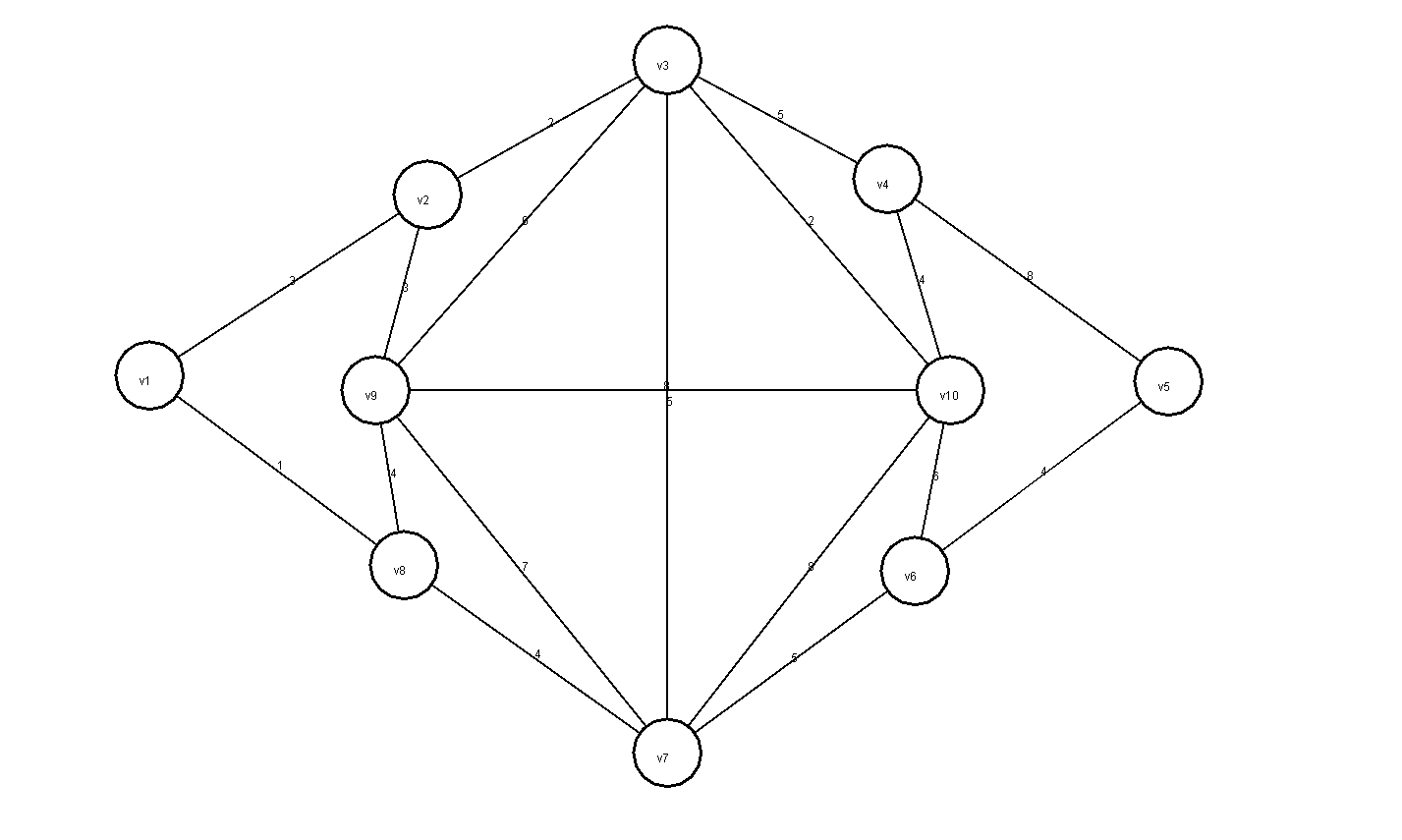
1. เมื่อกดปุ่ม table ระบบจะแสดงผลเป็นตารางคำตอบของกราฟที่ผ่านการประมวลผลตามวิธีการของ Prim’s Algorithm



