

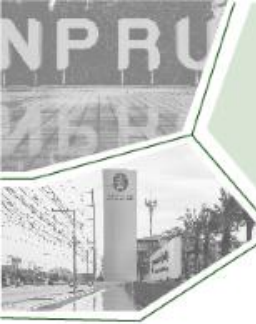


# ทฤษฎีกราฟ

ผู้ช่วยศาสตราจารย์สมเกียรติ ช่อเหมือน ([tko@webmail.npru.ac.th](mailto:tko@webmail.npru.ac.th))

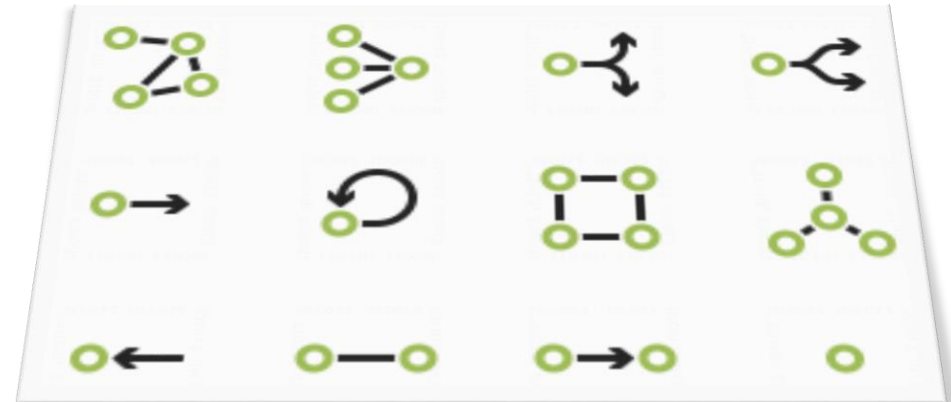
สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาศาสตร์และเทคโนโลยี





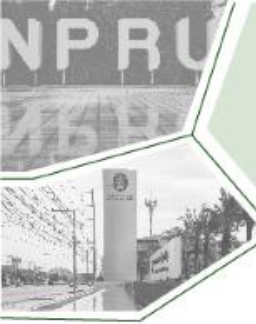
# เนื้อหาการเรียนรู้

- กราฟและส่วนประกอบของกราฟ
- การวาดแผนภาพกราฟ
- จุดยอดกราฟ
- เส้นเชื่อมกราฟ
- แนวเดินกราฟ



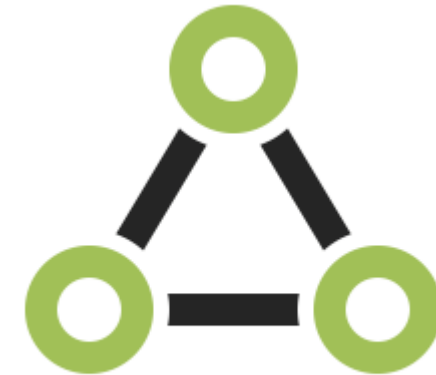
Graph





# ทฤษฎีกราฟ (Graph)

- ทฤษฎีกราฟในสาขาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
- คุณสมบัติต่าง ๆ ของกราฟ และแปลงปัญหาในรูปแบบกราฟ นำเสนอรูปภาพ
- จุดยอด (vertex) และเส้นเชื่อม (edge) การลากเส้นเชื่อมระหว่างจุด
- กราฟมีทิศทางจะใช้ลูกศรกำหนดทิศทาง และการจัดเก็บกราฟในระบบคอมพิวเตอร์
- โครงสร้างข้อมูลกราฟแบบรายการหรือเมทริกซ์ และขั้นตอนวิธีสำหรับประมวลผล
- การเข้าถึงจุดยอดของกราฟและจำแนกกราฟตามทฤษฎี



<https://www.shutterstock.com/th/image-vector/graph-vector-icon-282189968>

[https://www.iconexperience.com/o\\_collection/icons/?icon=graph\\_triangle](https://www.iconexperience.com/o_collection/icons/?icon=graph_triangle)



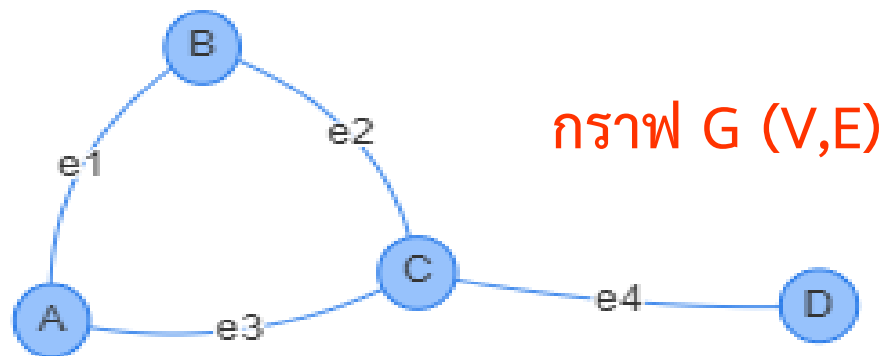


# ส่วนประกอบของกราฟ

• **บทนิยาม** กราฟ G ประกอบด้วย เซตจำกัด 2 เซต คือ

1. เซตที่ไม่เป็นเซตว่างของจุดยอด แทนด้วยสัญลักษณ์  $V(G)$
2. เซตของเส้นเชื่อมที่เชื่อมระหว่างจุดยอด แทนด้วยสัญลักษณ์  $E(G)$

