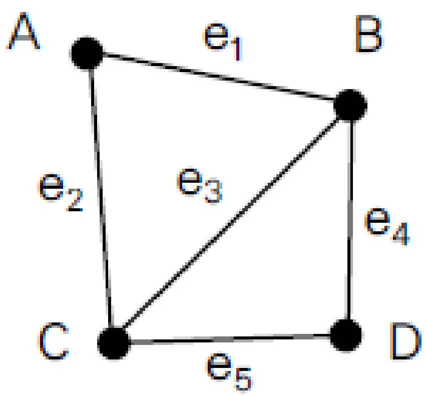
Sira Songpolrojjanakul

Graph Theory I
Basic Definition

กราฟ (Graph)

- กราฟ (Graph: G)ประกอบด้วย เซต 2 เซต คือ
  - จุดยอด (Vertex: V)
  - เส้นเชื่อม (Edge: E)



V(G) = {A,B,C,D}

E(G) = {AB,BD,CD,AC,BC}

or E(G) = $\{e_1,e_2,e_3,e_4,e_5\}$

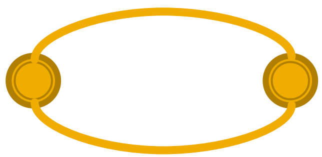
! กราฟ ต้องมีจุดยอด**อย่างน้อยหนึ่งจุด** แต่กราฟจะไม่มีเส้นเชื่อมเลยก็ได้

กราฟ (Graph)

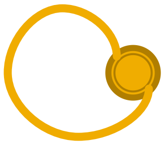
- กราฟที่มี จุดยอด หรือ เส้นเชื่อม **ไม่จำกัด** เรียก กราฟไม่จำกัด (infinite graph)
- กราฟที่มี จุดยอด และ เส้นเชื่อม **จำกัด** เรียก กราฟจำกัด (finite graph)
- กราฟที่**ไม่มี** เส้นเชื่อม เรียก กราฟว่าง (null graph/empty graph)
- ขนาดของกราฟ (size) = จำนวนเส้นเชื่อม
- อันดับของกราฟ (order) = จำนวนจุดยอด

ลักษณะของเส้นเชื่อม (Edge Type)

- เส้นเชื่อมขนาน (Parallel Edge)
  - คือ เส้นเชื่อมที่มีจุดเริ่มและจุดจบ เป็นคู่เดียวกัน



- วงวน (Loop)
  - คือ เส้นเชื่อมที่มีจุดเริ่มและจุดจบ เป็นจุดเดียวกัน



- เส้นเชื่อมไม่มีทิศทาง (Undirected Edge)
  - คือ เส้นเชื่อมที่สามารถ **เดินได้ทั้งสองทิศทาง**

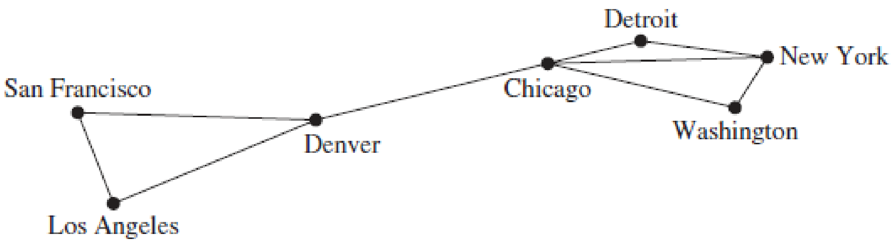
ไปได้ทั้ง u->v / v->u

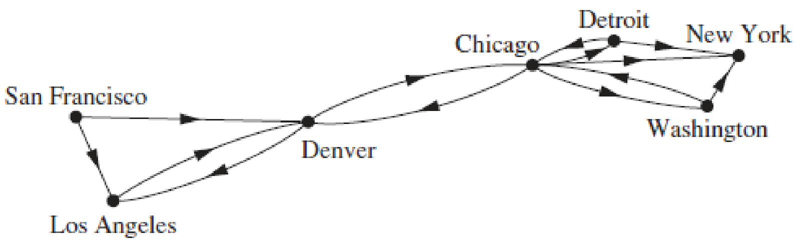
- เส้นเชื่อมมีทิศทาง (Directed Edge)
  - คือ เส้นเชื่อมที่สามารถ**เดินได้ทิศทางเดียว**

ไปได้แต่ u->v

ลักษณะของกราฟ จำแนกตามทิศทางเส้นเชื่อม

- กราฟไม่มีทิศทาง (undirected graph)
  - คือ กราฟที่ประกอบจากเส้นเชื่อมไม่มีทิศทาง
- กราฟมีทิศทาง (directed graph / digraph)
  - คือ กราฟที่ประกอบจากเส้นเชื่อมมีทิศทาง





แปลง undirected เป็น directed ได้โดย ใส่เส้นเชื่อมมีทิศทางทั้งไปกลับ (u,v)/(v,u)

ลักษณะของกราฟ ตามลักษณะเส้นเชื่อม

TABLE 1 Graph Terminology.

Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Simple graph = กราฟอย่างง่าย

คำศัพท์ (Terminology)

- จุดยอด u **ติดกับ** (adjacent) จุดยอด v ถ้า u กับ v มีเส้นเชื่อมไปถึง
- เส้นเชื่อม e **เกิดกับ** (incedent) จุดยอด u ถ้า e นั้น ติดกับ u

u ติดกับ v / v ติดกับ u

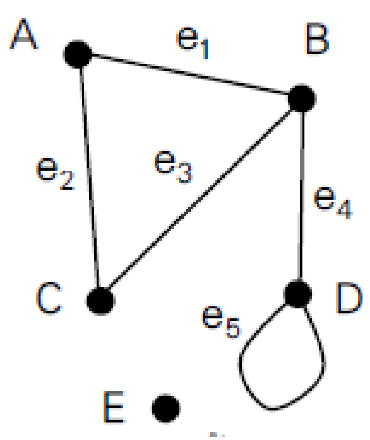
e1 เกิดกับ v / e1 เกิดกับ u

** นิยามบน undirected graph

คำศัพท์ (Terminology) II

- เพื่อนบ้าน (Neighborhood) ของ v $[N(v)]$ คือ เซตของจุดยอดทั้งหมดที่ติดกับ v
  - N(A) เมื่อ A เป็นสับเซตของ V คือ $\bigcup_{i=1}^{n} N(v_i)$
- ดีกรี (Degree) ของ v $[\deg(v)]$ คือ จำนวนเส้นเชื่อมที่เกิดกับ v (แต่ให้นับ loop เป็น 2 เส้น)

** นิยามบน undirected graph



N(A) = {B,C}	deg(A) = 2
N(B) = {A,C,D}	deg(B) = 3
N(C) = {A,B}	deg(C) = 2
N(D) = {B,D}	deg(D) = 3
N(E) = ϕ	deg(E) = 0

คำศัพท์ (Terminology) III

- สำหรับจุดยอด v ถ้า
- deg(v) = 0 เรียกว่า จุดยอดโดดเดี่ยว (Isolate)
- deg(v) = 1 เรียกว่า จุดยอด Pendant
- deg(v) เป็นเลขคู่ เรียกว่า จุดยอดคู่ (Even Vertex)
- deg(v) เป็นเลขคี่ เรียกว่า จุดยอดคี่ (Odd Vertex)

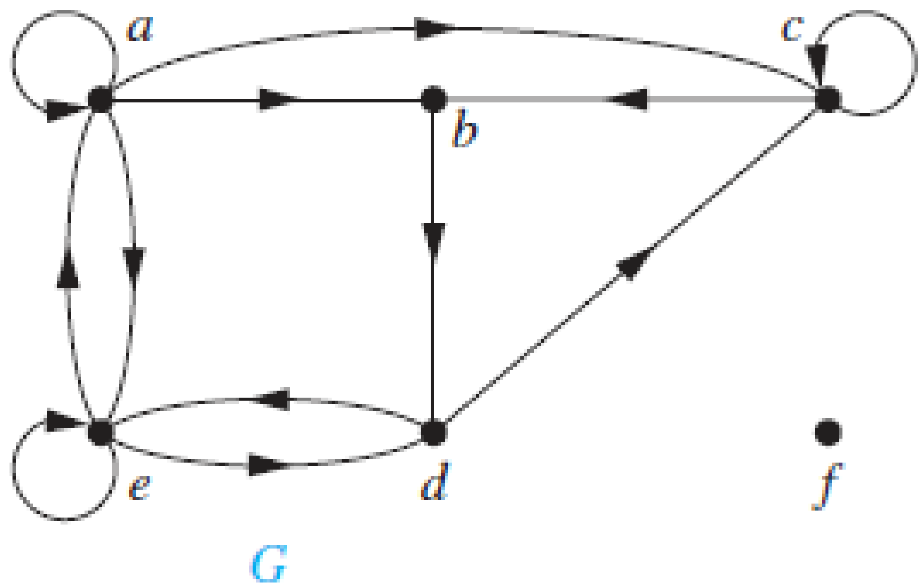
คำศัพท์ (Terminology) IV

- เส้นเชื่อมมีทิศทาง (u,v) จะเรียก u ว่าจุดเริ่ม (Initial Vertex) และเรียก v ว่าจุดปลาย (Terminal Vertex / End Vertex)
- เรียก u ติดกับ (adjacent to) v และ v อยู่ถัดมาจาก (adjacent from) u
- in-degree ของจุดยอด v $[\deg^{-}(v)]$ คือจำนวนเส้นเชื่อมที่พุ่งเข้า v
- out-degree ของจุดยอด v $[\deg^{+}(v)]$ คือจำนวนเส้นเชื่อมที่พุ่งออกจาก v