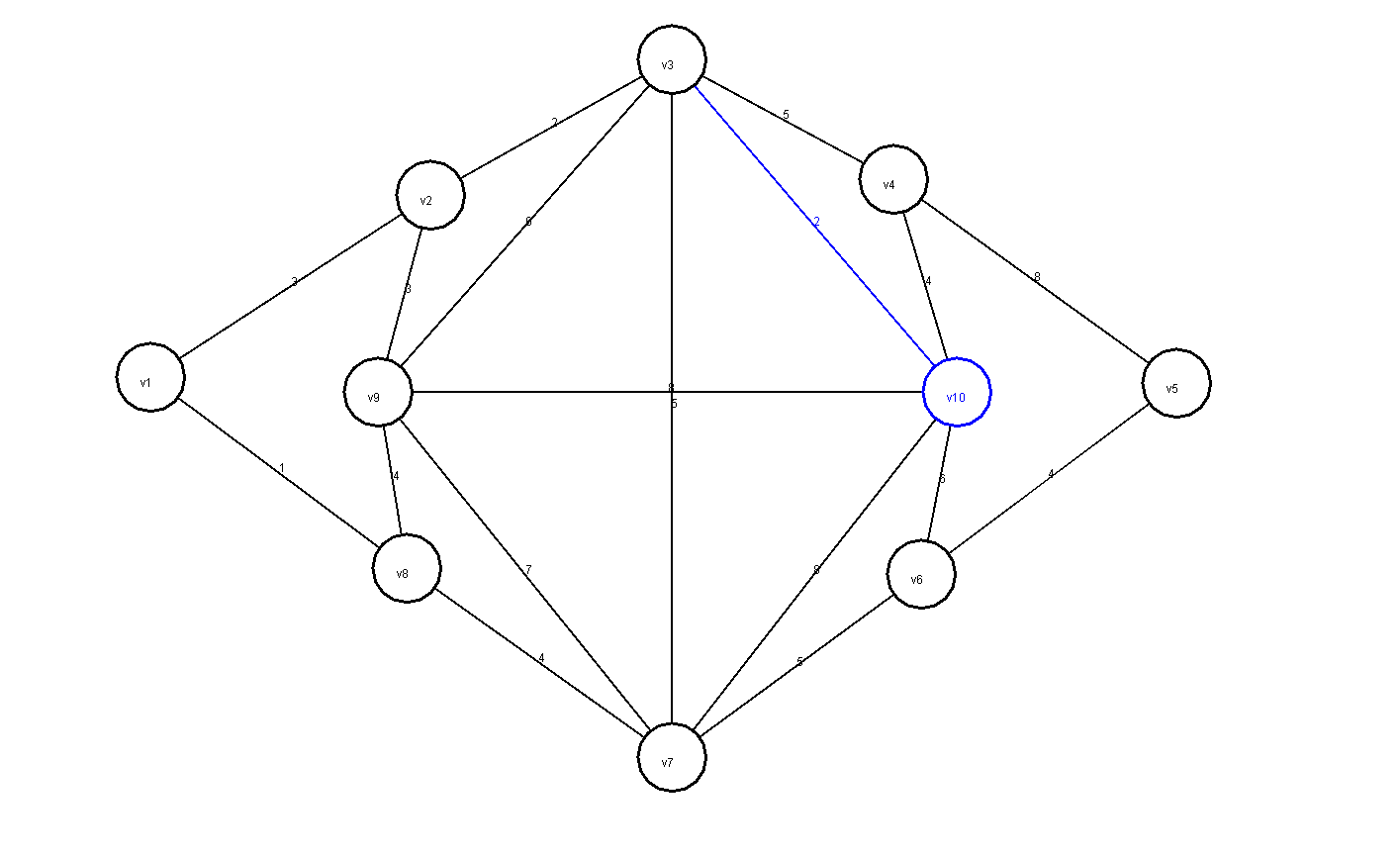
1. เมื่อกดปุ่ม next ระบบจะทำการแสดงเส้นเชื่อมที่ถูกเลือก หรือ ตารางถัดไปที่ผ่านการประมวลผลตามวิธีการของ Prim’s Algorithm



