

รายงาน

เรื่อง การหาดีกรีของจุดยอดในกราฟ โดยใช้โปรแกรมในการวาดกราฟ

จัดทำโดย

นายธนกฤต มีแย้ม	620710166
นายคทากาญจน์ ทรัพย์เสริม	620710650
นายณัฏฐวุฒิ อภินันท์กุล	620710657
นางสาวปัทมพร เอี่ยมณิรัตน์กุล	620710728

เสนอ

อาจารย์ ผู้ช่วยศาสตราจารย์ ดร.คทา ประดิษฐ์วงศ์

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา

โครงสร้างเชิงการคำนวณแบบไม่ต่อเนื่อง รหัสวิชา 517213-2560

มหาวิทยาลัยศิลปากร วิทยาเขตสนามจันทร์

ภาคเรียนที่ 1 ปีการศึกษา 2563

คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชาโครงสร้างเชิงการคำนวณแบบไม่ต่อเนื่อง รหัสวิชา 517213-2560 ของนักศึกษาปีการศึกษา 2563 เพื่อให้ได้ศึกษาหาความรู้ในเรื่องการหาดีกรีของจุดยอดในกราฟ โดยใช้โปรแกรมในการวาดกราฟ และได้ศึกษาอย่างเข้าใจเพื่อเป็นประโยชน์กับการเรียนมากที่สุด

รายงานฉบับนี้ จะเป็นประโยชน์ต่อผู้ที่สนใจจะศึกษา ที่กำลังหาข้อมูลอยู่ หากมีข้อผิดพลาดประการใด คณะผู้จัดทำขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

สารบัญ

ส่วนเพิ่มเติมของโปรแกรม.....	1
การเก็บกราฟ ข้อมูลในตารางและเซตจุดยอด และเส้นเชื่อม.....	9
การ compile หรือ run โปรแกรม.....	10
คุณสมบัติหรือความสามารถของโปรแกรม.....	18
วิธีการใช้งานของโปรแกรมและการแสดงผล.....	19
ตัวอย่างทดสอบแบบที่ 1.....	31
ตัวอย่างทดสอบแบบที่ 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
isSelected	boolean	จุดยอดนี้ถูกเมาส์คลิกเลือกไว้หรือไม่
id	int	เก็บ id เฉพาะของ Vertex
idGen	int	สร้าง id เฉพาะ Vertex
Vertex(x:int,y:int)	-	r = 36 , shift = 30 , name = “ ” , isSelected = false
inCircle(x0:int,y0:int)	boolean	ตรวจสอบว่า 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)
isSelected	boolean	เส้นเชื่อมนี้ถูกเมาส์คลิกเลือกไว้หรือไม่
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 ที่สามารถคลิกเพื่อเลือกเส้นได้
Edge_(a:Vertex,b:Vertex)	-	r_center = 50 , weight = "1" , isSelected = false
inLine(x0:int,y0:int)	boolean	ตรวจสอบว่า 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 อาจจะเป็นเส้นเชื่อมวงวนได้

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 จุดยอดหรือเส้นเชื่อมที่จะทำการแก้ไข โดยใช้เมาส์คลิกเลือก
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”
sanSerifFont	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	ใช้สำหรับเลือกไฟล์ 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 จะแสดงหน้าต่างให้เลือกตำแหน่งไฟล์ที่กราฟที่จะนำเข้ามา เปิดในโปรแกรม จากนั้นจะเรียกใช้ method open(path) เพื่ออ่านไฟล์กราฟ
save(path:String)	void	ใช้ Gson ในการแปลง object Backup เป็น json และบันทึกไฟล์ลงในเครื่องตามตำแหน่งที่กำหนด
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 ที่เมาส์คลิก ถ้าตำแหน่งที่คลิกคือ vertex แล้ว vertex นั้นจะเป็นวัตถุที่ถูกเลือก ถ้าไม่มี vertex ที่ตำแหน่งนั้นก็จะไปตรวจสอบเส้นเชื่อมในแบบเดียวกัน ถ้าตำแหน่งนั้นไม่ใช่ทั้งเส้นเชื่อมและจุดยอด วัตถุที่ถูกเลือกจะถูกยกเลิกการเลือก
Draw()	void	วาดจุดยอดและเส้นเชื่อมทั้งหมดลงบน canvas ถ้าเส้นเชื่อมหรือจุดยอดนั้นถูกเลือกจะแสดงสีน้ำเงิน ถ้าไม่ใช่จะเป็นสีดำ (วาดเส้นเชื่อม ก่อน แล้วค่อยวาดจุดยอดทับ)

keyTyped(ke:KeyEvent)	void	<p>เมื่อกดแป้นพิมพ์</p> <p>เมธอดนี้จะทำงานโดยจะตรวจว่ากดอะไรเข้ามา</p> <p>ถ้ากด ctrl+s จะทำการบันทึกกราฟลงไฟล์ backup.json</p> <p>ctrl+o จะเปิดกราฟจากไฟล์ backup.json</p> <p>ถ้า vertex ถูกเลือกอยู่ การพิมพ์จะเป็นการพิมพ์ชื่อให้แก่ vertex</p> <p>ถ้า edge_ ถูกเลือกอยู่ การพิมพ์จะเป็นการพิมพ์ค่าให้แก่ weight ของเส้น</p>
keyPressed(ke:KeyEvent)	void	<p>ถ้ากด spacebar ค้างในระลอกที่กดค้างโปรแกรมจะอยู่ใน mode “Edge_”</p>
keyReleased(ke:KeyEvent)	void	<p>ถ้าเลิกกด keyboard โปรแกรมจะอยู่ใน mode “Vertex”</p>
mouseMoved (me MouseEvent)	void	-
mouseClicked (e:MouseEvent)	void	<p>เมื่อคลิกเมาส์ลงไป 1 ครั้งจะเป็นการเลือกวัตถุในกราฟ (vertex , edge)</p> <p>ถ้า double click จะเป็นการสร้าง Vertex ใหม่ที่ตำแหน่งที่ double click</p>
mouseDragged (e:MouseEvent)	void	<p>การกดเมาส์ค้าง</p> <p>ถ้าอยู่ใน mode “Vertex”</p> <p>จะเป็นการเปลี่ยนตำแหน่งของเส้นเชื่อมหรือจุดยอดที่ถูกเลือกอยู่</p> <p>ถ้าอยู่ใน mode “Edge_” จะเป็นการวาดเส้นเชื่อม แต่จะวาดได้เมื่อเริ่มกดจากภายในจุดยอดไปสู่จุดยอดเท่านั้น</p>
mouseReleased (e:MouseEvent)	void	<p>การออกจากการคลิก (เมื่อคลิกเสร็จแล้ว) ถ้าอยู่ใน mode “Vertex” จะยกเลิกการวาดเส้นเชื่อมที่ยังวาดไม่เสร็จ</p> <p>ถ้าอยู่ใน mode “Edge_” จะทำการสร้างเส้นเชื่อมขึ้นมา โดยมี 2 แบบ คือเส้นเชื่อมระหว่าง 2 จุดยอดกับเส้นเชื่อมวงวน</p>

mousePressed (e:MouseEvent)	void	เมื่อเริ่มกดคลิกเมาส์ และอยู่ใน mode “Edge_” จะสร้างเส้นเชื่อมชั่วคราวขึ้นมาโดยจุดยอดเริ่มต้นคือจุดยอด ในตำแหน่งที่เมาส์เริ่มคลิก 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	ส่วนการทำงานของ 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>

2. ตาราง

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

2.2 ถ้าจุดยอด w ประชิดกับจุดยอด u ให้ทำการเปรียบเทียบน้ำหนักของ w ก่อนหน้าในตารางกับเส้นเชื่อมที่ตกกระทบ หาก $u < w$ ให้ทำการปรับค่าโดยใส่ น้ำหนัก และจุดยอด u คั่นด้วย , เพื่อแยกจุดยอดและน้ำหนัก

2.3 หาก w ไม่ประชิดกับ u จะไม่ปรับค่าลงตาราง

3. เซตจุดยอด

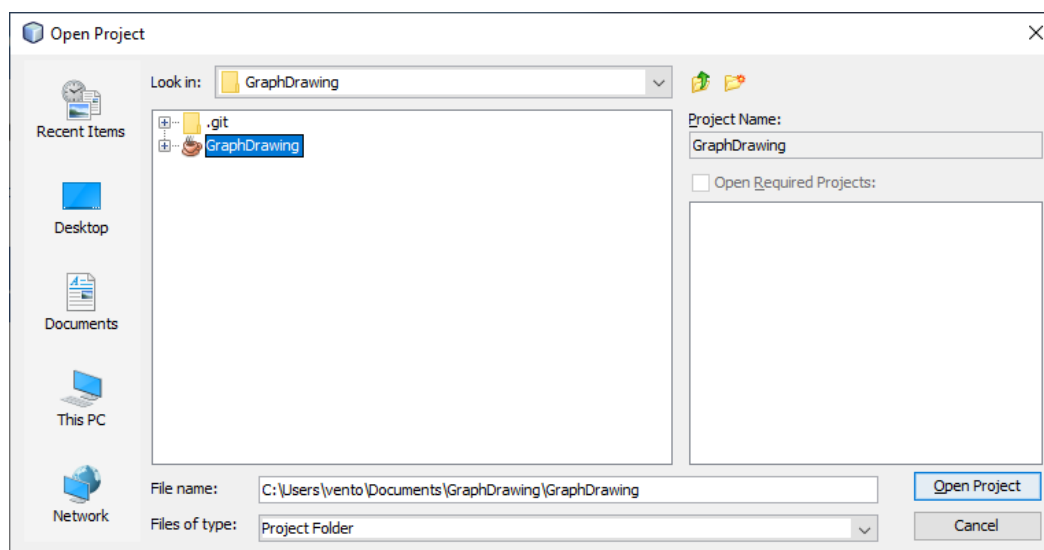
โปรแกรมจะเก็บเซตของจุดยอดในลักษณะของ LinkedList โดยจะมี LinkedList ที่เก็บจุดยอดที่ยังไม่ได้อยู่ในต้นไม้แบบทอดข้าม คือ LinkedList<Vertex> A และ LinkedList ที่เก็บจุดยอดที่อยู่ในต้นไม้แบบทอดข้าม คือ LinkedList<Vertex> N เมื่อ compile หรือ run โปรแกรมนั้น เมื่อเข้าสู่กระบวนการของ Prim's Algorithm ผู้ใช้จะทำการเลือกจุดเริ่มต้น 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 ไปเรื่อยๆ จนกว่า LinkedList<Vertex> A จะเป็นเซตว่างหรือไม่เหลือสมาชิกแล้ว

4. เซตเส้นเชื่อม

โปรแกรมจะเก็บเซตของเส้นเชื่อมในลักษณะ LinkedList โดยจะมี LinkedList ที่เก็บเส้นเชื่อมที่ คือ LinkedList<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 ไปเรื่อยๆ จนกว่า LinkedList<Vertex> A จะเป็นเซตว่างหรือไม่เหลือสมาชิกแล้ว

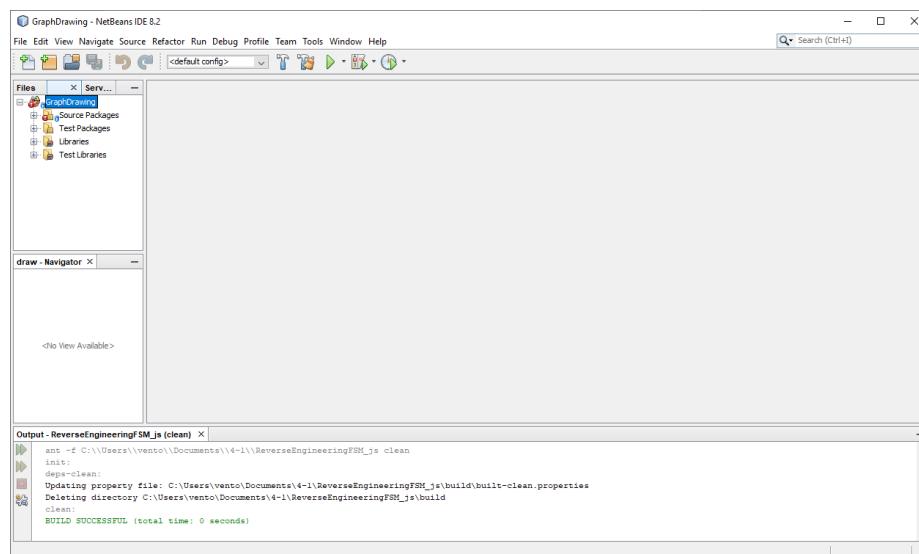
การ compile หรือ run โปรแกรม

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



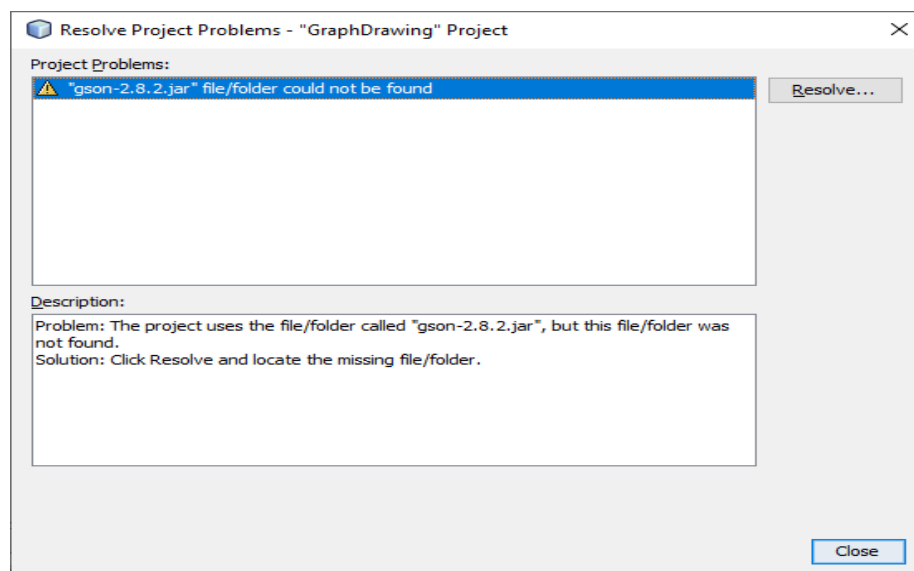
รูปที่ 1 แสดงการ open Project

2. ให้กดคลิกขวาที่ ชื่อโปรเจกต์ที่เปิดแล้วจากนั้นกด Resolve Project Problems ..เพื่อเพิ่ม GSON เข้ามาในโปรเจก