u ติดกับ v

ตัวอย่าง

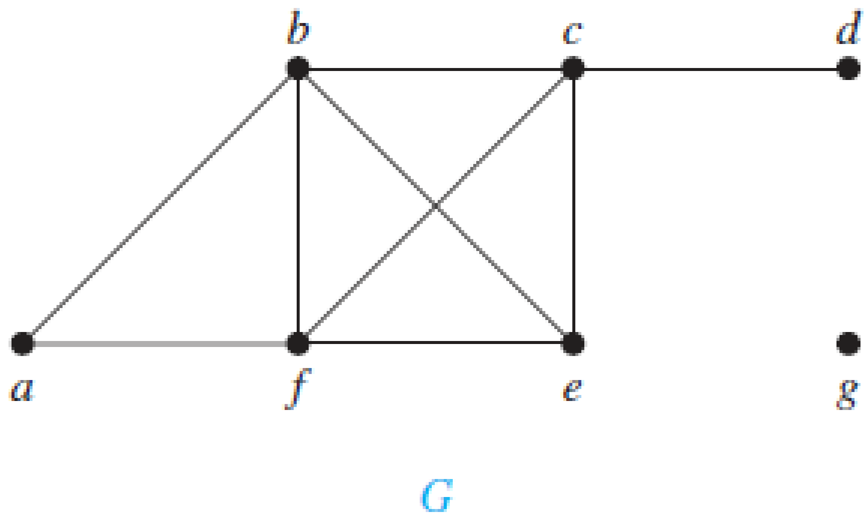
หา in/out degree ของกราฟ G



Solution: The in-degrees in G are $\deg^-(a) = 2$, $\deg^-(b) = 2$, $\deg^-(c) = 3$, $\deg^-(d) = 2$, $\deg^-(e) = 3$, and $\deg^-(f) = 0$. The out-degrees are $\deg^+(a) = 4$, $\deg^+(b) = 1$, $\deg^+(c) = 2$, $\deg^+(d) = 2$, $\deg^+(e) = 3$, and $\deg^+(f) = 0$.

ลองทำดู

- หาดีกรีของจุดยอดแต่ละจุดของกราฟ G



- หาผลรวมของดีกรีของจุดยอดของทุกจุด

ลองทำดู

- วาดกราฟที่มีผลรวมดีกรีของจุดยอดทุกจุดยอดเป็น **11**
- กราฟนี้ ทำได้หรือไม่ ?
- **ไม่**
- สังเกตว่า เส้นเชื่อม มันมีจุดปลายสองด้าน แน่นอนต้องไปลงกับจุดยอดในกราฟ **=** ดีกรีรวมเพิ่มขึ้นทีละ **2**

Handshake Theorem

ถ้ามี Undirected graph G = (V,E)

- จะได้ว่า $2|E| = \sum \deg(v)$

ตัวอย่าง

- กราฟจะมีเส้นเชื่อมกี่เส้น ถ้าหากมีจุดยอดอยู่ 10 จุด แต่ละจุดมีดีกรี 3
- Sol: ผลรวมดีกรีเป็น 30 ดังนั้นมีเส้นเชื่อม 30/2=15 เส้น

- จะมีกราฟที่มีจุดยอด 5 จุด โดยแต่ละจุดมีดีกรีเป็น 2,3,5,9,10 ตามลำดับ ได้หรือไม่
- Sol: ไม่ได้ เพราผลรวมดีกรีเป็น 29 นั่นคือมีเส้นเชื่อม 29/2=14.5 เส้น ซึ่งไม่มี

- **ทฤษฎีบทตาม :** กราฟไม่มีทิศทางใดๆ จะมีจุดยอดคี่ได้เป็นจำนวนคู่

ทฤษฎีบท (Theorem)

ถ้ามี Directed graph G = (V,E)

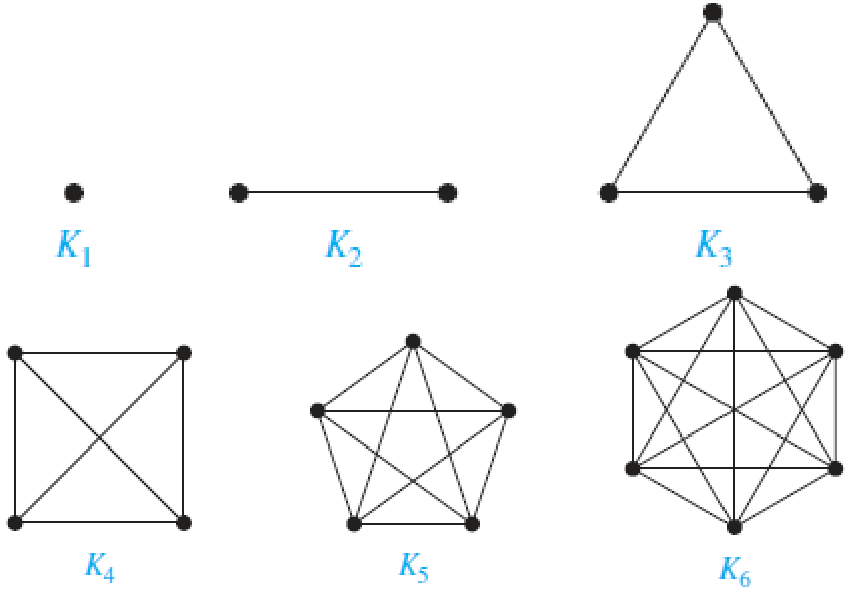
- จะได้ว่า $\sum \deg^{-}(v) = \sum \deg^{+}(v) = |E|$

- Note:
- underlying undirected graph หมายถึง
- undirected graph ที่เกิดจากการไม่สนใจทิศทางของ directed edge