**ตัวอย่างทดสอบแบบที่ 1**

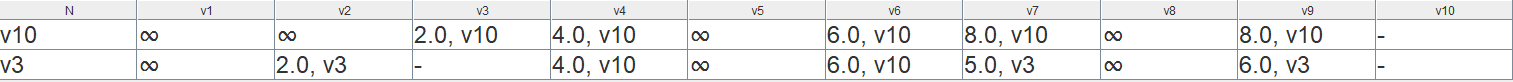
****

ให้จุดเริ่มต้นจากจุดยอด v10 ต่อไปก็ให้ค่าเริ่มต้นกับข้อมูลที่จำเป็น ให้ N เป็น {v10} ให้ T เป็นเซตว่างและ A เป็น {v1,v2,v3,v4,v5,v6,v7,v8,v9} ดังนั้นจึงหาเส้นเชื่อมที่มีจุดปลายใน A ที่มาตกกระทบกับ v10 ซึ่งเก็บได้เป็นตารางดังนี้

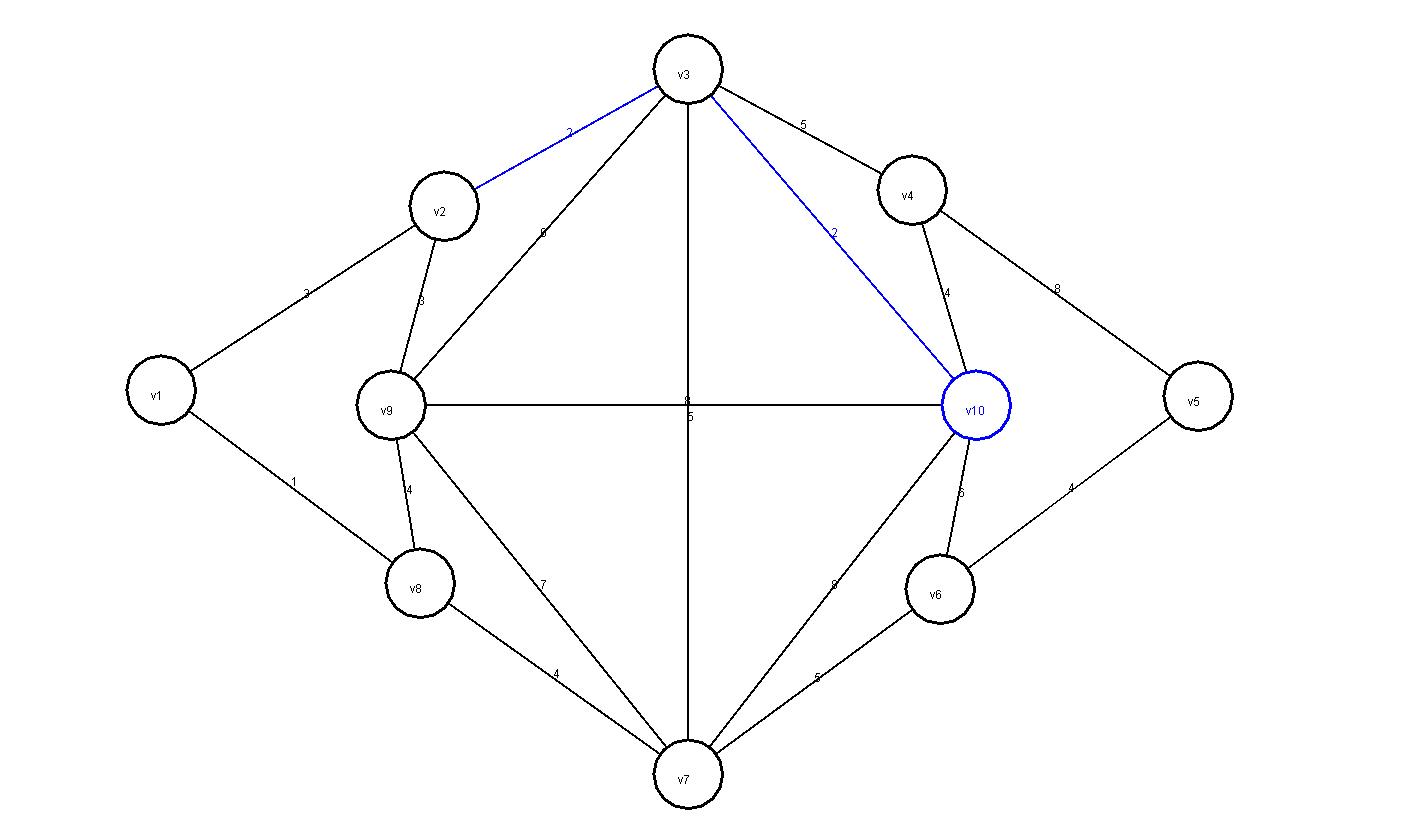
****

****

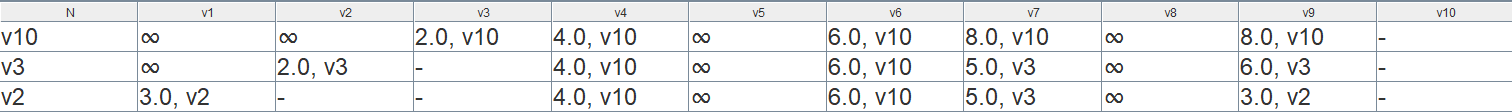
รอบที่ 2 จะมีการปรับค่าตำแหน่งน้อยที่สุดระหว่างจุดที่อยู่ในเซต A และเซต N เนื่องจากมีจุดใหม่ v3 ในเซต N ขณะนี้ก็จะเชื่อมเข้ามาผ่านจุดยอด v3 จุดยอด v7 และ v9 เมื่อเชื่อมทางจุดยอด v3 ก็จะมีน้ำหนักลดลง แต่จุดยอด v4 เชื่อมกับ v3 น้ำหนักไม่ลดลงก็จะไม่เปลี่ยนค่าในตาราง ซึ่งพิจารณาได้ในตารางต่อไปนี้