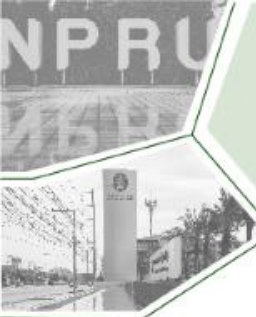
เส้นสมมุติหรือลูกศรสมมุติแสดงปริมาณและทิศทาง



$$V(G) = \{A, B, C, D\} \text{ และ } E(G) = \{e1, e2, e3, e4\}$$

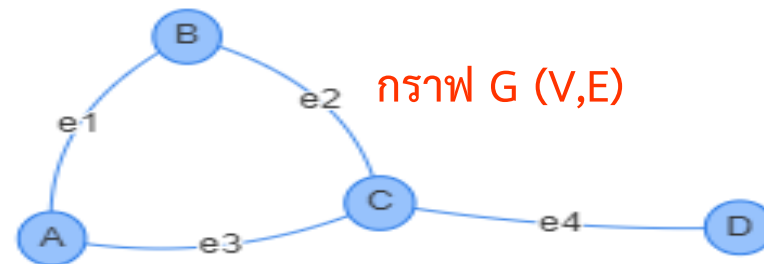
เมื่อนับจำนวนจุดยอด  $|V(G)| = 4$  จุดและจำนวนเส้นเชื่อม  $|E(G)| = 4$  เส้น คือ AB, BC, AC, CD





# การแสดงผลกราฟด้วยจาวาสคริปต์ไลบรารี vis.js

- การเขียนโปรแกรมภาษาจาวาสคริปต์และภาษา html



## ส่วนที่ 1 การแสดงผลกราฟและการเรียกใช้ vis.js

```
<!doctype html>
<html>
<head>
  <title>Network</title>
  <script src="https://unpkg.com/vis-
network/standalone/umd/vis-network.min.js">
</script>
  <style type="text/css">
    #mynetwork {
      width: 600px;
      height: 400px;
      border: 1px solid lightgray;
    }
  </style>
</head>
<body>
<div id="mynetwork"></div>
</body>
</html>
```

## ส่วนที่ 2 การจัดเตรียมชุดข้อมูล

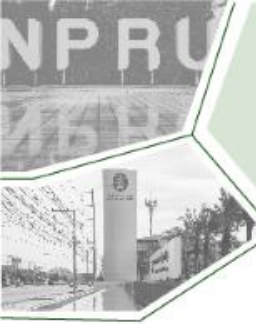
```
<script type="text/javascript">
  // create an array with nodes
  var nodes = new vis.DataSet([
    {id: 1, label: 'A'},
    {id: 2, label: 'B'},
    {id: 3, label: 'C'},
    {id: 4, label: 'D'}
  ]);
  // create an array with edges
  var edges = new vis.DataSet([
    {from: 1, to: 2, label:'e1'},
    {from: 2, to: 3, label:'e2'},
    {from: 3, to: 1, label:'e3'},
    {from: 3, to: 4, label:'e4'}
  ]);
</script>
```

## ส่วนที่ 3 การสร้างกราฟ

```
<script type="text/javascript">
  // create a network
  var container = document.getElementById('mynetwork');
  var data = {
    nodes: nodes,
    edges: edges
  };
  var options = {};
  var network = new vis.Network(container, data, options);
</script>
```

[https://www.iconfinder.com/icons/51264/js\\_icon](https://www.iconfinder.com/icons/51264/js_icon)



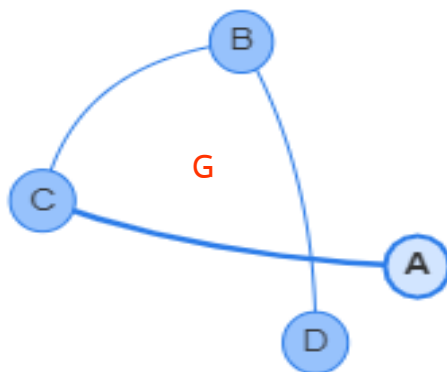


## การวาดแผนภาพกราฟ

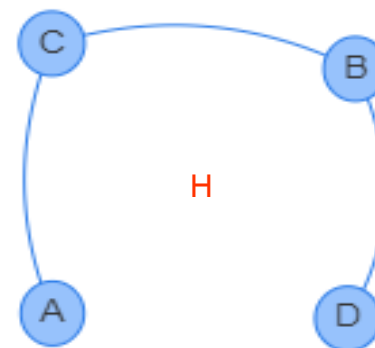
- การเขียนแผนภาพของกราฟ
- กำหนดตำแหน่งของจุดยอด ณ ตำแหน่งใดก็ได้
- ลากเส้นเชื่อมต่อจุดยอดของกราฟเป็นเส้นตรงหรือเส้นโค้งยาวเท่าใดก็ได้
- ไม่ลากเส้นตัดกับตัวเอง และไม่ลากผ่านจุดยอดที่ไม่ใช่จุดยอดของเส้นนั้น



กราฟถอดแบบ G และ H

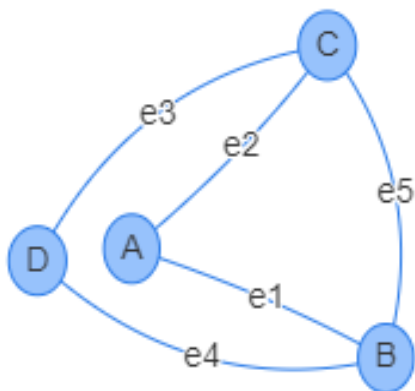


กราฟเดียวกัน





# กราฟเดียวกัน



G

เซตของจุดยอดเท่ากัน

เซตของเส้นเชื่อมต่อเท่ากัน

การวาดกราฟและปรับเปลี่ยนตำแหน่งจุดยอด

```
// create an array with nodes
```

```
var nodes = new vis.DataSet([
```

```
  {id: 1, label: 'A'},
```

```
  {id: 2, label: 'B'},
```

```
  {id: 3, label: 'C'},
```

```
  {id: 4, label: 'D'}]
```

```
]);
```

```
// create an array with edges
```

```
var edges = new vis.DataSet([
```

```
  {from: 1, to: 2,label:'e1'},
```

```
  {from: 1, to: 3,label:'e2'},
```

```
  {from: 3, to: 4,label:'e3'},
```

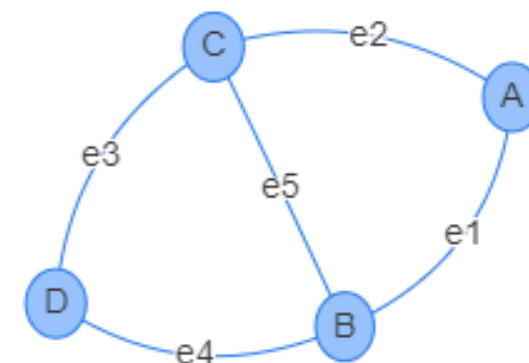
```
  {from: 2, to: 4,label:'e4'},
```

```
  {from: 2, to: 3,label:'e5'}]
```