กราฟที่ควรรู้จัก

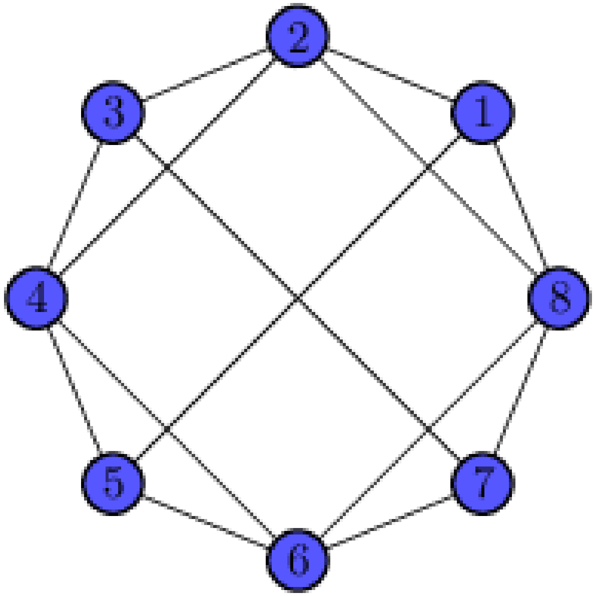
กราฟบริบูรณ์ (Complete Graph: K_n)

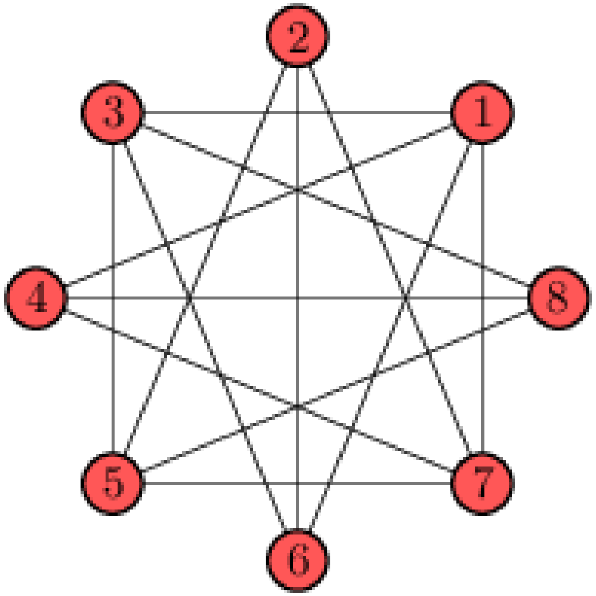
- กราฟอย่างง่าย ที่จุดยอดทุกจุดมีเส้นเชื่อมถึงกัน
- กราฟที่ไม่ใช้กราฟบริบูรณ์ เรียกกราฟไม่บริบูรณ์ (noncomplete graph)



กราฟส่วนเติมเต็ม (Complementary Graph: $\overline{G}$)

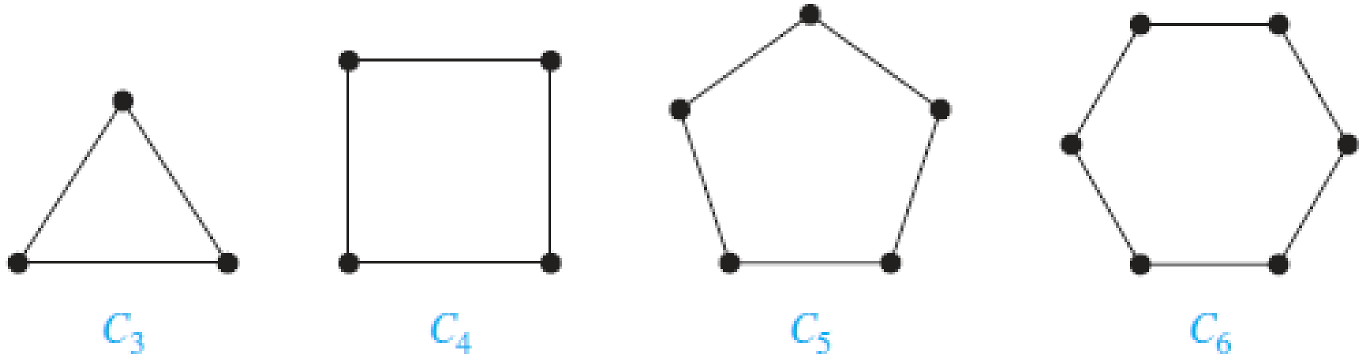
- G′ , $\bar{G}$ คือ กราฟอย่างง่ายที่สร้างจากคู่จุดที่ไม่มีเส้นเชื่อมถึงกันใน G
- $\bar{G} \cup G = K_n$





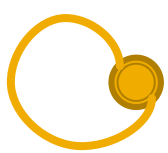
กราฟวัฏจักร (Cycle Graph: C_n)

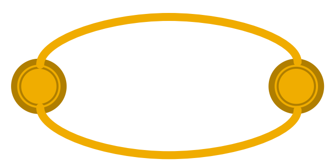
- กราฟที่มีจุดยอดอย่างน้อย 3 สมมติเรียก $v_1, v_2, \dots, v_n$
- จะมีเส้นเชื่อม v_i, v_j เมื่อ |i-j|=1 และมีเส้นเชื่อม เชื่อม v_n, v_1 ด้วย

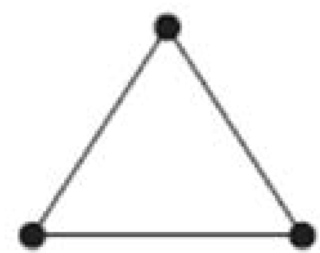


ขยายนิยามกราฟวัฏจักร

- C_1 = วงวน (loop)
- C_2 = digon
- C_3 = สามเหลี่ยม (triangle)