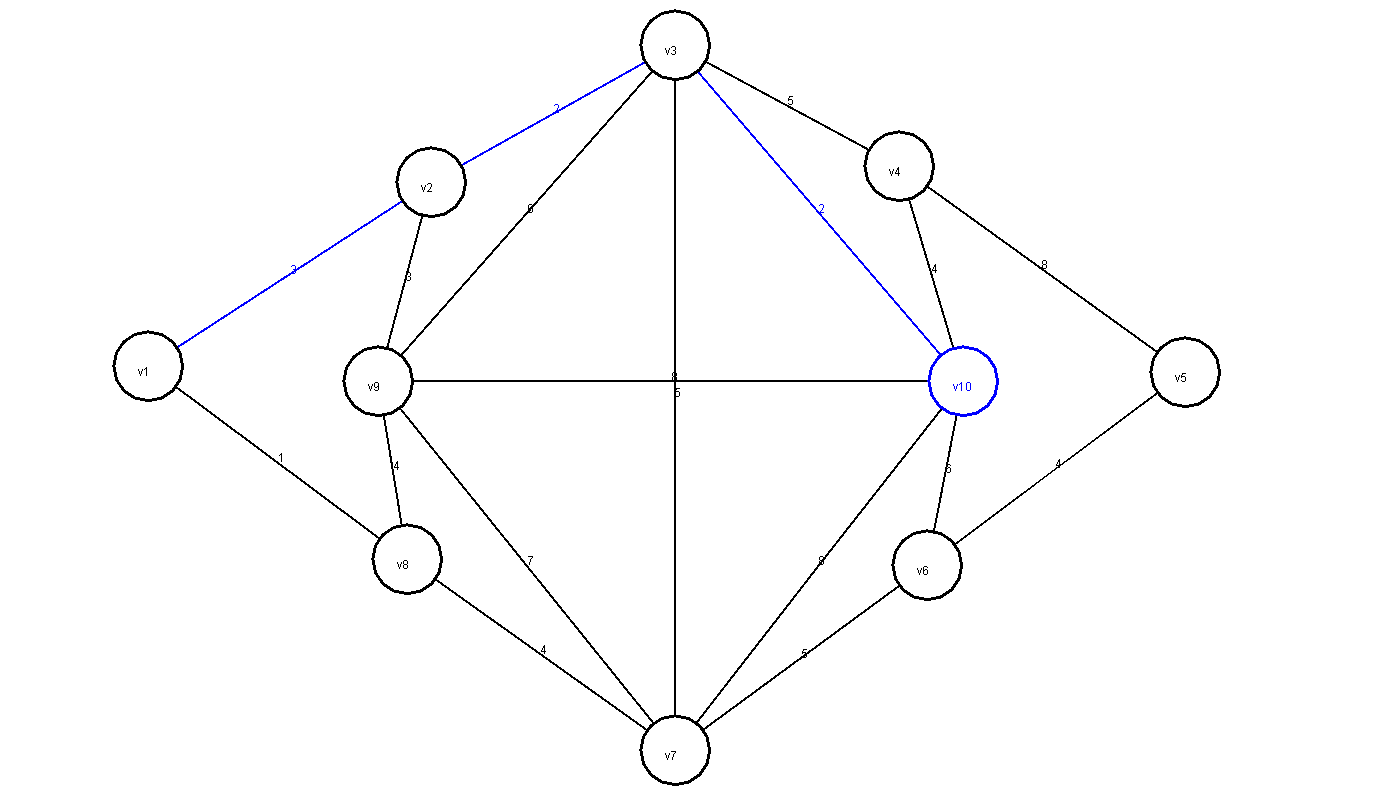
จากนั้นจึงหาเส้นเชื่อมที่มีจุดยอดจุดหนึ่งใน A มาตกกระทบกับ กับจุดยอดใน N ซึ่งคือ v3 และ v10 โดยต้องเป็นเส้นเชื่อมที่มีน้ำหนักที่น้อยที่สุด ซึ่งก็คือ v2v3 จากนั้นก็ปรับข้อมูลโดยเพิ่ม v2 ลงใน N พร้อมกันนั้นก็ลบ v2 ออกจาก A และต้นไม้แบบทอดข้าม T เป็น {v3v10 , v2v3} แสดงได้ดังรูปต่อไปนี้



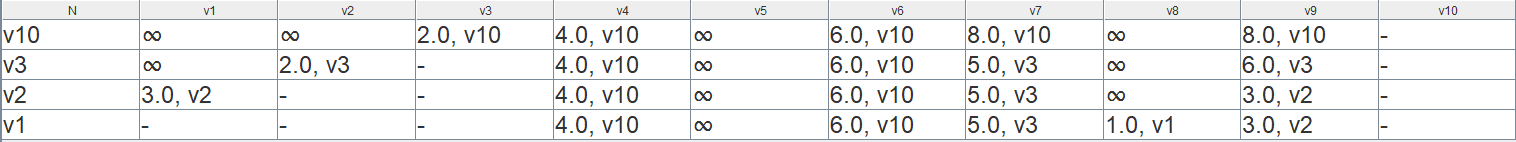
รอบที่ 3 เมื่อนำจุดยอด v2 ใส่ใน N แล้วจะต้องมีการปรับเปลี่ยนค่าน้ำหนักของการเชื่อมระหว่างเซต A และ N ใหม่ดังนี้



จากนั้นหาเส้นเชื่อมที่มีน้ำหนักน้อยที่สุดต่อไปก็คือ v1v2 หรือ v2v9 ในกรณีเลือกเส้นเชื่อม v1v2 ก่อน ใส่เข้าไปในต้นไม้แบบทอดข้าม T เป็น {v3v10 , v2v3 , v1v2} เพิ่มจุดยอด v1 เข้าไปใส่ใน N และลบจุดยอดนี้ออกจาก A จะได้



รอบที่ 4 จากการนำ v1 ใส่ใน N จะต้องมีการปรับค่าน้ำหนักใหม่ที่จุดยอด v8 จะเชื่อมมาได้ผ่าน v1 จะได้ตารางดังต่อไปนี้