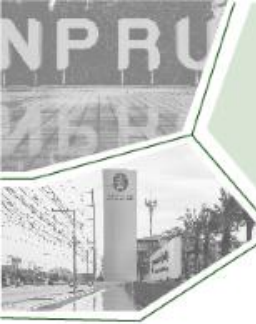
```
]);
```



vis.js

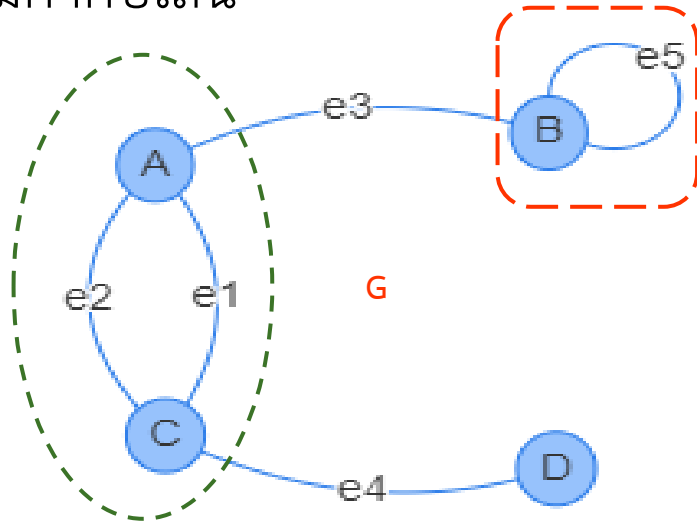


G



## จุดยอดคู่เดียวกัน

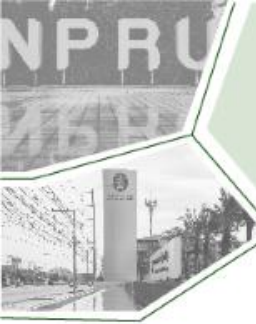
- โหนดและเส้นเชื่อมที่มีข้อความกำกับเส้น



- เส้นเชื่อม **e1** และ **e2** เป็นเส้นเชื่อมต่อระหว่างจุดยอดคู่เดียวกัน จุดยอด A และ จุดยอด C
- เส้นเชื่อม **e5** เป็นเส้นเชื่อมที่เชื่อมจุดยอด B เพียงจุดเดียว

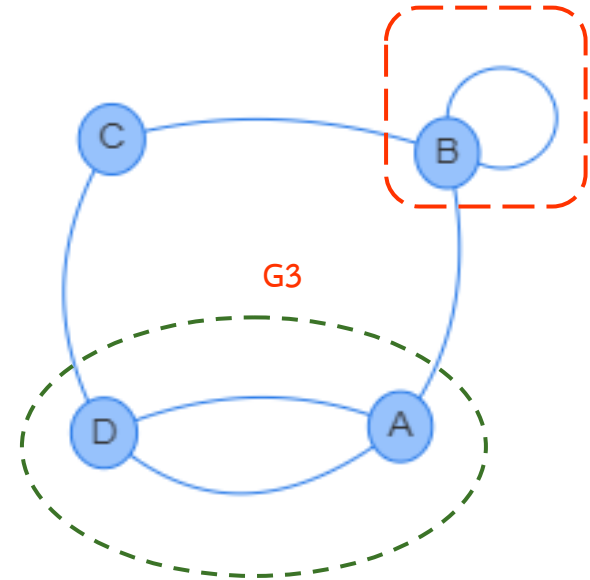
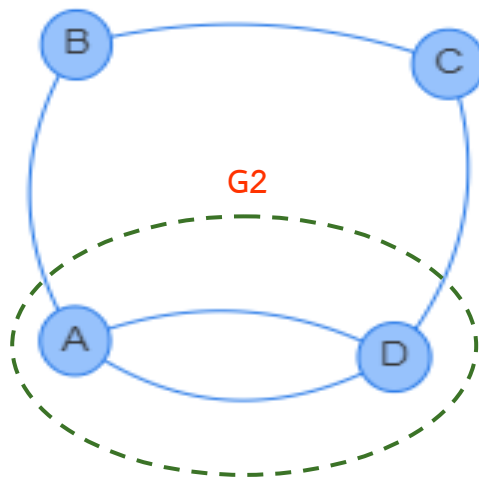
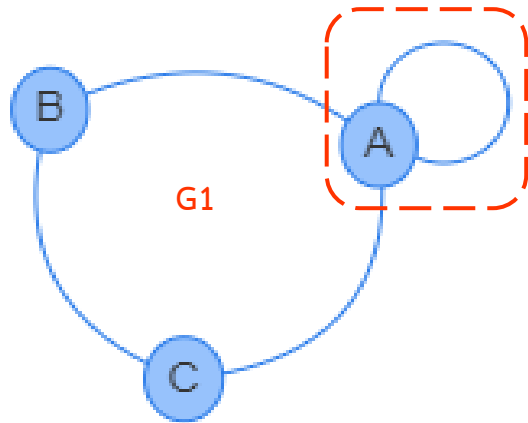






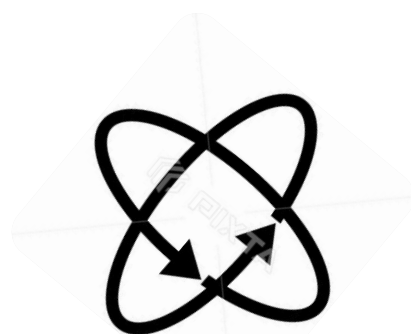
# เส้นเชื่อมขนานและเส้นวงวน

- **บทนิยาม** เส้นเชื่อมตั้งแต่ 2 เส้นที่เชื่อมระหว่างจุดยอดคู่เดียวกัน เรียกว่า **เส้นเชื่อมขนาน (Parallel Edges)** เส้นเชื่อมที่เชื่อมจุดยอดเพียงจุดเดียว เรียกว่า **วงวน (Loop)**