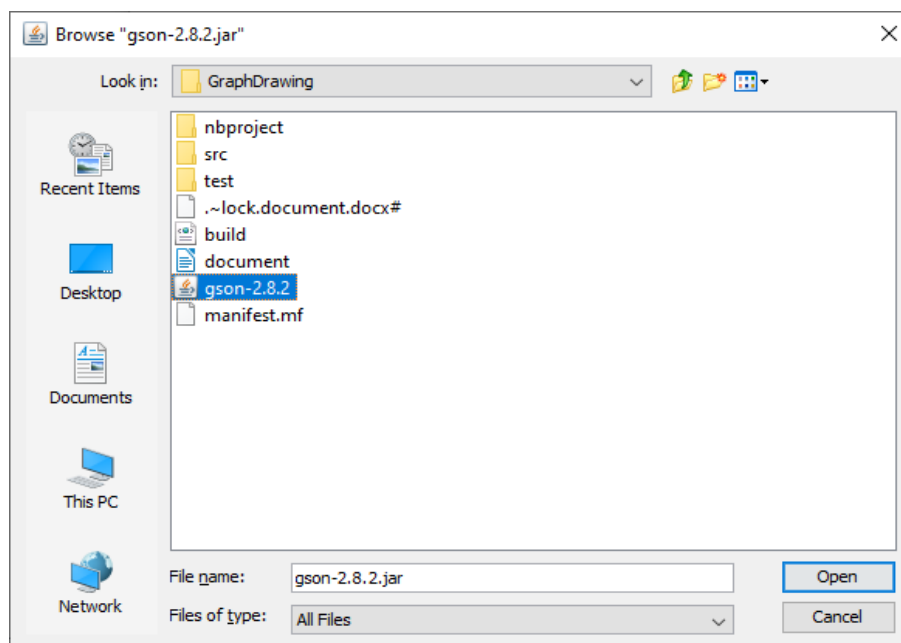
รูปที่ 2 แสดงการเพิ่ม GSON เข้ามาในโปรเจก

3. กด Resolve



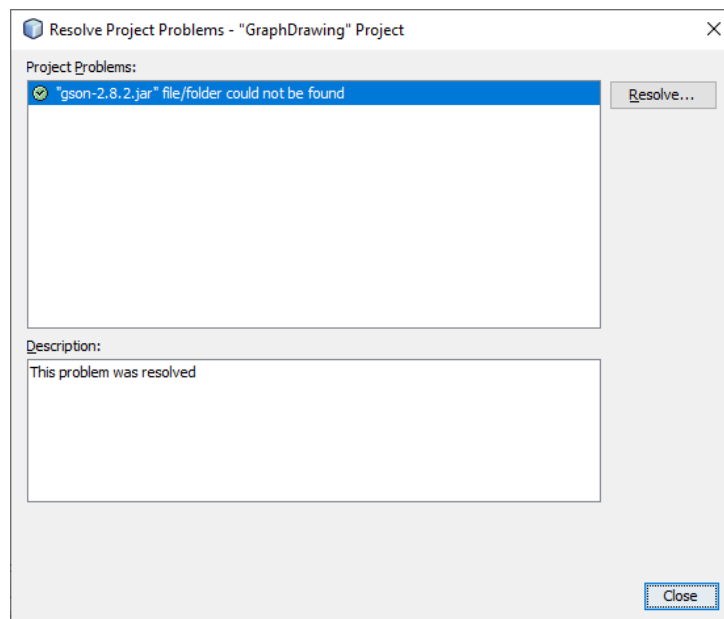
รูปที่ 3 แสดงหน้าต่าง Resolve

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



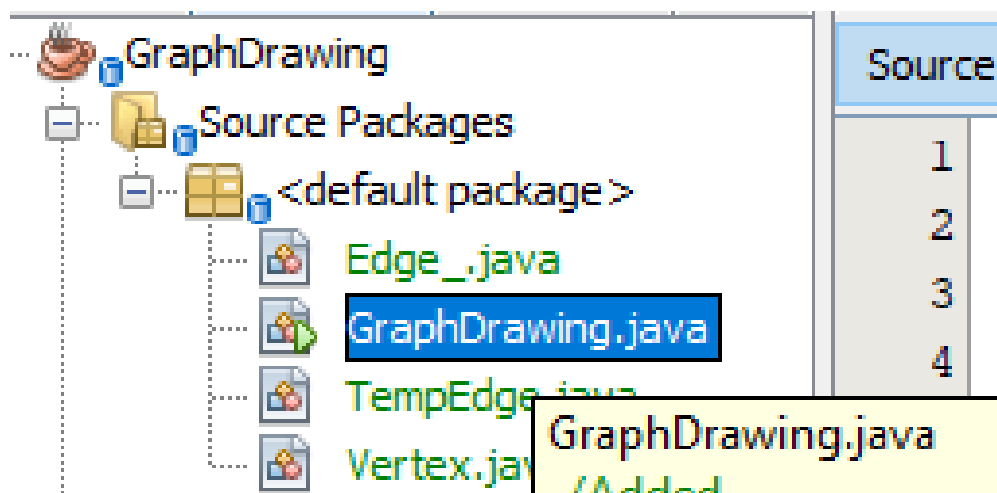
รูปที่ 4 แสดงการเปิดไฟล์ gson-2.8.2.jar

2. กด Close



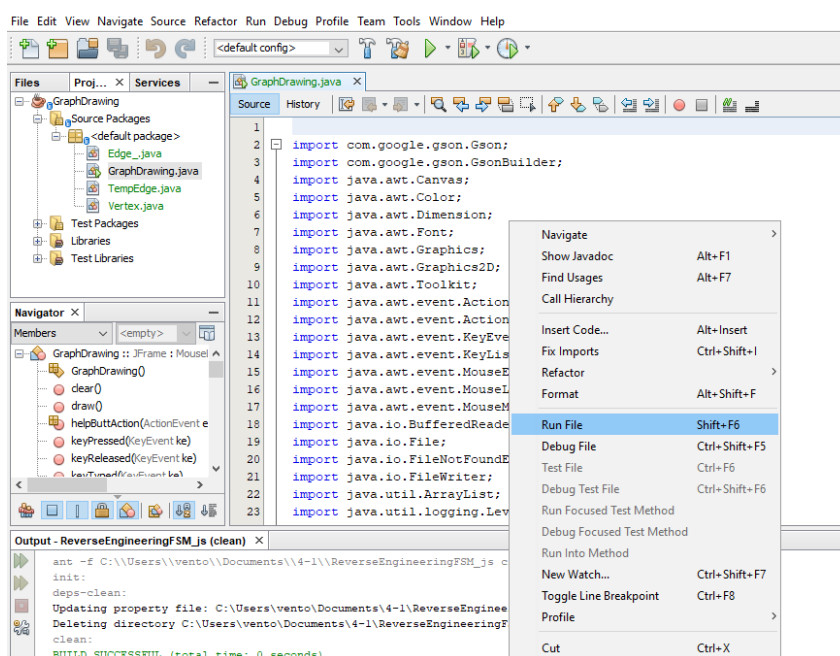
รูปที่ 1 แสดงหน้าต่างหลังจากเพิ่ม GSON

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



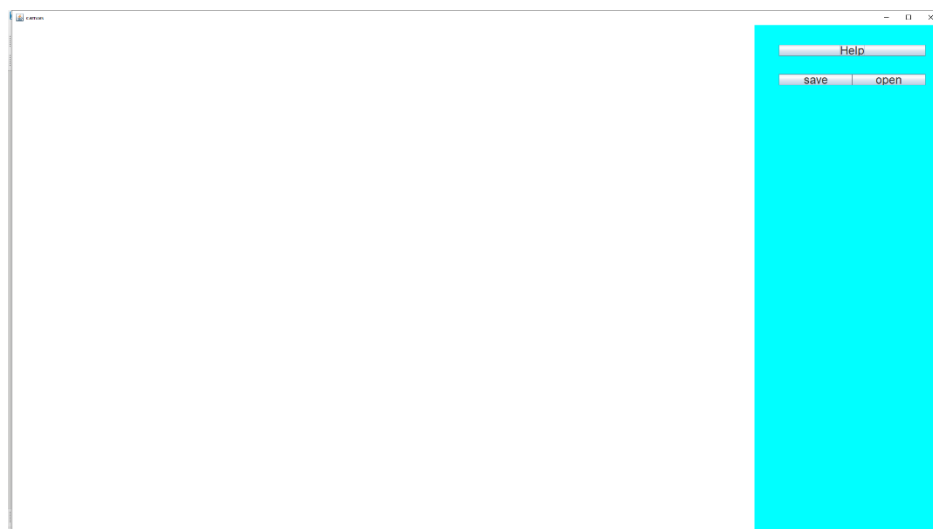
รูปที่ 6 แสดงการเปิดไฟล์ GraphDrawing.java

7. คลิกขวาที่ส่วนของโค้ดแล้วกด Run File (Shift + F6)



รูปที่ 7 แสดงวิธีการกดรัน

8. โปรแกรมหน้าต่าง GUI ขึ้นมาดังนี้

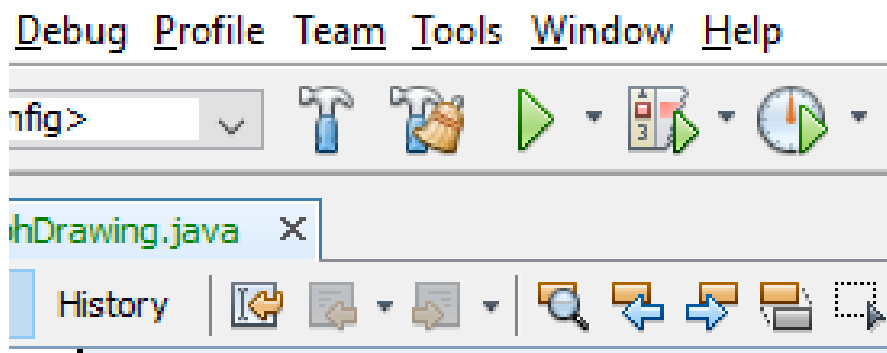


รูปที่ 8 แสดงตัวอย่างหน้าต่าง GUI

9. หากต้องการสร้างไฟล์ .jar ให้กด



Clean and Build Project (Shift + F11)



รูปที่ 9 แสดงวิธีการสร้างไฟล์ .jar

10. แสดงสถานะการ Build Project

```

GraphDrawing - C:\Users\vento\Documents\GraphDrawing X GraphDrawing (clean.jar) X
ant -f C:\Users\vento\Documents\GraphDrawing\GraphDrawing -Dnb.internal.action.name=rebuild clean jar
init:
deps-clean:
Updating property file: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\build-clean.properties
Deleting directory C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build
clean:
init:
deps-jar:
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build
Updating property file: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\build-jar.properties
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\classes
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\empty
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\generated-sources\ap-source-output
Compiling 4 source files to C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\classes
Note: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\src\GraphDrawing.java uses or overrides a deprecated
Note: Recompile with -Xlint:deprecation for details.
compile:
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\dist
Copying 1 file to C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build
Copy libraries to C:\Users\vento\Documents\GraphDrawing\GraphDrawing\dist\lib.
Building jar: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\dist\GraphDrawing.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\vento\Documents\GraphDrawing\GraphDrawing\dist\GraphDrawing.jar"
jar:
BUILD SUCCESSFUL (total time: 2 seconds)