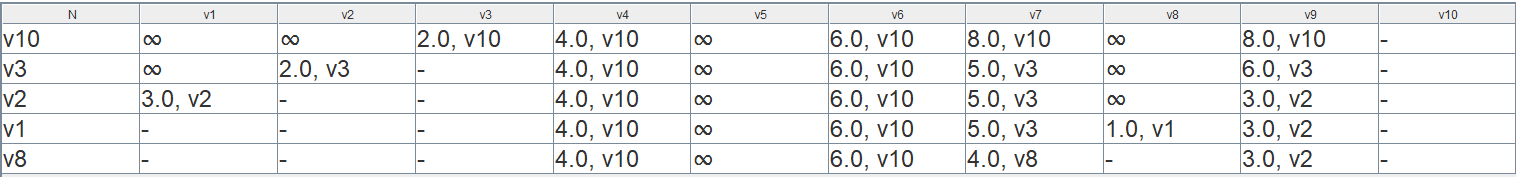


จากนั้นหาเส้นเชื่อมที่มีน้ำหนักน้อยที่สุดต่อไปก็คือ v8v1 ย้ายจุดยอด v8 จาก A ไป N จะได้ว่า A เป็น {v4 ,v5 , v6 , v7 , v9 } ในขณะที่ T เป็น {v3v10 , v2v3 , v1v2, v8v1} ต้นไม้จะเป็นดังรูป

A picture containing map, necklace, sitting, table

Description automatically generated

รอบที่ 5 เนื่องจากนำจุดยอด v8 ใส่ใน N ดังนั้นต้องมีการปรับค่าน้ำหนักในตาราง โดยที่การเชื่อมต่อของจุดยอด v7 จะมีค่าลดลงเป็น 4 จะได้ตารางดังต่อไปนี้



จากนั้นเลือกเส้นเชื่อม v2v9 เนื่องจากมีน้ำหนักน้อยที่สุด ดังนั้นต้นไม้แบบทอดข้าม T ก็จะเป็น {v3v10 , v2v3 , v1v2, v8v1, v2v9} ส่วน A เป็น {v4 ,v5 , v6 , v7 } และ N เป็น {v10 ,v3 , v2 , v1 , v8 , v9 }

Diagram

Description automatically generated

รอบที่ 6 เนื่องจากนำจุดยอด v9 ใส่ใน N โดยพบว่าไม่มีการปรับค่าน้ำหนักดังตารางต่อไปนี้

Graphical user interface, table

Description automatically generated

จะพบว่ามีเส้นเชื่อมจำนวน 2 เส้นที่มีน้ำหนักเป็น 4 คือเส้นเชื่อม v4v10 และเส้นเชื่อม v7v8 ดังนั้นเลือกเส้นเชื่อม v4v10 ที่สุด ดังนั้นต้นไม้แบบทอดข้าม T ก็จะเป็น {v3v10 , v2v3 , v1v2, v8v1, v2v9, v4v10} ส่วน A เป็น {v5 , v6 , v7 } และ N เป็น {v10 ,v3 , v2 , v1 , v8 , v9 , v4} ซึ่งแสดงเป็นต้นไม้แบบทอดข้ามได้ดังนี้

Diagram

Description automatically generated

รอบที่ 7 เนื่องจากนำจุดยอด v4 ใส่ใน N ดังนั้นต้องมีการปรับค่าในตาราง โดยที่การเชื่อมต่อของจุด v5 จะมีค่าเพิ่มขึ้นเป็น 8 ดังนั้นได้ตารางต่อไปนี้

A picture containing graphical user interface, table

Description automatically generated

จากนั้นเลือกเส้นเชื่อม v7v8 เนื่องจากมีน้ำหนักน้อยที่สุด ดังนั้นต้นไม้แบบทอดข้าม T ก็จะเป็น {v3v10 , v2v3 , v1v2, v8v1, v2v9, v4v10, v7v8} ส่วน A เป็น {v5 , v6 } และ N เป็น {v10 ,v3 , v2 , v1 , v8 , v9 , v4 , v7} ซึ่งแสดงเป็นต้นไม้แบบทอดข้ามได้ดังนี้