## จุดยอดกราฟ

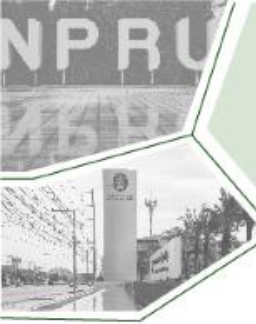
- เซตของจำนวนหรือค่าที่สนใจที่ใช้เป็นโหนด
- การนับจำนวนจุดยอดและจำนวนเส้นเชื่อม ทราบว่าผ่านได้หรือผ่านไม่ได้ตามวิธีการเดิน
- จุดยอดประชิด ติดกัน
- ดีกรีของจุดยอด จำนวนเส้นเชื่อมต่อจุดยอด
- ผลรวมดีกรีของจุดยอด



[https://th.pngtree.com/freepng/nodes-icon-for-your-project\\_4851159.html](https://th.pngtree.com/freepng/nodes-icon-for-your-project_4851159.html)

[https://th.pngtree.com/freepng/walking-icon-in-trendy-style-isolated-background\\_5001673.html](https://th.pngtree.com/freepng/walking-icon-in-trendy-style-isolated-background_5001673.html)

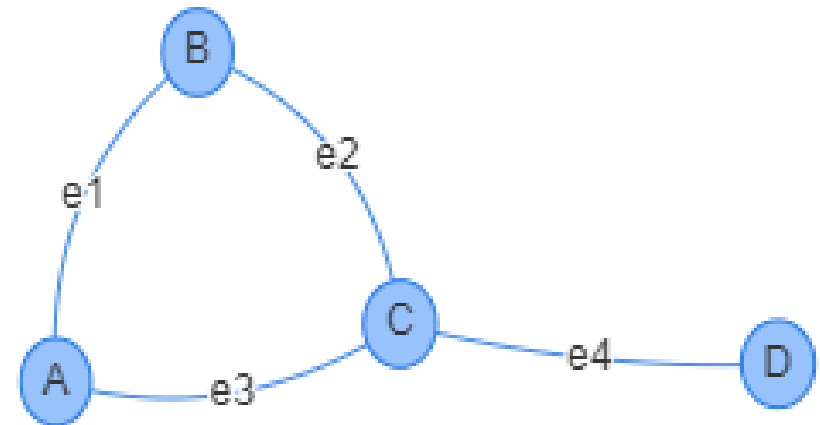
[https://www.flaticon.com/free-icon/the-sum-of\\_261941](https://www.flaticon.com/free-icon/the-sum-of_261941)



## จุดยอดประชิด

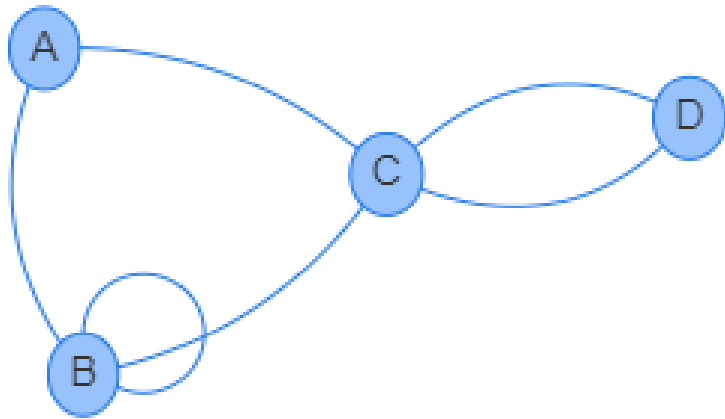
- **บทนิยาม** จุดยอด  $u$  และจุดยอด  $v$  ของกราฟ เป็น**จุดยอดประชิด (Adjacent Vertices)** ก็ต่อเมื่อมีเส้นเชื่อมระหว่างจุดทั้งสอง และจุดยอด  $u$  และ  $v$  เป็น**จุดปลาย (End Point)** ของเส้นเชื่อนั้น เส้นเชื่อม  $e$  ของกราฟ เกิดกับ (Incident) จุดยอด  $v$  ถ้าจุดยอด  $v$  เป็นจุดปลายจุดหนึ่งของเส้นเชื่อม

- 1) จุดยอด A และจุดยอด B เป็นจุดยอดประชิด
- 2) จุดยอด A และจุดยอด C เป็นจุดยอดประชิด
- 3) จุดยอด B และจุดยอด C เป็นจุดยอดประชิด
- 4) จุดยอด A และจุดยอด D ไม่เป็นจุดยอดประชิด



## ดีกรีของจุดยอด

- **บทนิยาม** ดีกรี (Degree) ของจุดยอดในกราฟ คือ จำนวนครั้งทั้งหมดที่เส้นเชื่อมเกิดกับจุดยอด ใช้สัญลักษณ์  $\text{deg } v$  แทนดีกรีของจุดยอด



จุดยอด (v)	จำนวนครั้งทั้งหมดที่เส้นเชื่อมเกิดกับจุดยอด v
A(1)	2
B(2)	4
C(3)	4
D(4)	2

[https://www.iconfinder.com/icons/51264/js\\_icon](https://www.iconfinder.com/icons/51264/js_icon)

<https://visjs.org/>

ตัวอย่างการโปรแกรมสร้างกราฟด้วยไลบรารี **vis.js**

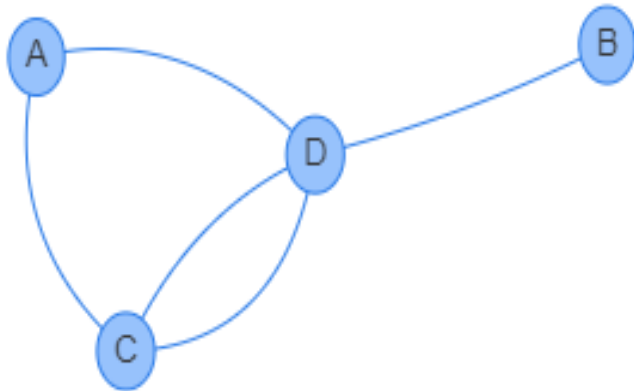
```
// create an array with nodes
var nodes = new vis.DataSet([
  {id: 1, label: 'A'},
  {id: 2, label: 'B'},
  {id: 3, label: 'C'},
  {id: 4, label: 'D'}
]);

// create an array with edges
var edges = new vis.DataSet([
  {from: 1, to: 2},
  {from: 1, to: 3},
  {from: 2, to: 3},
  {from: 2, to: 2},
  {from: 3, to: 4},
  {from: 4, to: 3}
]);
```