```

รูปที่ 10 แสดงสถานะการสร้าง

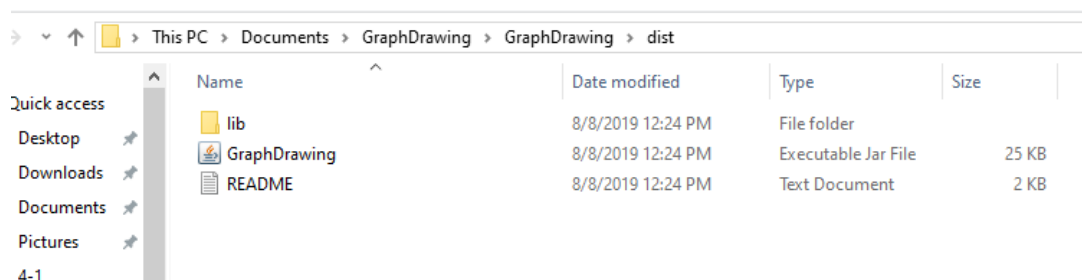
11. หลังจากขั้นตอน 10 จะมีโฟลเดอร์ dist ปรากฏขึ้นมา

PC > Documents > GraphDrawing > GraphDrawing

Name	Date modified	Type	Size
build	8/8/2019 12:24 PM	File folder	
dist	8/8/2019 12:24 PM	File folder	
nbproject	8/6/2019 12:00 PM	File folder	
src	8/6/2019 7:45 PM	File folder	
test	8/8/2019 12:08 PM	File folder	
~lock.document.docx#	8/8/2019 12:25 PM	DOCX# File	1 KB
build	8/8/2019 12:08 PM	XML Document	4 KB
document	8/8/2019 12:25 PM	Microsoft Word D...	1,655 KB
gson-2.8.2	4/24/2019 11:08 AM	Executable Jar File	228 KB

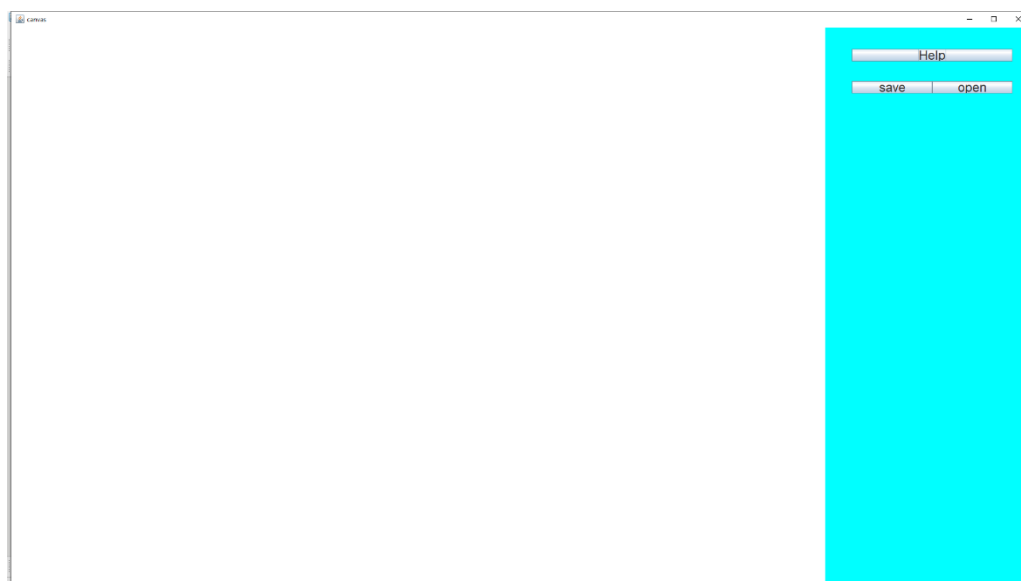
รูปที่ 11 แสดงโฟลเดอร์เมื่อสร้างสำเร็จ

12. เมื่อเข้าไปใน Folder dist ก็จะมีการโปรแกรม 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

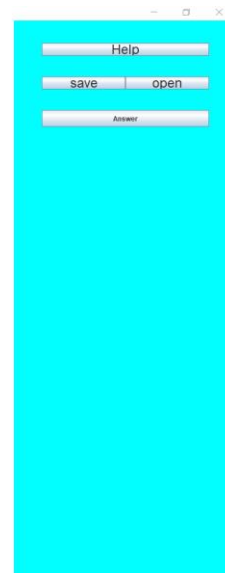
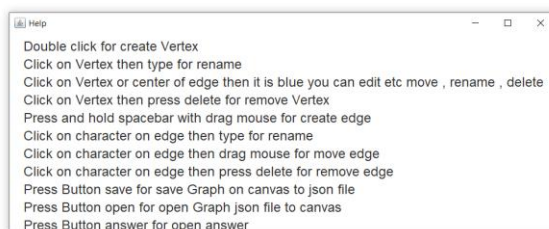
ข้อเสียของโปรแกรม

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

วิธีการใช้งานของโปรแกรมและการแสดงผล

1. เมื่อคลิกที่ Help ในขณะที่ยังไม่มีโปรแกรมจะมีหน้าต่างอธิบายการใช้งานโปรแกรม

canvas



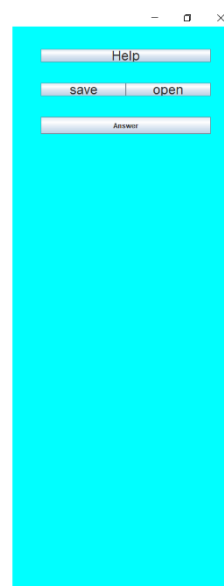
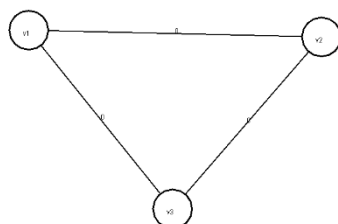
2. หากดับเบิลคลิกจะทำการสร้างจุดยอด

canvas



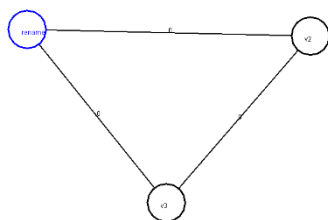
3. หากกดค้างที่แถบสเปซบาร์ และกดค้างที่เมาส์จะเป็นการสร้างเส้นเชื่อม

canvas

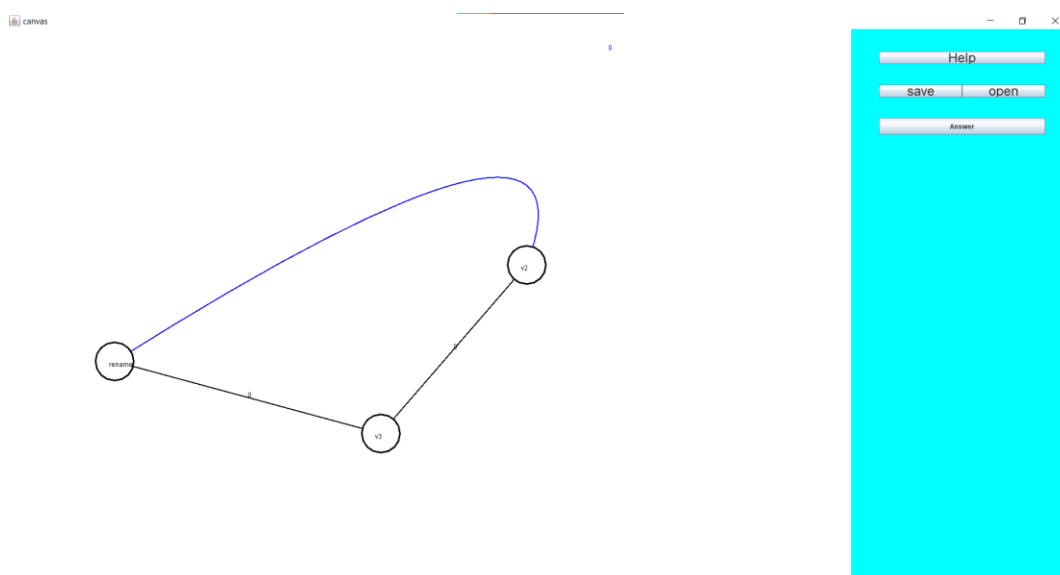
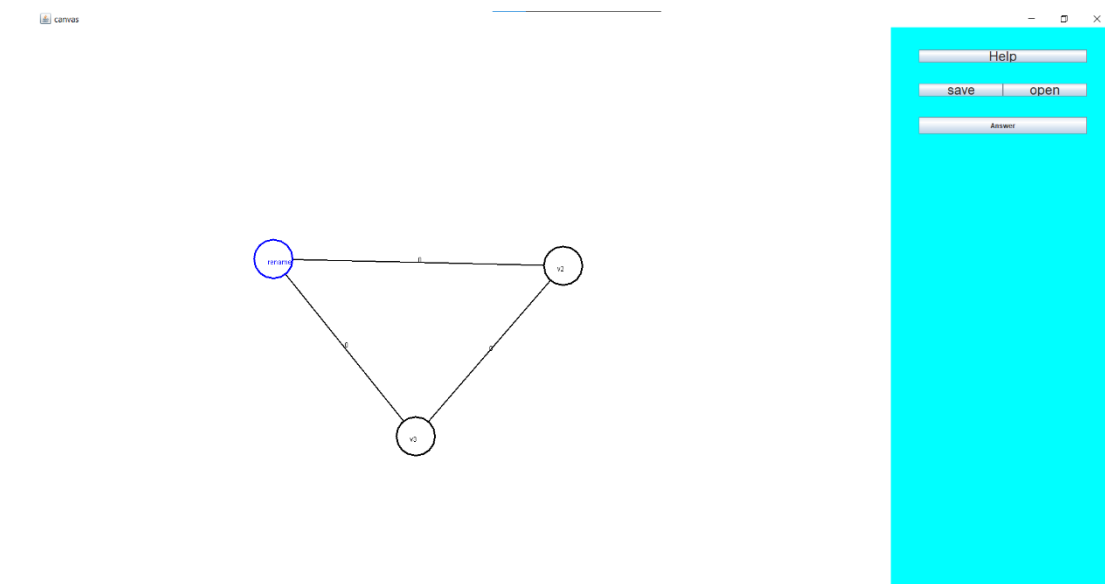


4. หากต้องการเปลี่ยนชื่อของจุดยอด ให้ทำการคลิกที่จุดยอดจะมีสีน้ำเงินขึ้นมา และทำการพิมพ์หรือแก้ไขชื่อที่จุดยอด

canvas

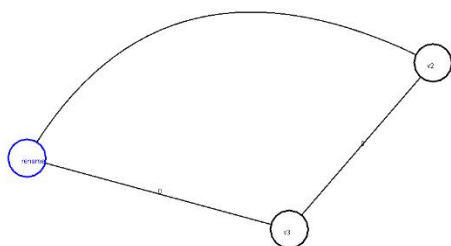


5. หากคลิกที่จุดยอดหรือเส้นเชื่อมและขึ้นเป็นสีน้ำเงิน จะสามารถเคลื่อนย้าย ลบ หรือแก้ไขชื่อได้