Diagram

Description automatically generated

รอบที่ 8 เนื่องจากนำจุดยอด v7 ใส่ใน N ดังนั้นต้องมีการปรับค่าในตาราง โดยที่การเชื่อมต่อของจุด v6 จะมีค่าลดลงเป็น 5 ดังนั้นได้ตารางต่อไปนี้

Table

Description automatically generated

จากนั้นเลือกเส้นเชื่อม v6v7 เนื่องจากมีน้ำหนักน้อยที่สุด ดังนั้นต้นไม้แบบทอดข้าม T ก็จะเป็น {v3v10 , v2v3 , v1v2, v8v1, v2v9, v4v10, v7v8, v6v7} ส่วน A เป็น { v5 } และ N เป็น {v10 ,v3 , v2 , v1 , v8 , v9 , v4 , v7 , v6} ซึ่งแสดงเป็นต้นไม้แบบทอดข้ามได้ดังนี้

Diagram

Description automatically generated

รอบที่ 9 เนื่องจากนำจุดยอด v6 ใส่ใน N ดังนั้นต้องมีการปรับค่าในตาราง โดยที่การเชื่อมต่อของจุด v5 จะมีค่าลดลงเป็น 4 ดังนั้นได้ตารางต่อไปนี้

Graphical user interface, table

Description automatically generated

จากตารางพบว่าเส้นเชื่อม v5v6 เป็นเส้นเชื่อมเดียวที่เหลืออยู่ ดังนั้นต้นไม้แบบทอดข้าม ข้าม T ก็จะเป็น {v3v10 , v2v3 , v1v2, v8v1, v2v9, v4v10, v7v8, v6v7 , v5v6 } ส่วน A เป็น เซตว่าง และ N เป็น {v10 ,v3 , v2 , v1 , v8 , v9 , v4 , v7 , v6 , v5} ซึ่งได้จำนวนเส้นเชื่อมครบแล้วก็จะได้คำตอบดังรูปต่อไปนี้

Diagram

Description automatically generated

Text, letter

Description automatically generatedTable

Description automatically generated

Diagram

Description automatically generated**ตัวอย่างทดสอบแบบที่ 2**

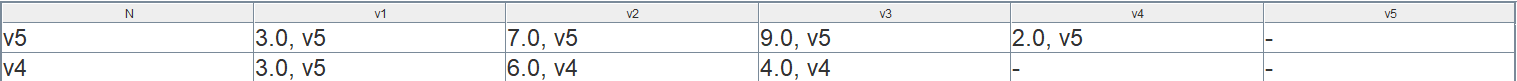
ให้จุดเริ่มต้นจากจุดยอด v5 ต่อไปก็ให้ค่าเริ่มต้นกับข้อมูลที่จำเป็น ให้ N เป็น {v5} ให้ T เป็นเซตว่างและ A เป็น {v1,v2,v3,v4 } ดังนั้นจึงหาเส้นเชื่อมที่มีจุดปลายใน A ที่มาตกกระทบกับ v5 ซึ่งเก็บได้เป็นตารางดังนี้



Diagram

Description automatically generatedจากนั้นเลือกเส้นเชื่อมที่มีน้ำหนักน้อยสุดก็คือ v4v5 ใส่ลงไปใน T และเพิ่มจุดยอด N เป็น {v5 , v4} และ A คือ {v1,v2,v3} จะได้รูปต่อไปนี้