## ผลรวมดีกรีของจุดยอด

- จากเซตของเส้นเชื่อมต่อดังกล่าวและถึงสามารถนับจำนวนได้ตามตารางที่กำหนด
- เมื่อทดลองนับค่าดีกรีของจุดยอดของกราฟ



จุดยอด (v)	จำนวนครั้งทั้งหมดที่เส้นเชื่อมเกิดกับจุดยอด v
A(1)	2
B(2)	1
C(3)	3
D(4)	4
ผลรวม	10

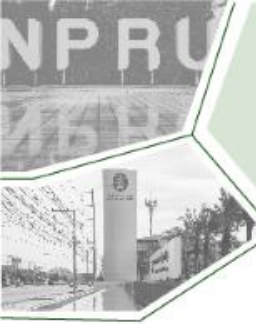
[https://www.iconfinder.com/icons/51264/js\\_icon](https://www.iconfinder.com/icons/51264/js_icon) <https://visjs.org/>

ตัวอย่างที่ 2 การโปรแกรมสร้างกราฟด้วยไลบรารี vis.js

```
// create an array with nodes
var nodes = new vis.DataSet([
  {id: 1, label: 'A'},
  {id: 2, label: 'B'},
  {id: 3, label: 'C'},
  {id: 4, label: 'D'}
]);

// create an array with edges
var edges = new vis.DataSet([
  {from: 1, to: 3},
  {from: 1, to: 4},
  {from: 3, to: 4},
  {from: 4, to: 3},
  {from: 4, to: 2}
]);
```





# เส้นเชื่อมกราฟ

- เส้นเชื่อมระหว่างจุดยอด การวาดเส้นความสัมพันธ์ของกราฟและการนับค่าดีกรี
- การเดินตามวิถีกราฟเพื่ออธิบายการเปลี่ยนแปลงและการเข้าถึงโหนดตามทฤษฎี
  - ทิศทาง (Direction)
  - ค่าน้ำหนัก (Weight)
    - ขนาดเส้น
      - สี
      - รูปแบบเส้น
    - ความยาว



# เส้นเชื่อมกราฟ

จุดยอด (v)	จำนวนครั้งทั้งหมดที่เส้นเชื่อมเกิดกับจุดยอด v
A	2
B	3
C	5
D	4
ผลรวม	14

สังเกตว่า  $\text{deg A} + \text{deg B} + \text{deg C} + \text{deg D} = 14$

กราฟมีจำนวนเส้นเชื่อมต่อทั้งหมด 7 เส้น

<https://visjs.org/>

[https://www.iconfinder.com/icons/51264/js\\_icon](https://www.iconfinder.com/icons/51264/js_icon)

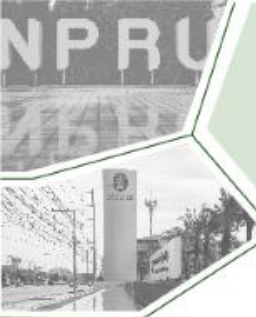
ตัวอย่างการโปรแกรมสร้างกราฟ  $G(V,E)$

```
// create an array with nodes
var nodes = new vis.DataSet([
  {id: 1, label: 'A'},
  {id: 2, label: 'B'},
  {id: 3, label: 'C'},
  {id: 4, label: 'D'}
]);

// create an array with edges
var edges = new vis.DataSet([
  {from: 1, to: 3},
  {from: 1, to: 4},
  {from: 3, to: 4},
  {from: 4, to: 3},
  {from: 3, to: 3},
  {from: 4, to: 2},
  {from: 2, to: 2}
]);
```



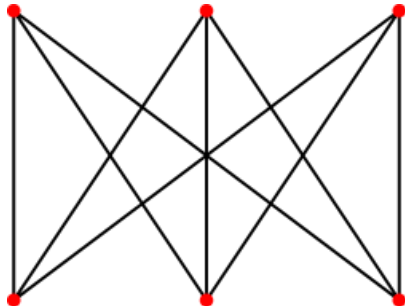
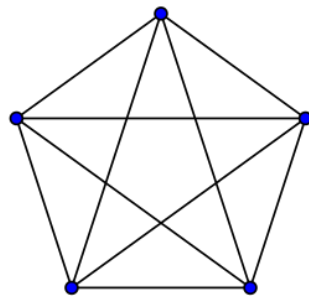
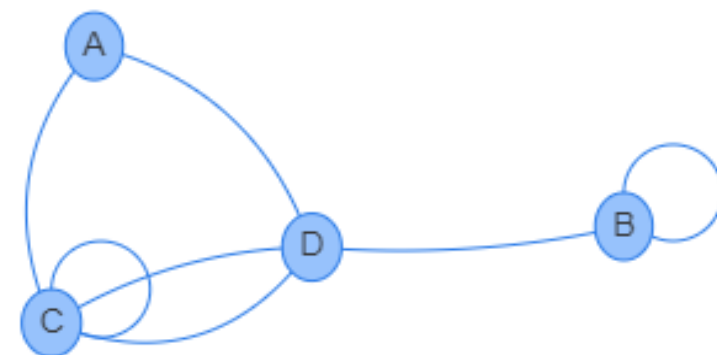