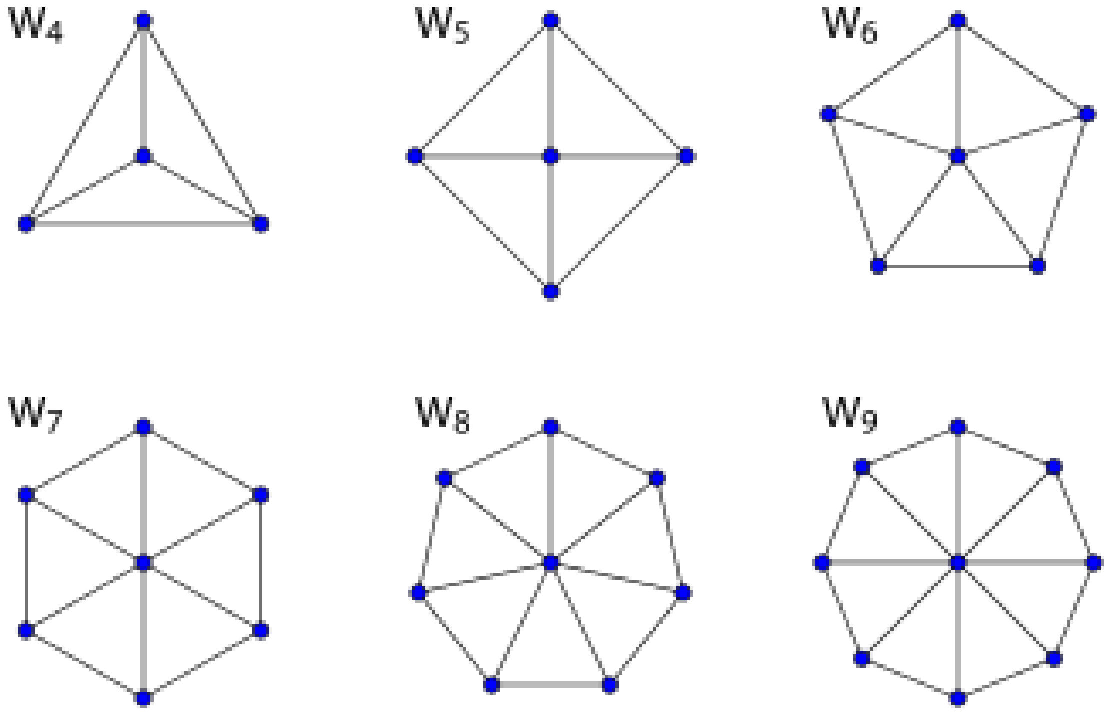






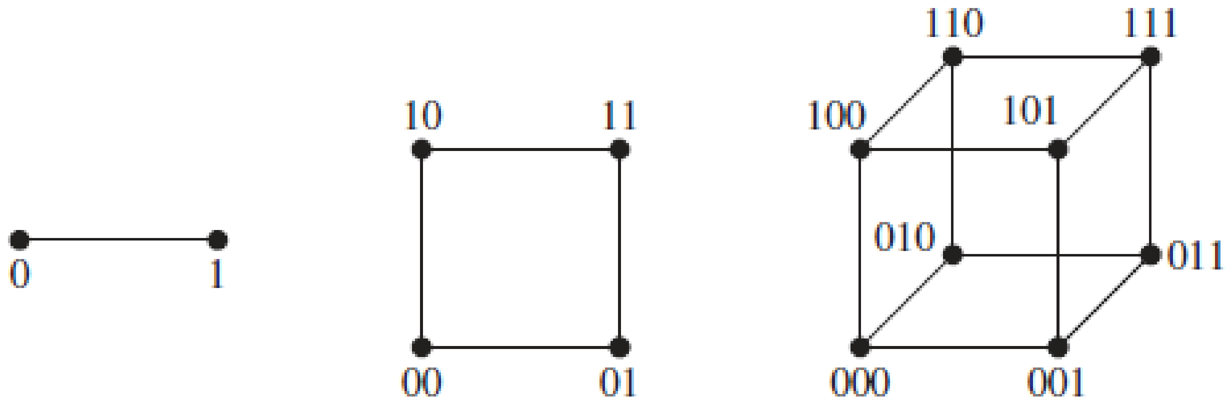
กราฟกงล้อ (Wheel Graph: W_n)

- เหมือนกราฟวัฏจักร
- แต่มีจุดอยู่จุดหนึ่ง เชื่อมไปยังทุกจุดรอบๆ $n-1$ จุด



กราฟลูกบาศก์ (n-Cube: Q_n)