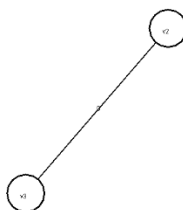


3. หากคลิกที่จุดยอด จากนั้นกดปุ่มลบ จะทำการลบจุดยอดนั้นทันที

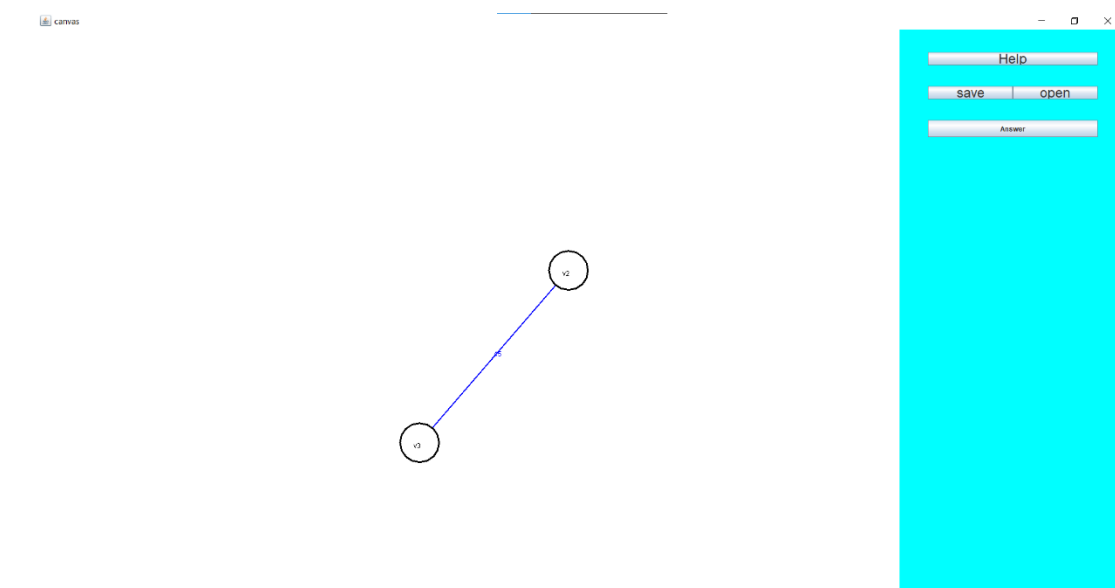
canvas



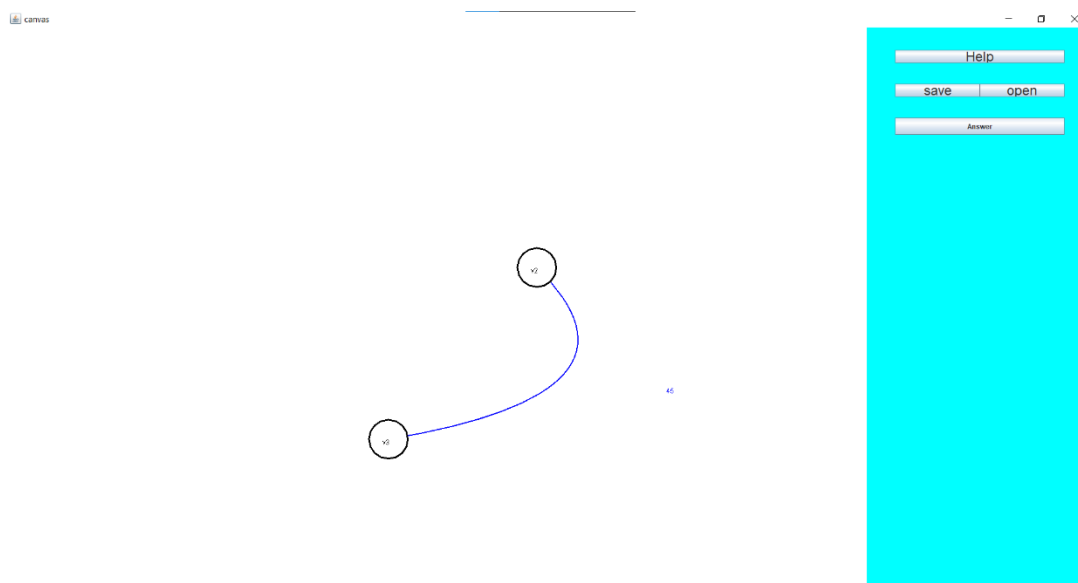
canvas



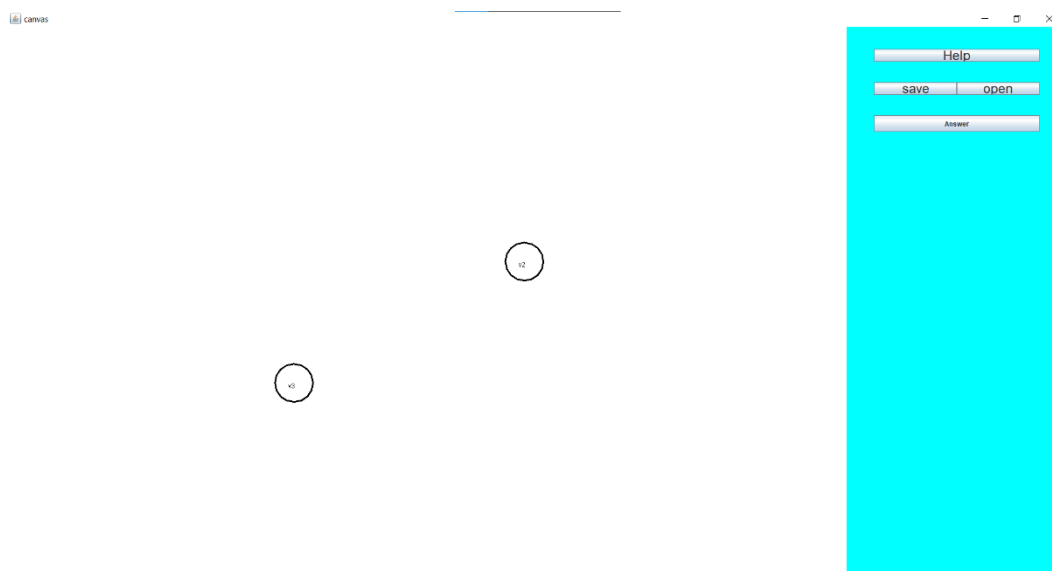
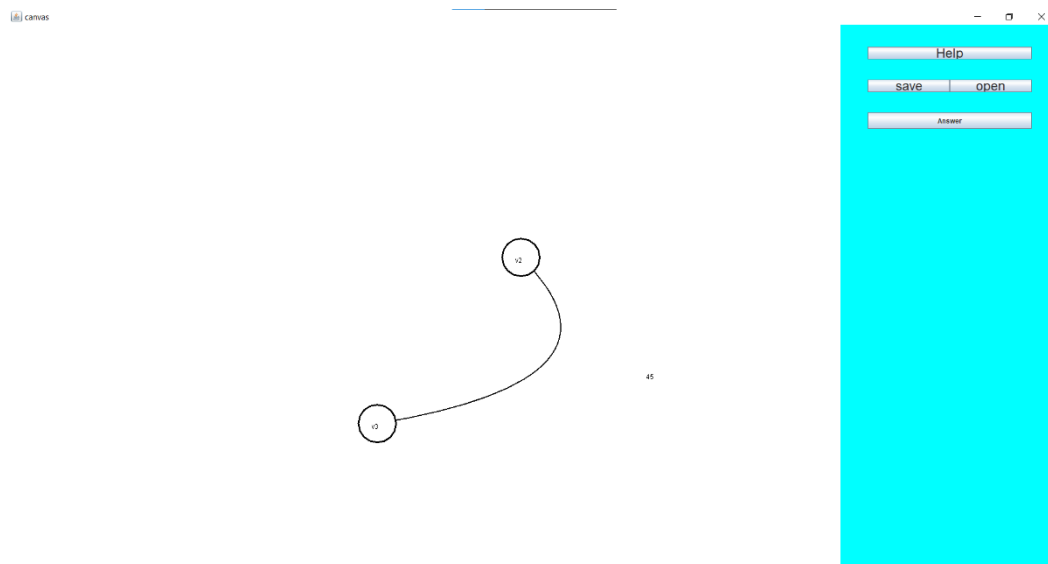
4. หากต้องการเปลี่ยนชื่อของเส้นเชื่อม ให้ทำการคลิกที่ตัวอักษรบนเส้นเชื่อมและพิมพ์เปลี่ยนชื่อของเส้นเชื่อม



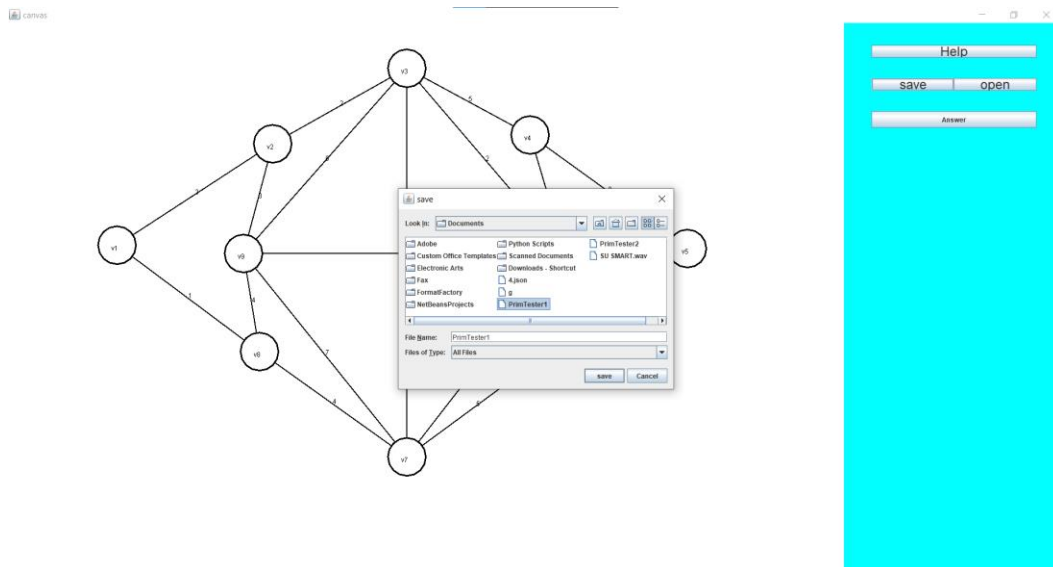
5. หากต้องการเคลื่อนย้ายเส้นเชื่อม ให้ทำการคลิกที่เส้นเชื่อมและลากเพื่อเคลื่อนย้าย



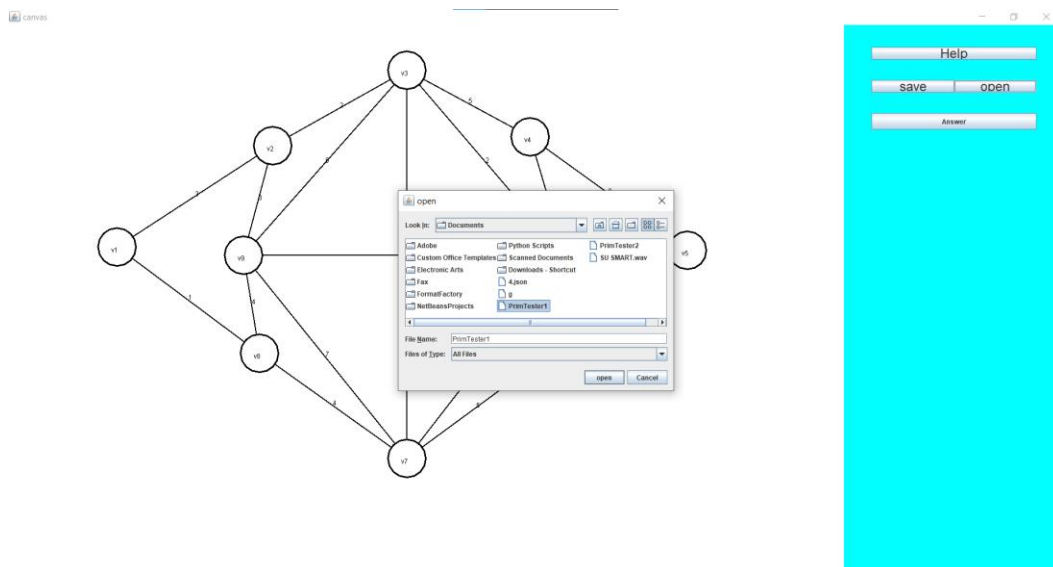
6. หากคลิกที่เส้นเชื่อมจะมีสีน้ำเงินขึ้น จากนั้นกดปุ่มลบ จะทำการลบเส้นเชื่อมนั้นทันที



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



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



9. หากสร้างเส้นเชื่อมระหว่างจุดยอดใดๆแล้ว จะให้กรอกน้ำหนักทันที โดยสามารถกรอกได้เพียงตัวเลขเท่านั้น และขณะกรอกหากมีช่องว่าง อักขระ หรือจำนวนเต็มลบ ระบบจะให้ผู้ใช้ใส่ค่าน้ำหนักใหม่

canvas



Input

Enter your weight:

OK Cancel

Help

save open

Answer

canvas



Input

Enter your weight:

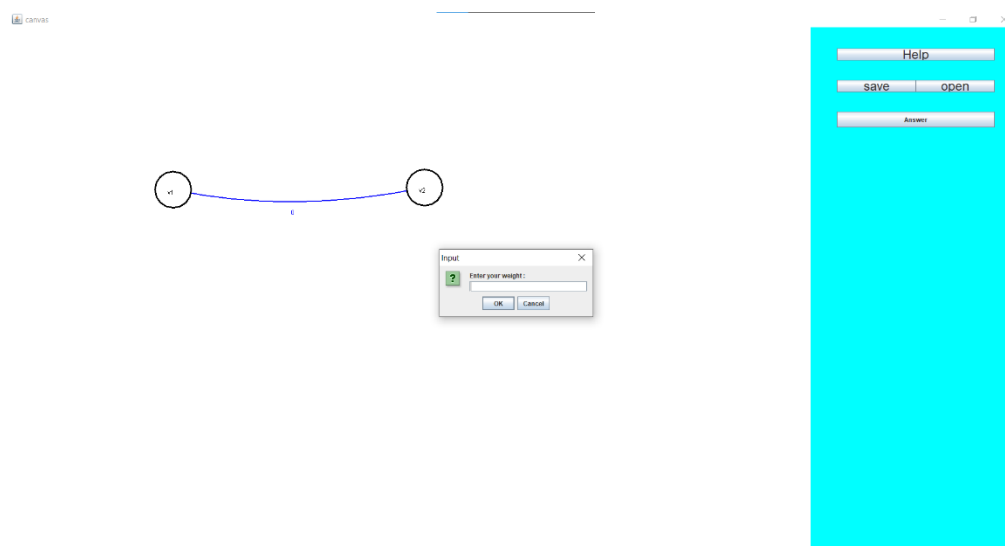
OK Cancel

Help

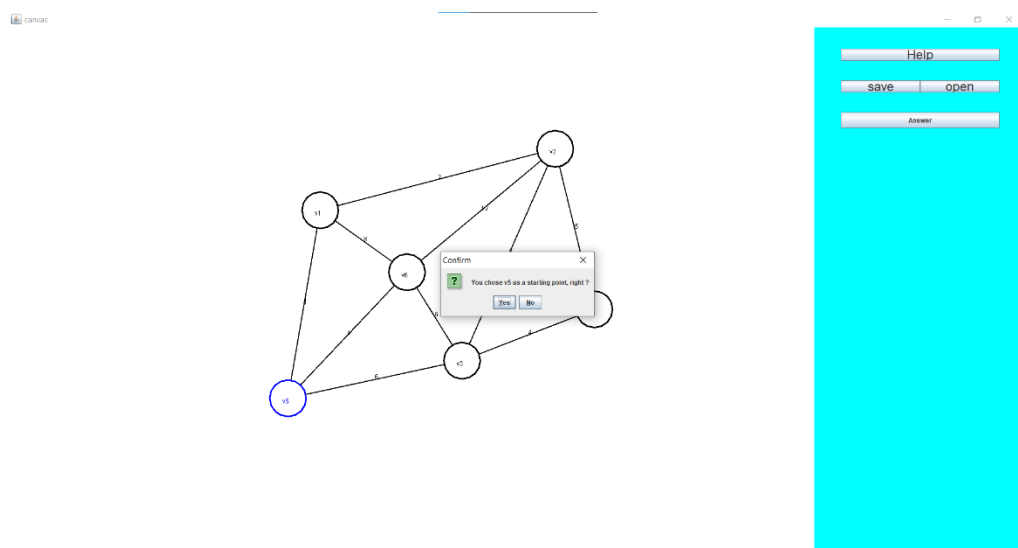
save open

Answer

10. หากผู้ใช้ไม่ได้ใส่น้ำหนัก ผู้ใช้สามารถกดที่เส้นเชื่อม และกด Enter ระบบจะแสดง frame ขึ้นมาใหม่ เพื่อให้ผู้ใช้กรอกน้ำหนักได้อีกครั้ง

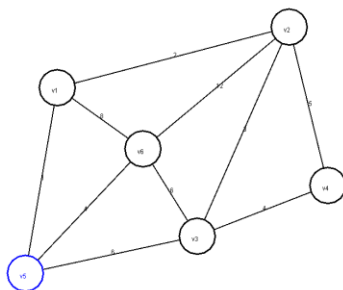


11. เมื่อผู้ใช้สร้างกราฟเสร็จแล้วนั้น ผู้เลือกสามารถเลือกจุดเริ่มต้นได้ โดยผู้ใช้สามารถเลือกจุดยอดให้เป็นสีน้ำเงิน จากนั้นกด Enter จะมี frame ขึ้นมาสอบถามผู้ใช้ว่า เลือกจุดยอดนี้เป็นจุดเริ่มต้นใช่หรือไม่



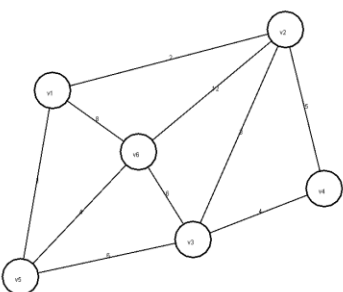
12. เมื่อกดปุ่ม answer จะแสดง frame ขึ้นมาใหม่ ซึ่งจะประกอบด้วย
ปุ่ม home , graph , table , next

canvas

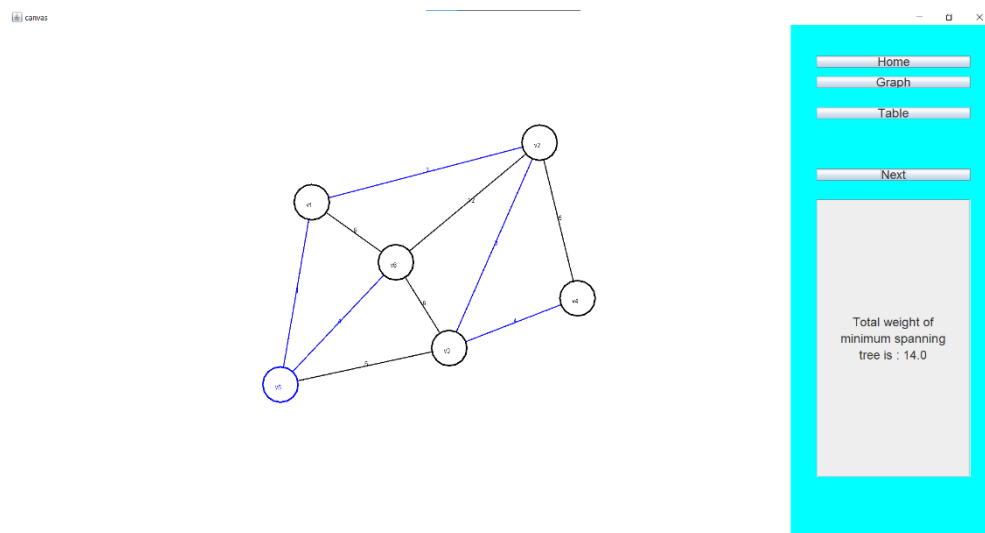


13. เมื่อกดปุ่ม home ระบบจะแสดงกราฟดังต้น

canvas



14. เมื่อกดปุ่ม graph ระบบจะแสดงผลเป็นรูปภาพของกราฟที่ผ่านการประมวลผลตามวิธีการของ Prim's Algorithm

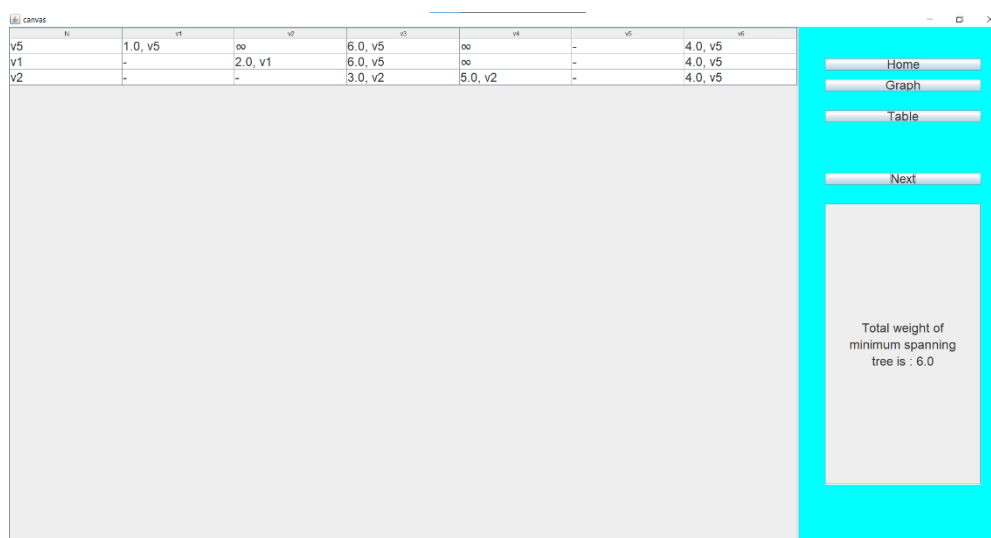
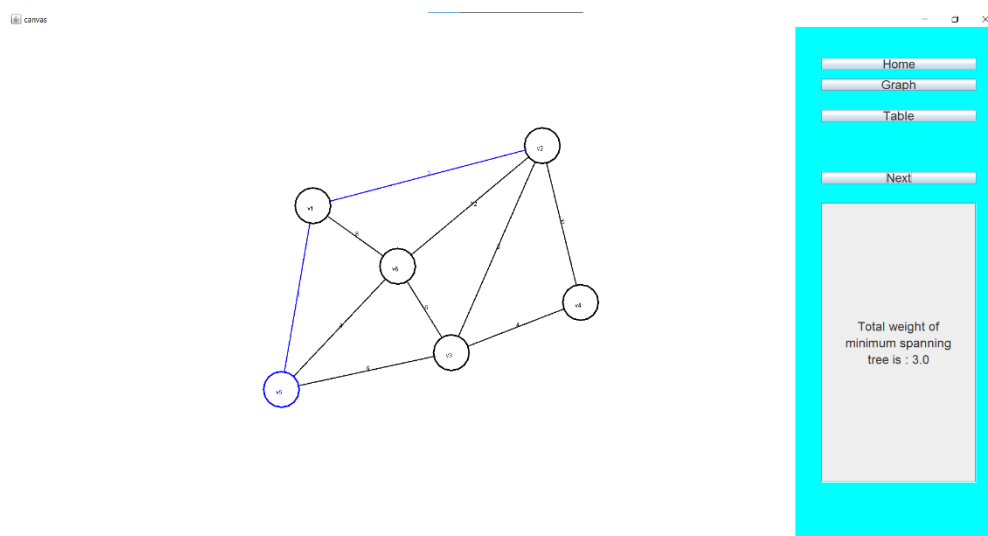


15. เมื่อกดปุ่ม table ระบบจะแสดงผลเป็นตารางคำตอบของกราฟที่ผ่านการประมวลผลตามวิธีการของ Prim's Algorithm

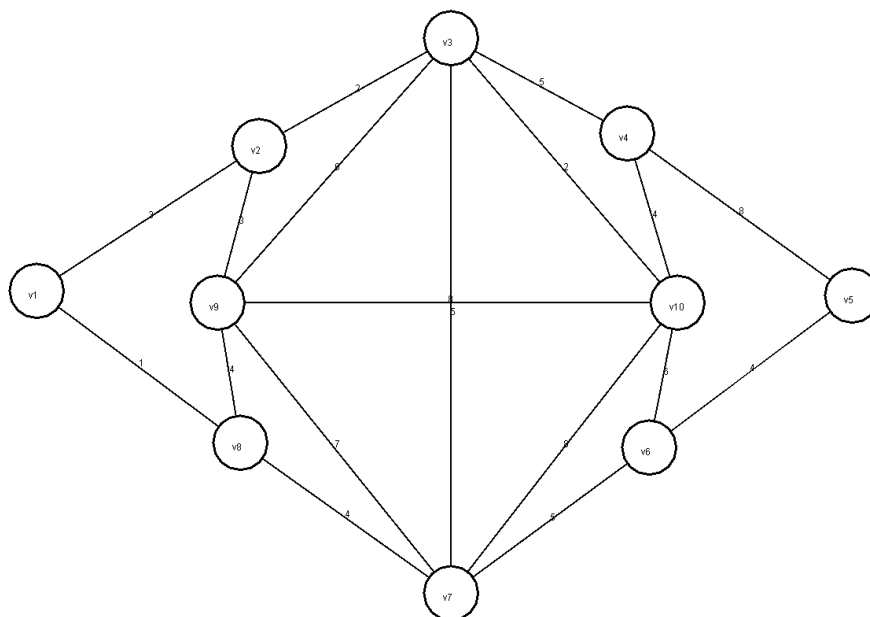
	v1	v2	v3	v4	v5	v6
v5	1.0, v5	∞	6.0, v5	∞	-	4.0, v5
v1	-	2.0, v1	6.0, v5	∞	-	4.0, v5
v2	-	-	3.0, v2	5.0, v2	-	4.0, v5
v3	-	-	-	4.0, v3	-	4.0, v5
v4	-	-	-	-	-	4.0, v5

The sidebar menu on the right is identical to the one in the previous figure, with buttons for Home, Graph, Table, and Next, and a box displaying the text: 'Total weight of minimum spanning tree is : 14.0'.