# ผลรวมดีกรีจุดยอดและจำนวนเส้นเชื่อม

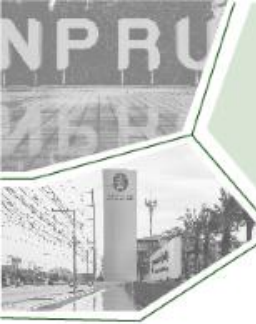
- **ทฤษฎีบท** ให้  $u_1, u_2, u_3, \dots, u_n \in V(G)$  เป็นจุดยอดทั้งหมดในกราฟ  $G$
- **ตัวอย่าง** การหาจำนวนเส้นเชื่อมของกราฟ ผลรวมของดีกรีจุดยอด
- ทุกจุดเท่ากับ **22** จากทฤษฎีบทตั้งนั้น  **$22 = 2n$**  นั่นคือ  **$n = 11$**
- กราฟมีเส้นเชื่อม 11 เส้น
- ผลรวมของดีกรีจุดยอดและจำนวนเส้นเชื่อมต่อจากกราฟ  $G$

$$\sum_{i=1}^{|V(G)|} \text{deg } u_i = 2|E(G)|$$



<https://th.wikipedia.org/wiki/กราฟเชิงระนาบ>



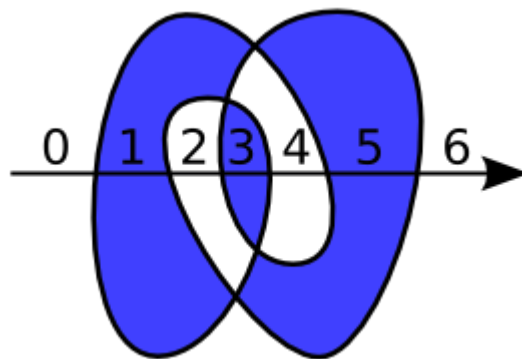


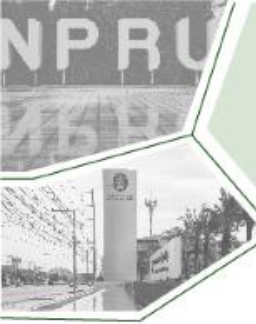
## ผลรวมของดีกรีของจุดยอด

- **ทฤษฎีบท 1** ผลรวมของดีกรีของจุดยอดทุกจุดในกราฟเท่ากับสองเท่าของจำนวนเส้นเชื่อม  
บทนิยาม

จุดยอดที่มีดีกรีเป็นจำนวนคู่ เรียกว่า **จุดยอดคู่ (Even Vertex)**

จุดยอดที่มีดีกรีเป็นจำนวนคี่ เรียกว่า **จุดยอดคี่ (Odd Vertex)**





# การเชื่อมจุดยอดกราฟ

- **ทฤษฎีบท 2** ทุกกราฟจะมีจุดยอดที่เป็นจำนวนคู่

**บทนิยาม** ให้  $u$  และ  $v$  เป็นจุดยอดของกราฟแนวเดิน

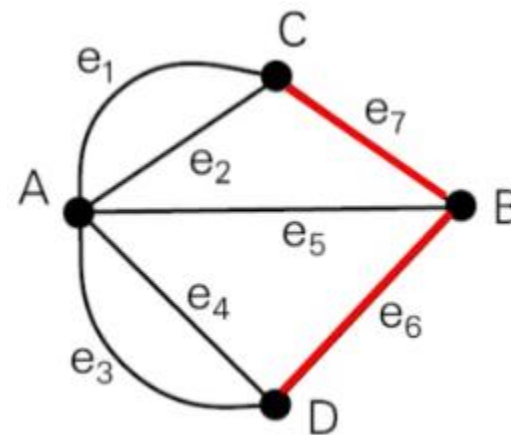
$u - v$  ( $u - v$  walk) คือ ลำดับจำกัดของจุดยอดและเส้นเชื่อมสลับกัน

$$u = u_0, e_1, u_1, e_2, u_2, \dots, u_{n-1}, e_n, u_n = v$$

โดย**เริ่มต้น**ที่จุดยอด  $u$  และ**สิ้นสุด**ที่จุดยอด  $v$  และแต่ละเส้นเชื่อม  $e_i$  จะเกิดกับจุดยอด  $u_{i-1}$  และ  $u_i$  เมื่อ  $i \in \{1, 2, \dots, n\}$

แนวเดิน

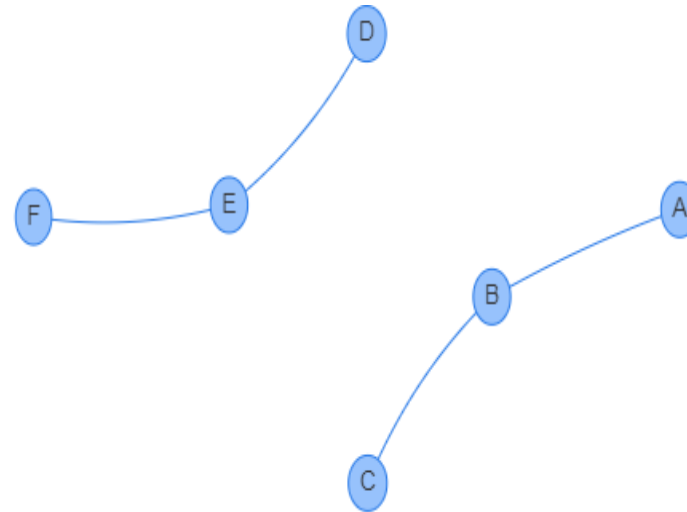
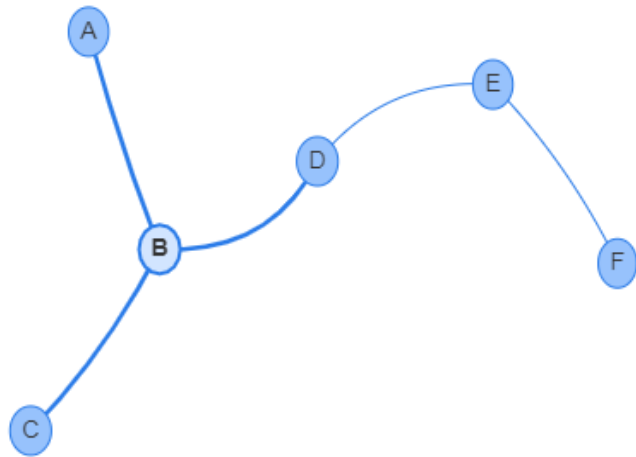
$C, e_7, B, e_6, D$