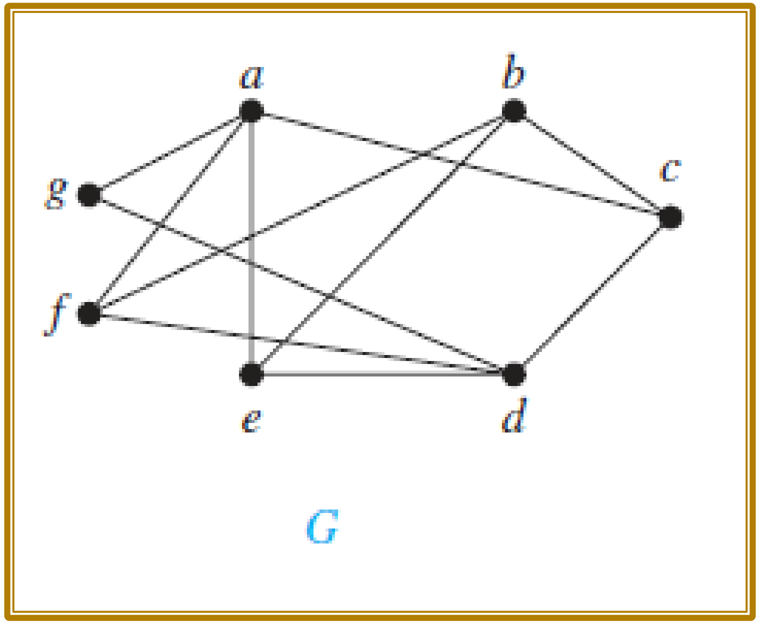
- มีจุดยอด 2^n จุด
- สมมติให้แต่ละจุดมีหมายเลขตั้งแต่ $0 - 2^n - 1$
- จะมีเส้นเชื่อมถึงกัน ถ้าเลขจุดยอดนั้นต่างกันเพียงบิตเดียว



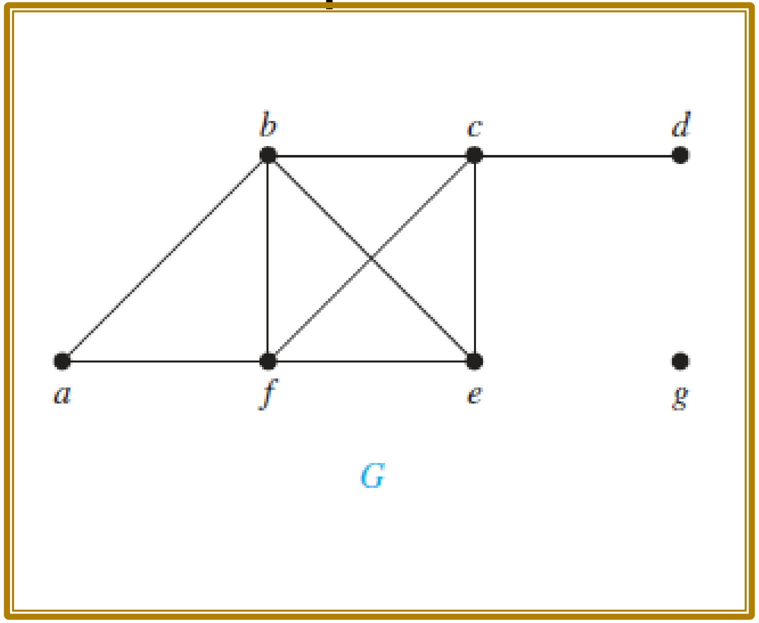
Q_1 Q_2 Q_3

กราฟเชื่อมโยง (Connected Graph)

- กราฟที่จุดทุกคู่ สามารถไปถึงกันได้
- กราฟที่ไม่เชื่อมโยง เรียก Disconnected Graph



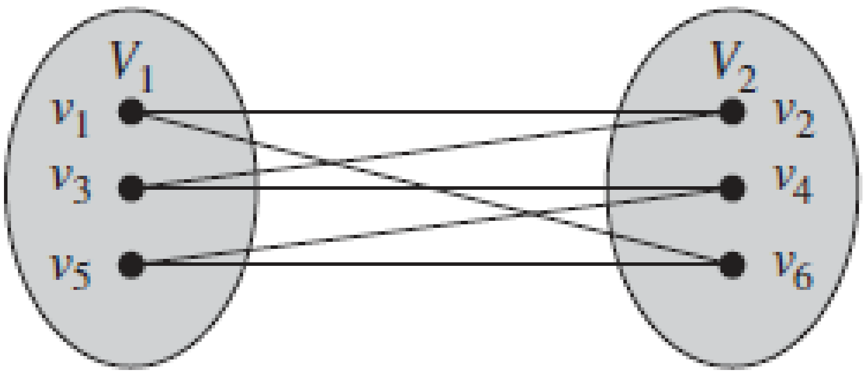
Connected Graph



Disconnected Graph

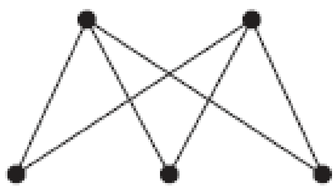
กราฟสองส่วน (Bipartite Graph)

- คือ กราฟอย่างง่าย $G = (V_1 \cup V_2, E)$ ซึ่ง**เป็นกราฟเชื่อมโยง** และไม่มี edge เชื่อมกันภายใน V_1 และ V_2

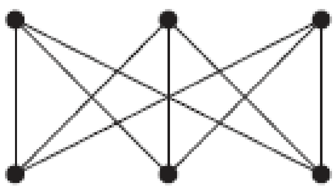


กราฟสองส่วนบริบูรณ์ (Complete Bipartite Graph $K_{n,m}$)

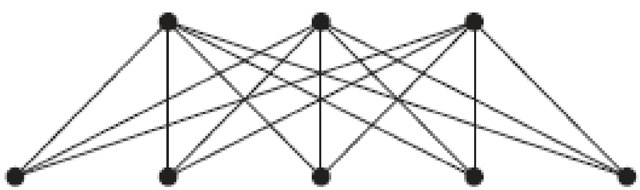
- กราฟสองส่วน ที่มีเส้นเชื่อมตามเงื่อนไข ครบทุกคู่