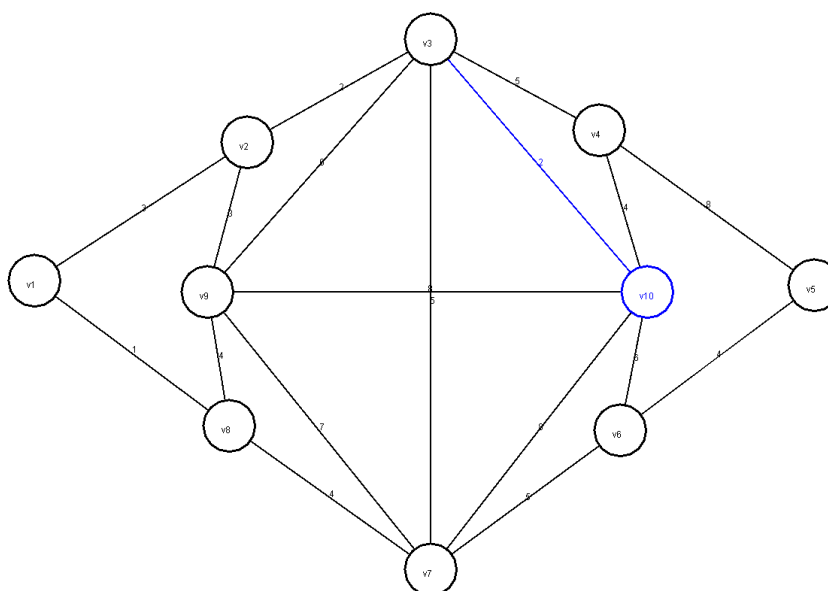
16. เมื่อกดปุ่ม next ระบบจะทำการแสดงเส้นเชื่อมที่ถูกเลือก หรือ ตารางถัดไปที่ผ่านการประมวลผลตามวิธีการของ Prim's Algorithm



ตัวอย่างทดสอบแบบที่ 1



ให้จุดเริ่มต้นจากจุดยอด v_{10} ต่อไปก็ให้ค่าเริ่มต้นกับข้อมูลที่จำเป็น ให้ N เป็น $\{v_{10}\}$ ให้ T เป็นเซตว่างและ A เป็น $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$ ดังนั้นจึงหาเส้นเชื่อมที่มีจุดปลายใน A ที่มาตกกระทบบกับ v_{10} ซึ่งเก็บได้เป็นตารางดังนี้

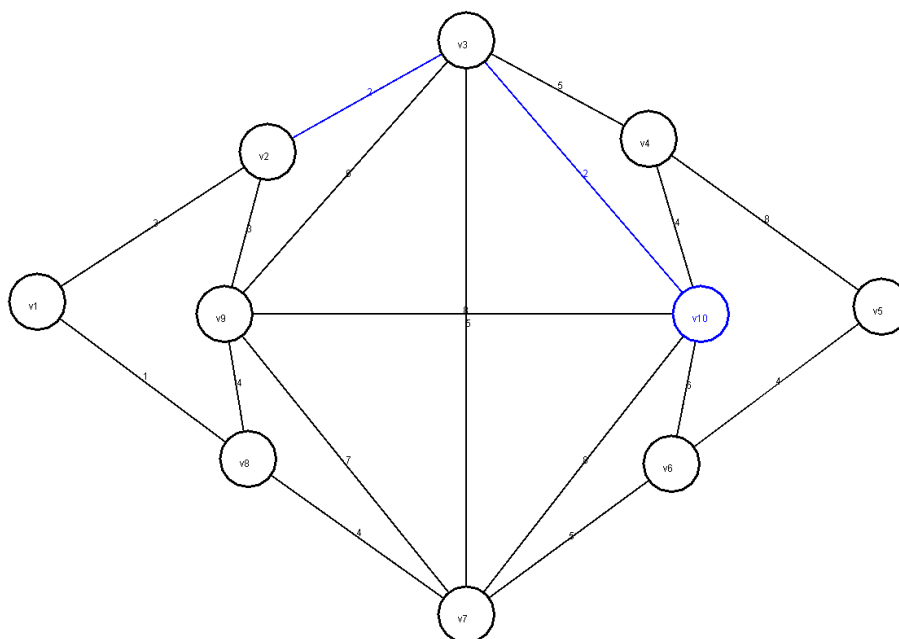


	N	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_{10}	∞	∞		2.0, v_{10}	4.0, v_{10}	∞	6.0, v_{10}	8.0, v_{10}	∞	8.0, v_{10}	-

รอบที่ 2 จะมีการปรับค่าตำแหน่งน้อยที่สุดระหว่างจุดที่อยู่ในเซต A และเซต N เนื่องจากมีจุดใหม่ v_3 ในเซต N ขณะนี้ก็จะเชื่อมเข้ามาผ่านจุดยอด v_3 จุดยอด v_7 และ v_9 เมื่อเชื่อมทางจุดยอด v_3 ก็จะมีน้ำหนักลดลง แต่จุดยอด v_4 เชื่อมกับ v_3 น้ำหนักไม่ลดลงก็จะไม่เปลี่ยนค่าในตาราง ซึ่งพิจารณาได้ในตารางต่อไปนี้

	N	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_{10}		∞	∞	2.0, v_{10}	4.0, v_{10}	∞	6.0, v_{10}	8.0, v_{10}	∞	8.0, v_{10}	-
v_3		∞	2.0, v_3	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	∞	6.0, v_3	-

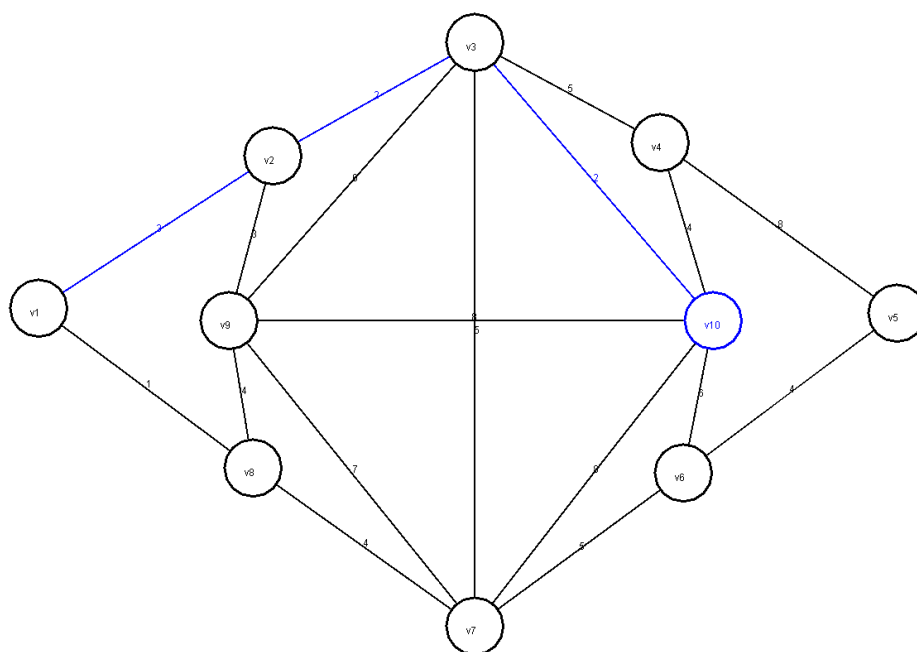
จากนั้นจึงหาเส้นเชื่อมที่มีจุดยอดจุดหนึ่งใน A มาตกกระทบกับ กับจุดยอดใน N ซึ่งคือ v_3 และ v_{10} โดยต้องเป็นเส้นเชื่อมที่มีน้ำหนักที่น้อยที่สุด ซึ่งก็คือ v_2v_3 จากนั้นก็ปรับข้อมูลโดยเพิ่ม v_2 ลงใน N พร้อมกันนั้นก็ลบ v_2 ออกจาก A และต้นไม้แบบทอดข้าม T เป็น $\{v_3v_{10}, v_2v_3\}$ แสดงได้ดังรูปต่อไปนี้



รอบที่ 3 เมื่อนำจุดยอด v_2 ใส่ใน N แล้วจะต้องมีการปรับเปลี่ยนค่าน้ำหนักของการเชื่อมระหว่างเซต A และ N ใหม่ดังนี้

N	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_{10}	∞	∞	2.0, v_{10}	4.0, v_{10}	∞	6.0, v_{10}	8.0, v_{10}	∞	8.0, v_{10}	-
v_3	∞	2.0, v_3	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	∞	6.0, v_3	-
v_2	3.0, v_2	-	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	∞	3.0, v_2	-

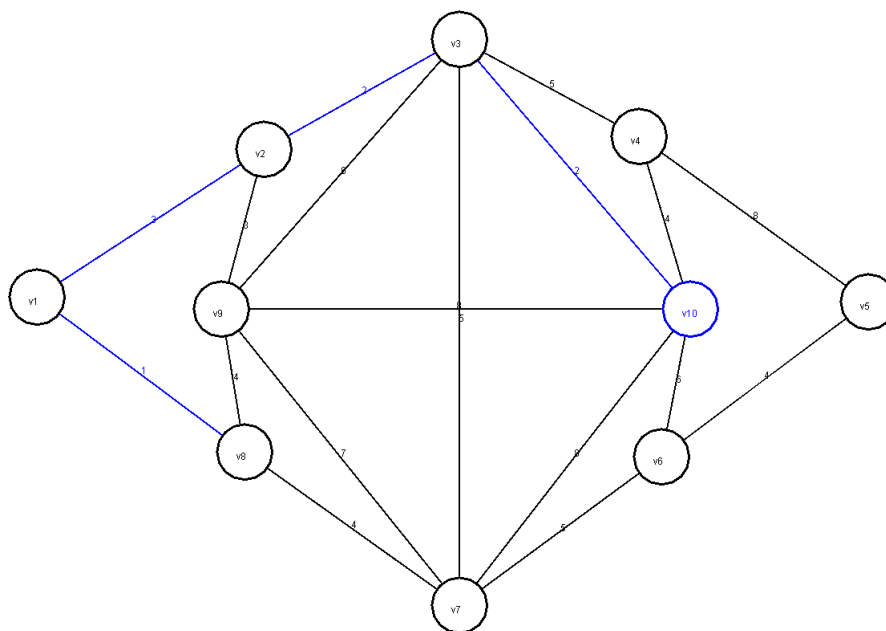
จากนั้นหาเส้นเชื่อมที่มีน้ำหนักน้อยที่สุดต่อไปก็คือ v_1v_2 หรือ v_2v_9 ในกรณีเลือกเส้นเชื่อม v_1v_2 ก่อน ใส่เข้าไปในต้นไม้แบบทอดข้าม T เป็น $\{v_3v_{10}, v_2v_3, v_1v_2\}$ เพิ่มจุดยอด v_1 เข้าไปใน N และลบจุดยอดนี้ออกจาก A จะได้



รอบที่ 4 จากการนำ v_1 ใส่ใน N จะต้องมีการปรับค่าน้ำหนักใหม่ที่จุดยอด v_8 จะเชื่อมมาได้ผ่าน v_1 จะได้ตารางดังต่อไปนี้

	N	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_{10}		∞	∞	2.0, v_{10}	4.0, v_{10}	∞	6.0, v_{10}	8.0, v_{10}	∞	8.0, v_{10}	-
v_3		∞	2.0, v_3	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	∞	6.0, v_3	-
v_2		3.0, v_2	-	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	∞	3.0, v_2	-
v_1		-	-	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	1.0, v_1	3.0, v_2	-

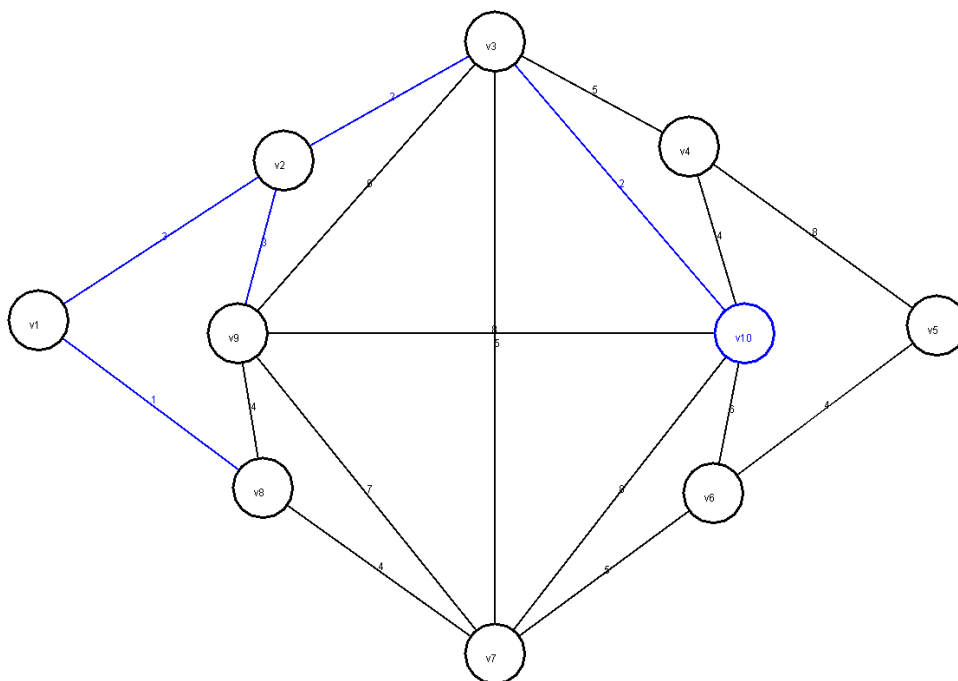
จากนั้นหาเส้นเชื่อมที่มีน้ำหนักน้อยที่สุดต่อไปก็คือ v_8v_1 ย้ายจุดยอด v_8 จาก A ไป N จะได้ว่า A เป็น $\{v_4, v_5, v_6, v_7, v_9\}$ ในขณะที่ T เป็น $\{v_3v_{10}, v_2v_3, v_1v_2, v_8v_1\}$ ต้นไม้จะเป็นดังรูป