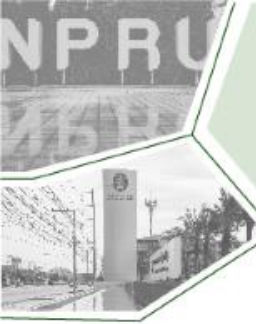




## กราฟเชื่อมโยง

- **บทนิยาม** กราฟ  $G$  เป็น**กราฟเชื่อมโยง (connected graph)** ถ้าจุดยอด 2 จุดใดๆ ใน  $G$  เชื่อมได้ด้วยวิถีจะมีแนวเดิน  $u - v$





# แนวเดินกราฟเชื่อมโยง

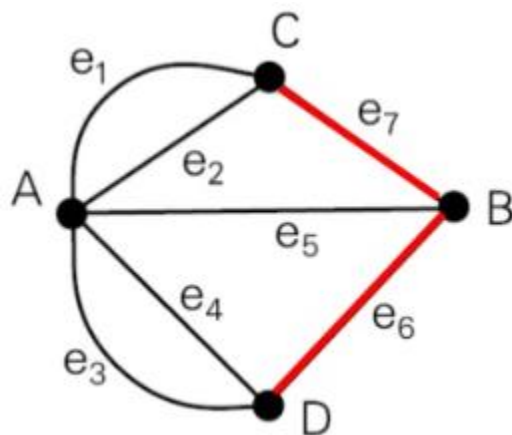
- บทนิยาม

รอยเดิน (trail) คือ แนวเดินในกราฟที่เส้นเชื่อมทั้งหมดแตกต่างกัน

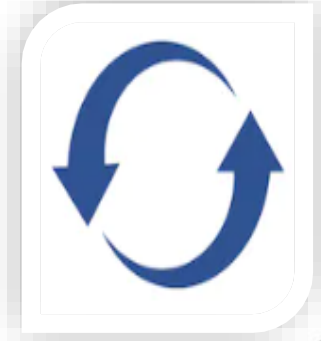
วิถี (Path) คือ แนวเดินในกราฟที่จุดยอดทั้งหมดแตกต่างกัน

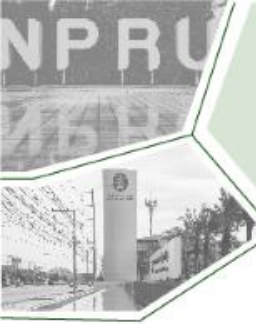
วงจร (Circuit) คือ แนวเดินที่เส้นเชื่อมทั้งหมดแตกต่างกัน โดยมีจุดเริ่มต้นและจุดสุดท้ายเป็นจุดยอดเดียวกัน