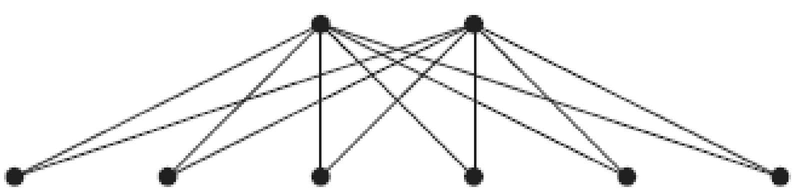
$K_{2,3}$



$K_{3,3}$



$K_{3,5}$



$K_{2,6}$

การให้สีกราฟ (Graph Coloring)

- มีหลายลักษณะ เช่น
- Vertex coloring = จุดยอดที่ติดกัน มีสีไม่เหมือนกัน
- Edge coloring = เส้นเชื่อมที่ติดกัน มีสีไม่เหมือนกัน
- Total coloring = Vertex + Edge coloring

- ! ปกติมักจะพูดถึง Vertex coloring เป็นหลัก

คำศัพท์ (Terminology) V

- กราฟที่สามารถ**ให้สีจุดยอด** k สีได้ เรียก k-coloring
- สีที่ใช้น้อยสุด ในการ**ระบายจุดยอด**ในกราฟ G เรียก Chromatic number ($\chi(G)$ บ้างก็ใช้ $\gamma(G)$)
- จำนวนวิธีการระบายสีจุดยอด โดยใช้ t สี คือ Chromatic polynomial
- กราฟที่สามารถ**ให้สีเส้นเชื่อม** k สีได้ เรียก k-edge-coloring
- สีที่ใช้น้อยสุด ในการ**ระบายเส้นเชื่อม**ในกราฟ G เรียก Chromatic index หรือ Edge chromatic number ($\chi'(G)$)

ทฤษฎีบท

- กราฟอย่างง่าย เป็นกราฟสองส่วน ก็ต่อเมื่อ เป็นกราฟ 2-coloring
- พิสูจน์
- → เป็นกราฟสองส่วน จะมีจุดยอดสองกลุ่ม ก็ให้กลุ่มหนึ่งเป็นสี 1 อีกกลุ่มเป็นสี 2 ก็จะได้ว่าสามารถให้สีสองสีตามต้องการ
- ← เมื่อเป็นกราฟ 2-coloring ก็ให้จับกลุ่มสีเดียวกันเข้าด้วยกัน จะได้ว่าในกลุ่มสีเดียวกันนี้ไม่มีเส้นเชื่อมถึงกัน ดังนั้นก็ได้กราฟสองส่วนตามต้องการ

คำถาม

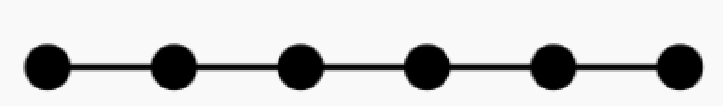
- กราฟซึ่งมีจุดยอดหนึ่งจุด เป็นกราฟสองส่วนหรือไม่
- คำตอบ คือ **เป็น**
- K_n, C_n, W_n, Q_n สำหรับแต่ละข้อ n เป็นเท่าใด จึงเป็นกราฟสองส่วน
  - K_n เมื่อ n <= 2
  - C_n เมื่อ n เป็นเลขคู่
  - W_n ไม่มีค่า n ที่ทำได้
  - Q_n ได้ทุกค่า

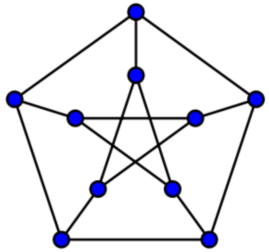
Chromatic number

- หา Chromatic number ของแต่ละกราฟ
  - K_n n
  - C_n If odd 3 / even 2
  - W_n If odd 3 / even 4
  - Q_n 2

Sample Chromatic polynomial

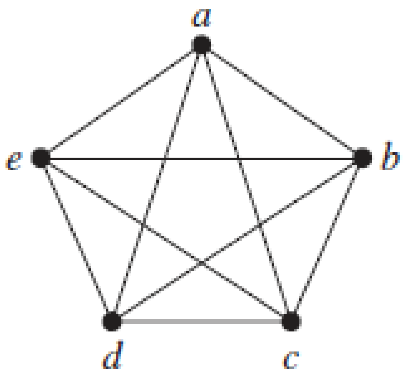
- มีสีระบาย t สี และมี n จุดยอด
- $K_n = t(t-1)(t-2)\ldots(t-(n-1))$
- $C_n = (t-1)^n + (-1)^n(t-1)$
- $W_n = t\left[(t-2)^{n-1} - (-1)^{n(t-2)}\right]$
- $Path\ graph\ P_n = t(t-1)^{n-1}$
- $Petersen\ graph = t(t-1)(t-2)(t^7 - 12t^6 + 67t^5 - 230t^4 + 529t^3 - 814t^2 + 775t - 352)$