รอบที่ 5 เนื่องจากนำจุดยอด v_8 ใส่ใน N ดังนั้นต้องมีการปรับค่าน้ำหนักในตาราง โดยที่การเชื่อมต่อของจุดยอด v_7 จะมีค่าลดลงเป็น 4 จะได้ตารางดังต่อไปนี้

	N	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_{10}		∞	∞	2.0, v_{10}	4.0, v_{10}	∞	6.0, v_{10}	8.0, v_{10}	∞	8.0, v_{10}	-
v_3		∞	2.0, v_3	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	∞	6.0, v_3	-
v_2		3.0, v_2	-	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	∞	3.0, v_2	-
v_1		-	-	-	4.0, v_{10}	∞	6.0, v_{10}	5.0, v_3	1.0, v_1	3.0, v_2	-
v_8		-	-	-	4.0, v_{10}	∞	6.0, v_{10}	4.0, v_8	-	3.0, v_2	-

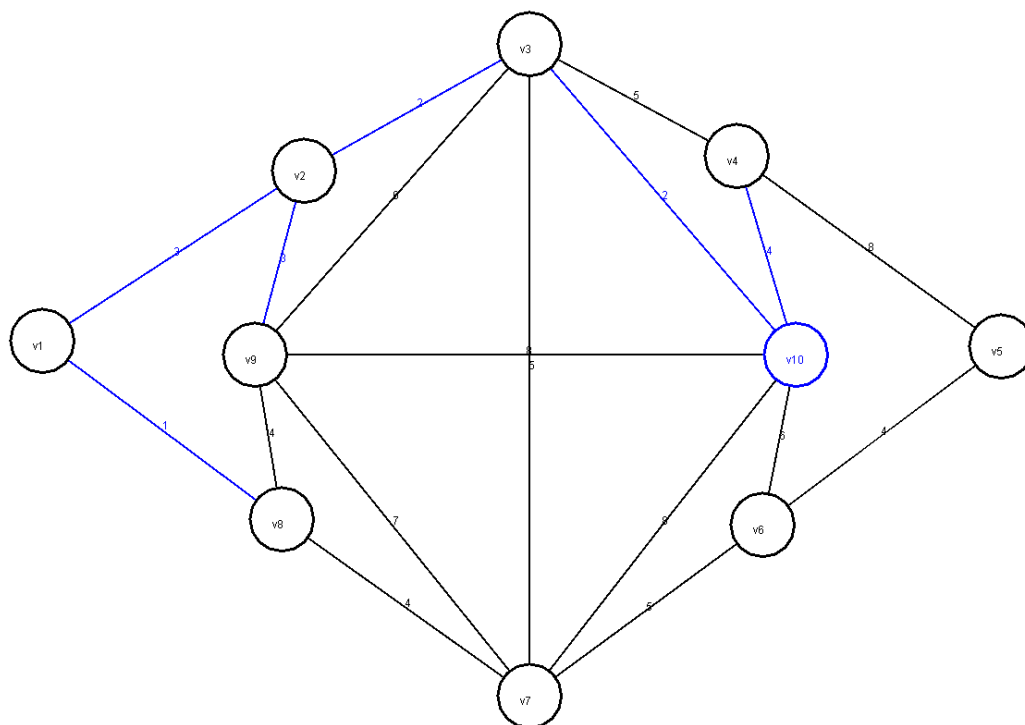
จากนั้นเลือกเส้นเชื่อม v_2v_9 เนื่องจากมีน้ำหนักน้อยที่สุด ดังนั้นต้นไม้แบบทอดข้าม T ก็จะเป็น $\{v_3v_{10}, v_2v_3, v_1v_2, v_8v_1, v_2v_9\}$ ส่วน A เป็น $\{v_4, v_5, v_6, v_7\}$ และ N เป็น $\{v_{10}, v_3, v_2, v_1, v_8, v_9\}$



รอบที่ 6 เนื่องจากนำจุดยอด v_9 ใสใน N โดยพบว่าไม่มีการปรับค่าน้ำหนักดังตารางต่อไปนี้

N	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
v10	∞	∞	2.0, v10	4.0, v10	∞	6.0, v10	8.0, v10	∞	8.0, v10	-
v3	∞	2.0, v3	-	4.0, v10	∞	6.0, v10	5.0, v3	∞	6.0, v3	-
v2	3.0, v2	-	-	4.0, v10	∞	6.0, v10	5.0, v3	∞	3.0, v2	-
v1	-	-	-	4.0, v10	∞	6.0, v10	5.0, v3	1.0, v1	3.0, v2	-
v8	-	-	-	4.0, v10	∞	6.0, v10	4.0, v8	-	3.0, v2	-
v9	-	-	-	4.0, v10	∞	6.0, v10	4.0, v8	-	-	-

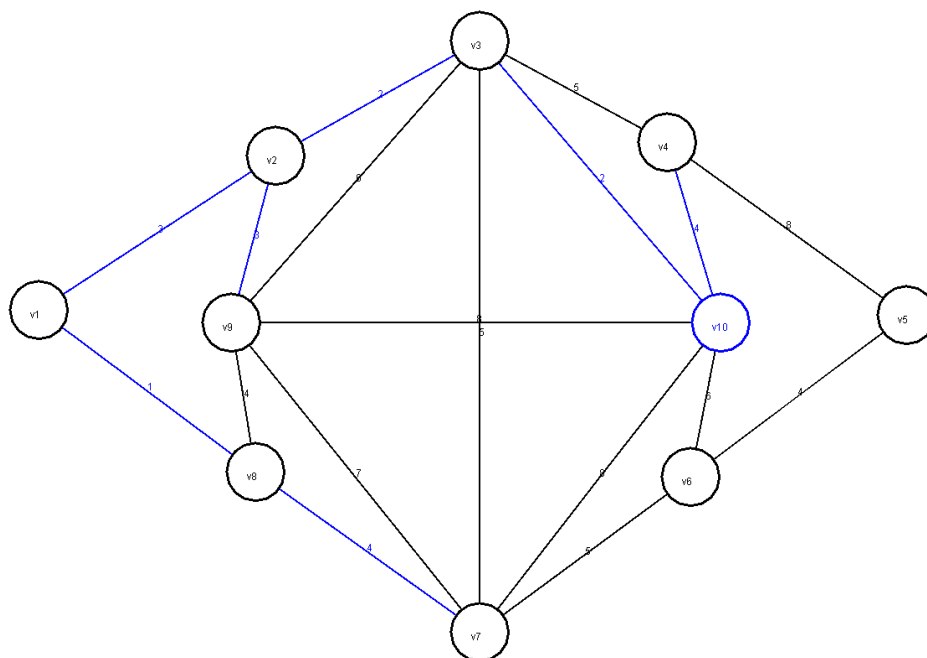
จะพบว่าเส้นเชื่อมจำนวน 2 เส้นที่มีน้ำหนักเป็น 4 คือเส้นเชื่อม v_4v_{10} และเส้นเชื่อม v_7v_8 ดังนั้นเลือกเส้นเชื่อม v_4v_{10} ที่สุด ดังนั้นต้นไม้แบบทอดข้าม T ก็จะเป็น $\{v_3v_{10}, v_2v_3, v_1v_2, v_8v_1, v_2v_9, v_4v_{10}\}$ ส่วน A เป็น $\{v_5, v_6, v_7\}$ และ N เป็น $\{v_{10}, v_3, v_2, v_1, v_8, v_9, v_4\}$ ซึ่งแสดงเป็นต้นไม้แบบทอดข้ามได้ดังนี้



รอบที่ 7 เนื่องจากนำจุดยอด v_4 ใส่ใน N ดังนั้นต้องมีการปรับค่าในตาราง โดยที่การเชื่อมต่อของจุด v_5 จะมีค่าเพิ่มขึ้นเป็น 8 ดังนั้นได้ตารางต่อไปนี้

	N	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
v10		∞	∞	2.0, v10	4.0, v10	∞	6.0, v10	8.0, v10	∞	8.0, v10	-
v3		∞	2.0, v3	-	4.0, v10	∞	6.0, v10	5.0, v3	∞	6.0, v3	-
v2		3.0, v2	-	-	4.0, v10	∞	6.0, v10	5.0, v3	∞	3.0, v2	-
v1		-	-	-	4.0, v10	∞	6.0, v10	5.0, v3	1.0, v1	3.0, v2	-
v8		-	-	-	4.0, v10	∞	6.0, v10	4.0, v8	-	3.0, v2	-
v9		-	-	-	4.0, v10	∞	6.0, v10	4.0, v8	-	-	-
v4		-	-	-	-	8.0, v4	6.0, v10	4.0, v8	-	-	-

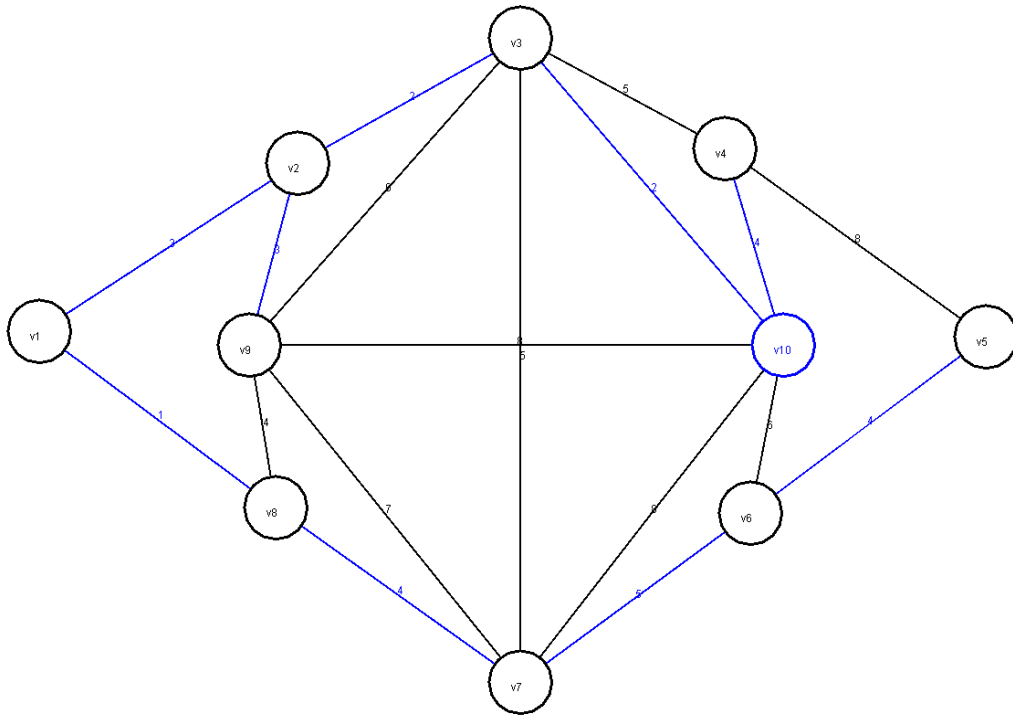
จากนั้นเลือกเส้นเชื่อม v_7v_8 เนื่องจากมีน้ำหนักน้อยที่สุด ดังนั้นต้นไม้แบบทอดข้าม T ก็จะเป็น $\{v_3v_{10}, v_2v_3, v_1v_2, v_8v_1, v_2v_9, v_4v_{10}, v_7v_8\}$ ส่วน A เป็น $\{v_5, v_6\}$ และ N เป็น $\{v_{10}, v_3, v_2, v_1, v_8, v_9, v_4, v_7\}$ ซึ่งแสดงเป็นต้นไม้แบบทอดข้ามได้ดังนี้



รอบที่ 8 เนื่องจากนำจุดยอด v_7 ใส่ใน N ดังนั้นต้องมีการปรับค่าในตาราง โดยที่การเชื่อมต่อของจุด v_6 จะมีค่าลดลงเป็น 5 ดังนั้นได้ตารางต่อไปนี้