วัฏจักร (Cycle) คือวงจรที่ไม่มีจุดยอดซ้ำกัน ยกเว้นจุดเริ่มต้นและจุดสุดท้าย



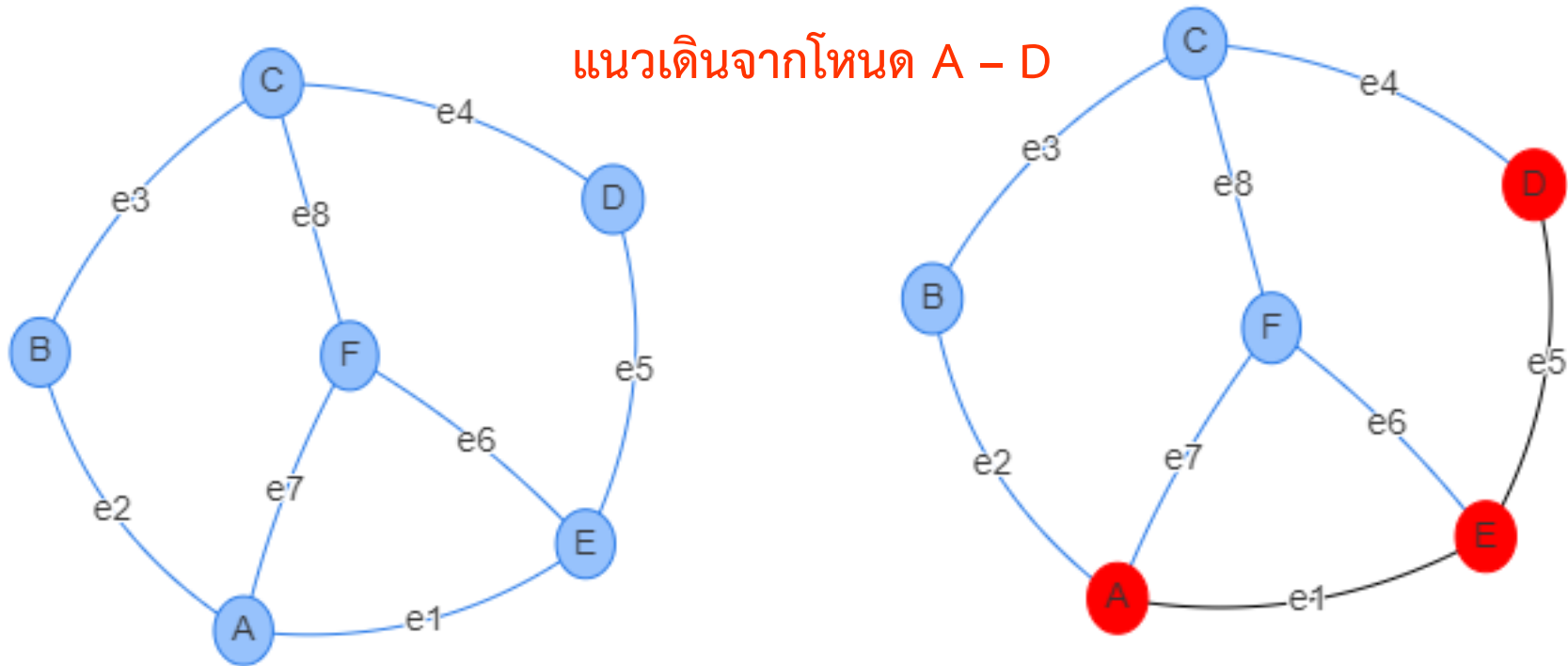
## รูปแบบแนวเดิน

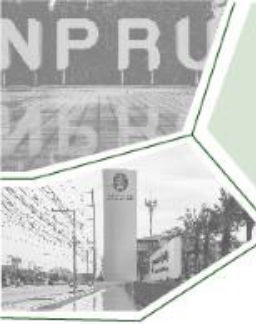




# การวิเคราะห์แนวเดิน

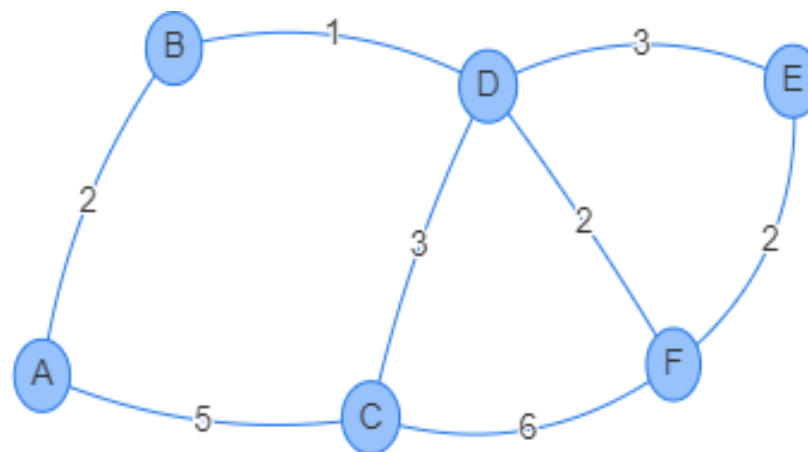
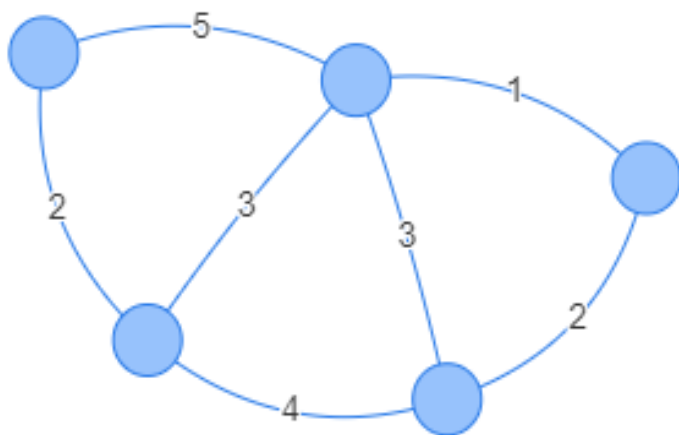
- การแก้ปัญหาการเดินทางจากแผนผังของเมืองหนึ่ง
- โดยให้จุดยอดแทนอำเภอ และเส้นเชื่อมแทนถนนที่เชื่อมระหว่างสองอำเภอ



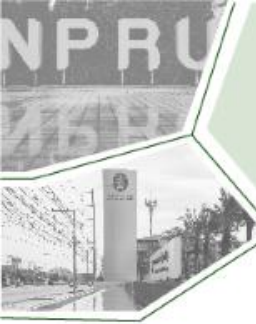


# ค่าน้ำหนัก

- **บทนิยาม** ค่าน้ำหนัก(weight) ของเส้นเชื่อม  $e$  ในกราฟ คือ จำนวนที่ไม่เป็นลบที่กำหนดไว้บนเส้นเชื่อม  $e$  กราฟถ่วงน้ำหนัก(weight graph) คือ กราฟที่เส้นเชื่อมทุกเส้นมีค่าน้ำหนัก

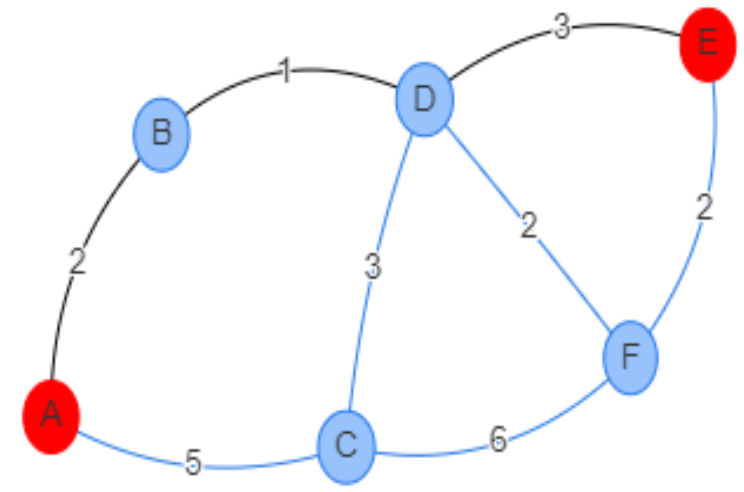






# วิธีที่สั้นที่สุด

- **บทนิยาม** วิธีที่สั้นที่สุด จากจุด A ถึงจุดยอด Z ในกราฟถ่วงน้ำหนัก คือวิธี A - Z ที่ผลรวมของค่าน้ำหนักของเส้นเชื่อมทุกเส้นในวิธี A-Z น้อยที่สุด



ลำดับที่	เส้นทาง	ระยะทางยาว	รวม (ก.ม)
เส้นทางที่ 1	A, B, D, E	$2 + 1 + 3$	6
เส้นทางที่ 2	A, B, D, F, E	$2 + 1 + 2 + 2$	7
เส้นทางที่ 3	A, B, D, C, F, E	$2 + 1 + 3 + 6 + 2$	14
เส้นทางที่ 4	A, C, F, E	$5 + 6 + 2$	13
เส้นทางที่ 5	A, C, F, D, E	$5 + 6 + 2 + 3$	16
เส้นทางที่ 6	A, C, D, E	$5 + 3 + 3$	11
เส้นทางที่ 7	A, C, D, F, E	$5 + 3 + 2 + 2$	12