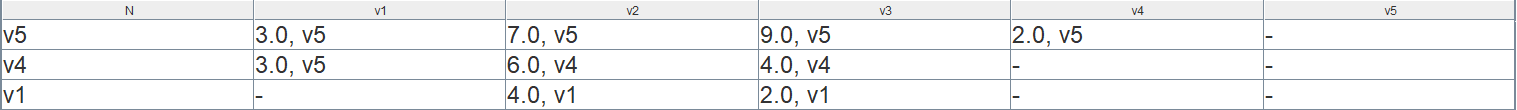
รอบที่ 2 จะมีการปรับค่าตำแหน่งน้อยที่สุดระหว่างจุดที่อยู่ในเซต A และเซต N เนื่องจากมีจุดใหม่ v4 ในเซต N ขณะนี้ก็จะเชื่อมเข้ามาผ่านจุดยอด v4 คือจุดยอด v2 และ v3 เมื่อเชื่อมทางจุดยอด v3 ก็จะมีน้ำหนักลดลง แต่จุดยอด v1เชื่อมกับ v4 น้ำหนักไม่ลดลงก็จะไม่เปลี่ยนค่าในตาราง ซึ่งพิจารณาได้ในตารางต่อไปนี้



Chart, radar chart

Description automatically generatedจากนั้นจึงหาเส้นเชื่อมที่มีน้ำหนักที่น้อยที่สุด ซึ่งก็คือ v1v5 จากนั้นก็ปรับข้อมูลโดยเพิ่ม v1 ลงใน N พร้อมกันนั้นก็ลบ v1 ออกจาก A และต้นไม้แบบทอดข้าม T เป็น {v5v4 , v1v5} โดย N เป็น {v5 , v4 , v1} และ A เป็น {v2,v3} แสดงได้ดังรูปต่อไปนี้

รอบที่ 3 เมื่อนำจุดยอด v1 ใส่ใน N แล้วจะต้องมีการปรับเปลี่ยนค่าน้ำหนักของการเชื่อมระหว่างเซต A และ N ใหม่ดังนี้

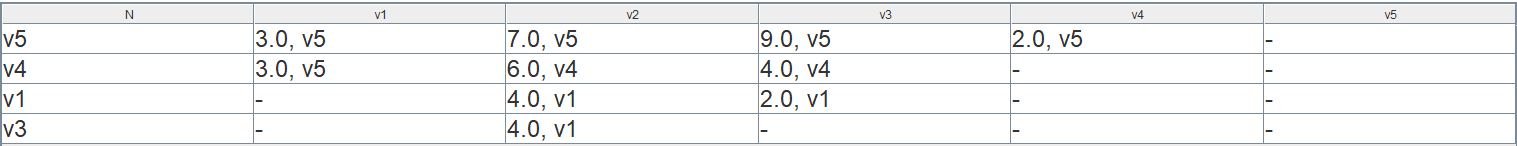


จากนั้นจึงหาเส้นเชื่อมที่มีน้ำหนักที่น้อยที่สุด ซึ่งก็คือ v1v3 จากนั้นก็ปรับข้อมูลโดยเพิ่ม v3 ลงใน N พร้อมกันนั้นก็ลบ v3 ออกจาก A และต้นไม้แบบทอดข้าม T เป็น {v5v4 , v1v5 , v1v3 } โดย N เป็น {v5 , v4 , v1, v3} และ A เป็น {v2} แสดงได้ดังรูปต่อไปนี้

Radar chart

Description automatically generated

รอบที่ 4 เนื่องจากนำจุดยอด v3 ใส่ใน N พบว่าไม่มีการปรับค่าตารางดังนี้



จากตารางพบว่าเส้นเชื่อม v1v2 เป็นเส้นเชื่อมเดียวที่เหลืออยู่ ดังนั้นต้นไม้แบบทอดข้าม ข้าม T ก็จะ{v5v4 , v1v5 , v1v3 , v1v2 } ส่วน A เป็น เซตว่าง และ N เป็น {v5 , v4 , v1, v3 , v2 } ซึ่งได้จำนวนเส้นเชื่อมครบแล้วก็จะได้คำตอบดังรูปต่อไปนี้

Text, letter

Description automatically generatedDiagram, radar chart

Description automatically generated

**อ้างอิง**

รูปแบบโปรแกรม

* อาจารย์ ผู้ช่วยศาสตราจารย์ ดร.คทา ประดิษฐวงศ์ จากเว็บไซต์
* <https://github.com/PTaati/GraphDrawing/tree/master/GraphDrawing>