	N	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
v10		∞	∞	2.0, v10	4.0, v10	∞	6.0, v10	8.0, v10	∞	8.0, v10	-
v3		∞	2.0, v3	-	4.0, v10	∞	6.0, v10	5.0, v3	∞	6.0, v3	-
v2		3.0, v2	-	-	4.0, v10	∞	6.0, v10	5.0, v3	∞	3.0, v2	-
v1		-	-	-	4.0, v10	∞	6.0, v10	5.0, v3	1.0, v1	3.0, v2	-
v8		-	-	-	4.0, v10	∞	6.0, v10	4.0, v8	-	3.0, v2	-
v9		-	-	-	4.0, v10	∞	6.0, v10	4.0, v8	-	-	-
v4		-	-	-	-	8.0, v4	6.0, v10	4.0, v8	-	-	-
v7		-	-	-	-	8.0, v4	5.0, v7	-	-	-	-

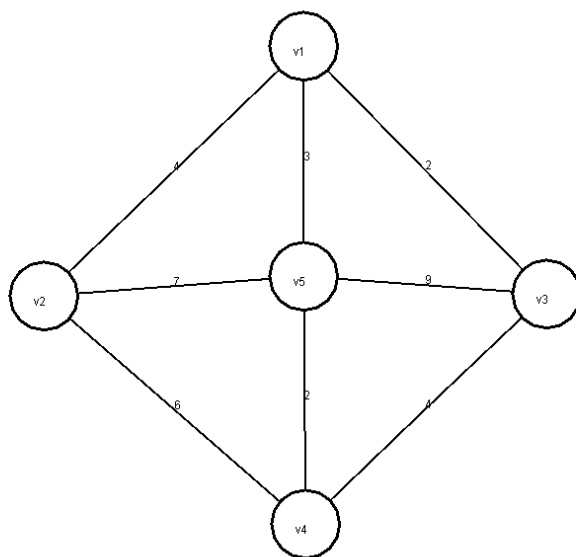
จากนั้นเลือกเส้นเชื่อม v_6v_7 เนื่องจากมีน้ำหนักน้อยที่สุด ดังนั้นต้นไม้แบบทอดข้าม T ก็จะเป็น $\{v_3v_{10}, v_2v_3, v_1v_2, v_8v_1, v_2v_9, v_4v_{10}, v_7v_8, v_6v_7\}$ ส่วน A เป็น $\{v_5\}$ และ N เป็น $\{v_{10}, v_3, v_2, v_1, v_8, v_9, v_4, v_7, v_6\}$ ซึ่งแสดงเป็นต้นไม้แบบทอดข้ามได้ดังนี้



N	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
v10	∞	∞	2.0, v10	4.0, v10	∞	6.0, v10	8.0, v10	∞	8.0, v10	-
v3	∞	2.0, v3	-	4.0, v10	∞	6.0, v10	5.0, v3	∞	6.0, v3	-
v2	3.0, v2	-	-	4.0, v10	∞	6.0, v10	5.0, v3	∞	3.0, v2	-
v1	-	-	-	4.0, v10	∞	6.0, v10	5.0, v3	1.0, v1	3.0, v2	-
v8	-	-	-	4.0, v10	∞	6.0, v10	4.0, v8	-	3.0, v2	-
v9	-	-	-	4.0, v10	∞	6.0, v10	4.0, v8	-	-	-
v4	-	-	-	-	8.0, v4	6.0, v10	4.0, v8	-	-	-
v7	-	-	-	-	8.0, v4	5.0, v7	-	-	-	-
v6	-	-	-	-	4.0, v6	-	-	-	-	-

Total weight of
minimum spanning
tree is : 28.0

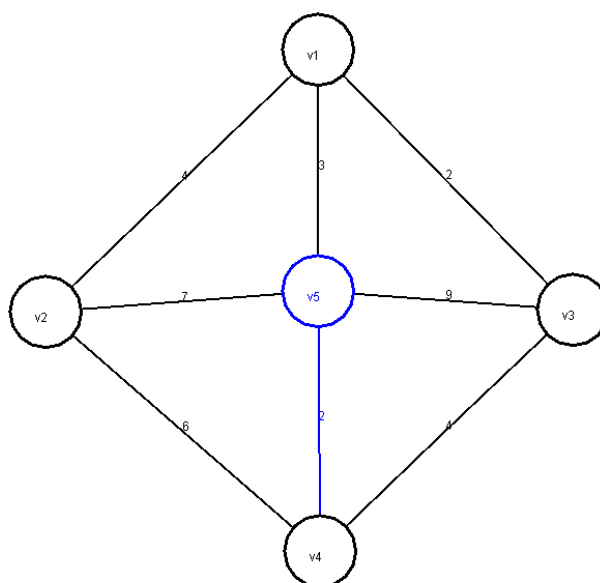
ตัวอย่างทดสอบแบบที่ 2



ให้จุดเริ่มต้นจากจุดยอด v_5 ต่อไปก็ให้ค่าเริ่มต้นกับข้อมูลที่จำเป็น ให้ N เป็น $\{v_5\}$ ให้ T เป็นเซตว่างและ A เป็น $\{v_1, v_2, v_3, v_4\}$ ดังนั้นจึงหาเส้นเชื่อมที่มีจุดปลายใน A ที่มาตกกระทบกับ v_5 ซึ่งเก็บได้เป็นตารางดังนี้

	N	v_1	v_2	v_3	v_4	v_5
v_5		3.0, v_5	7.0, v_5	9.0, v_5	2.0, v_5	-

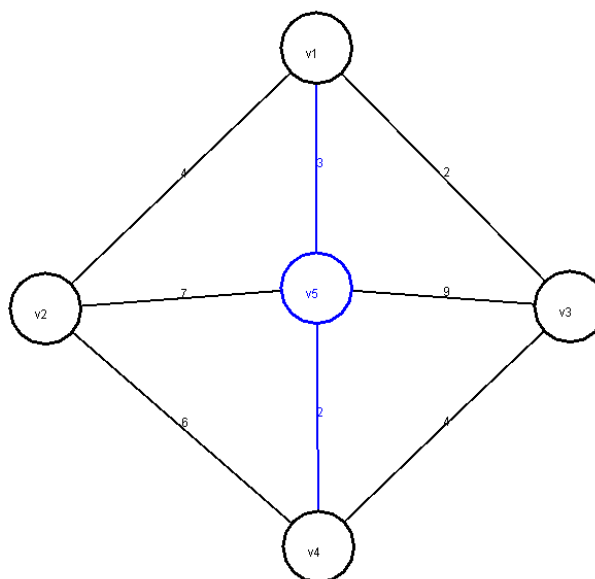
จากนั้นเลือกเส้นเชื่อมที่มีน้ำหนักน้อยสุดก็คือ v_4v_5 ใส่ลงไปใน T และเพิ่มจุดยอด N เป็น $\{v_5, v_4\}$ และ A คือ $\{v_1, v_2, v_3\}$ จะได้รูปต่อไปนี้



รอบที่ 2 จะมีการปรับค่าตำแหน่งน้อยที่สุดระหว่างจุดที่อยู่ในเซต A และเซต N เนื่องจากมีจุดใหม่ v_4 ในเซต N ขณะนี้จะเชื่อมเข้ามาผ่านจุดยอด v_4 คือจุดยอด v_2 และ v_3 เมื่อเชื่อมทางจุดยอด v_3 ก็จะมีน้ำหนักลดลง แต่จุดยอด v_1 เชื่อมกับ v_4 น้ำหนักไม่ลดลงก็จะไม่เปลี่ยนค่าในตาราง ซึ่งพิจารณาได้ในตารางต่อไปนี้

	N	v1	v2	v3	v4	v5
v5		3.0, v5	7.0, v5	9.0, v5	2.0, v5	-
v4		3.0, v5	6.0, v4	4.0, v4	-	-

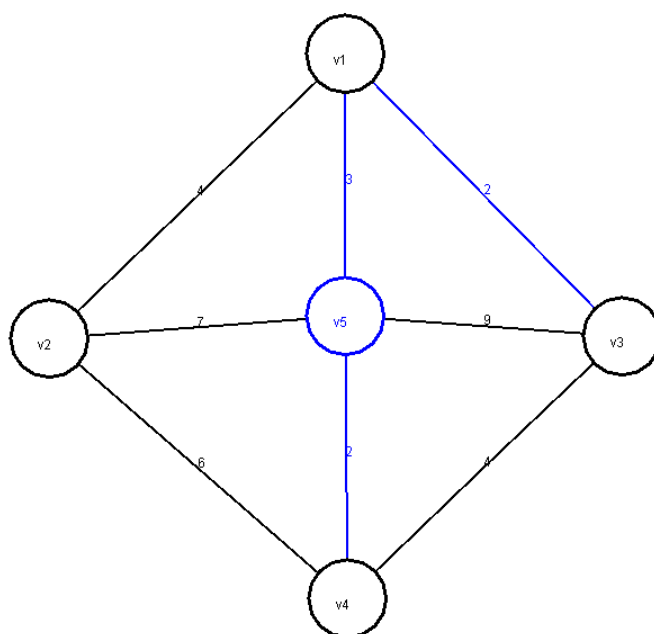
จากนั้นจึงหาเส้นเชื่อมที่มีน้ำหนักที่น้อยที่สุด ซึ่งก็คือ v_1v_5 จากนั้นก็ปรับข้อมูลโดยเพิ่ม v_1 ลงใน N พร้อมกันนั้นก็ลบ v_1 ออกจาก A และต้นไม้แบบทอดข้าม T เป็น $\{v_5v_4, v_1v_5\}$ โดย N เป็น $\{v_5, v_4, v_1\}$ และ A เป็น $\{v_2, v_3\}$ แสดงได้ดังรูปต่อไปนี้



รอบที่ 3 เมื่อนำจุดยอด v_1 ใส่ใน N แล้วจะต้องมีการปรับเปลี่ยนค่าน้ำหนักของการเชื่อมระหว่างเซต A และ N ใหม่ดังนี้

	N	v1	v2	v3	v4	v5
v5		3.0, v5	7.0, v5	9.0, v5	2.0, v5	-
v4		3.0, v5	6.0, v4	4.0, v4	-	-
v1		-	4.0, v1	2.0, v1	-	-

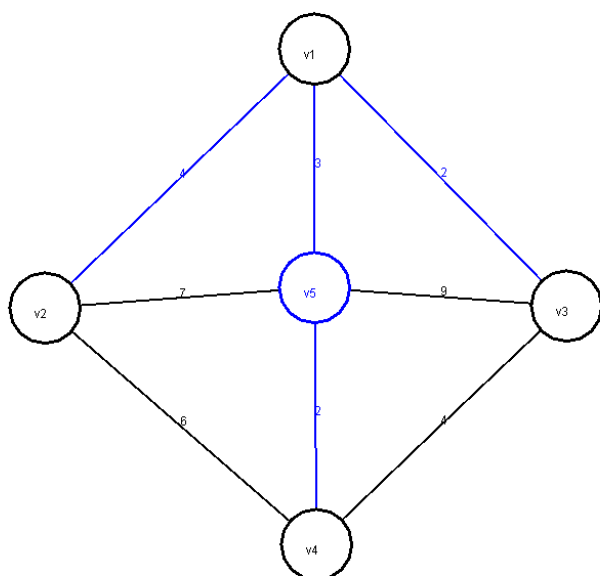
จากนั้นจึงหาเส้นเชื่อมที่มีน้ำหนักที่น้อยที่สุด ซึ่งก็คือ v_1v_3 จากนั้นก็ปรับข้อมูลโดยเพิ่ม v_3 ลงใน N พร้อมกันนั้นก็ลบ v_3 ออกจาก A และต้นไม้แบบทอดข้าม T เป็น $\{v_5v_4, v_1v_5, v_1v_3\}$ โดย N เป็น $\{v_5, v_4, v_1, v_3\}$ และ A เป็น $\{v_2\}$ แสดงได้ดังรูปต่อไปนี้



รอบที่ 4 เนื่องจากนำจุดยอด v_3 ใส่ใน N พบว่าไม่มีการปรับค่าตารางดังนี้

	N	v1	v2	v3	v4	v5
v5		3.0, v5	7.0, v5	9.0, v5	2.0, v5	-
v4		3.0, v5	6.0, v4	4.0, v4	-	-
v1		-	4.0, v1	2.0, v1	-	-
v3		-	4.0, v1	-	-	-

จากตารางพบว่าเส้นเชื่อม v_1v_2 เป็นเส้นเชื่อมเดียวที่เหลืออยู่ ดังนั้นต้นไม้แบบทอดข้าม ข้าม T ก็จะมี $\{v_5v_4, v_1v_5, v_1v_3, v_1v_2\}$ ส่วน A เป็น เซตว่าง และ N เป็น $\{v_5, v_4, v_1, v_3, v_2\}$ ซึ่งได้จำนวนเส้นเชื่อมครบแล้วก็ได้คำตอบดังรูปต่อไปนี้



Total weight of
minimum spanning
tree is : 11.0

อ้างอิง

รูปแบบโปรแกรม

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