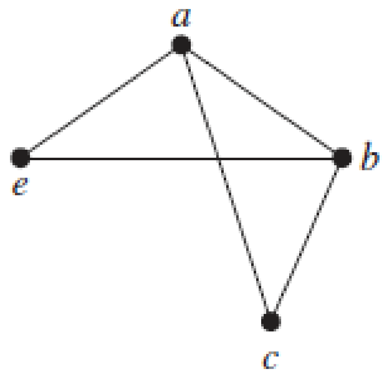

A path graph on 6 vertices



คำศัพท์ (Terminology) VI

- กราฟย่อย (subgraph) ของ G = (V,E) คือ กราฟ H=(V′,E′) โดย $V' \subseteq V$ และ $E' \subseteq E$
- กราฟย่อยแท้ (proper subgraph) ของ G คือ กราฟย่อยของ G ที่ไม่ใช่ G (นึกถึง เซตย่อยแท้ / ไม่แท้)
- กราฟย่อยที่เป็นกราฟบริบูรณ์ เรียกว่า คลีก (clique)
- จะเรียก H=(V′,E′) เป็น subgraph induced ของ G=(V,E) ถ้าใน E′ มีทุก edge ที่อยู่ใน E ที่มีจุดเริ่ม-จบ ใน V′





- ด้านขวาเป็น กราฟย่อย(แท้) ของทางซ้าย

- ถ้ามีเส้นเชื่อม ec จะได้ว่าทางขวาเป็น subgraph induced

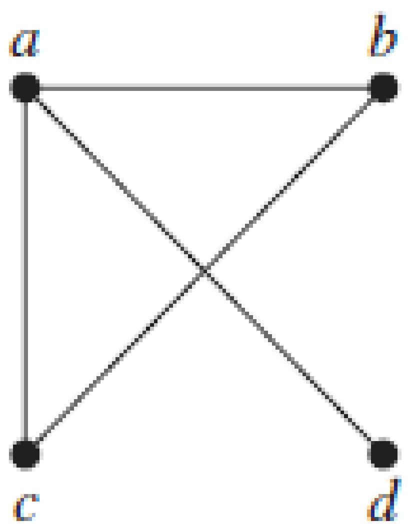
Represent graph in computer

เก็บกราฟลงในคอมพิวเตอร์อย่างไรดี

- มีหลากหลายวิธีในการเก็บกราฟลงบนคอมพิวเตอร์
- เลือกใช้ให้เหมาะสมกับวิธีที่จะใช้งาน
- ที่นิยมใช้มี 3 แบบ และแถมอีก 1 วิธี มาให้ดู

รายการเส้นเชื่อม (Edge list)

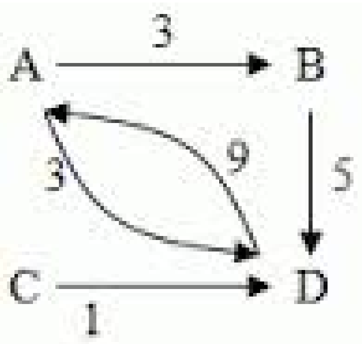
- เก็บแค่ว่า มีเส้นเชื่อมอะไรในกราฟบ้าง
- อาจใช้อาเรย์ หรือ vector<pair<?,?> > มาเก็บก็ได้



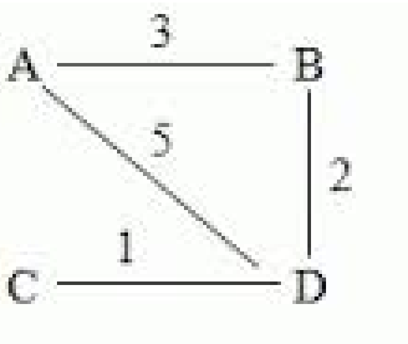
ab
ad
ac
bc

คำศัพท์ (Terminology) VIII

- กราฟไม่มีน้ำหนัก (unweighted graph) คือ กราฟที่เส้นเชื่อมทุกเส้นไม่มีน้ำหนัก
- กราฟมีน้ำหนัก (weighted graph) คือ กราฟที่เส้นเชื่อมมีน้ำหนัก
- ในทางปฏิบัติ กราฟที่มีน้ำหนักเท่ากันทุกเส้น เราก็มองเป็นกราฟไม่มีน้ำหนัก



Directed-weighted

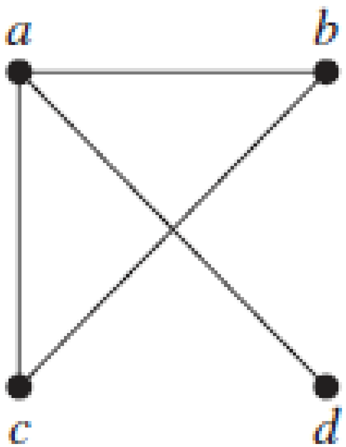


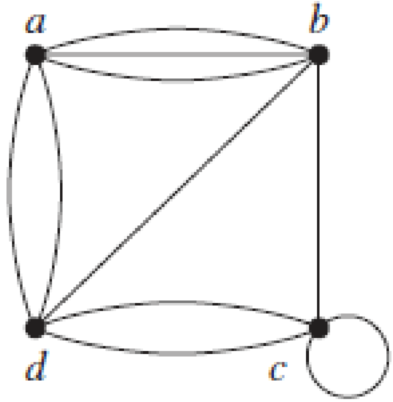
Undirected-weighted

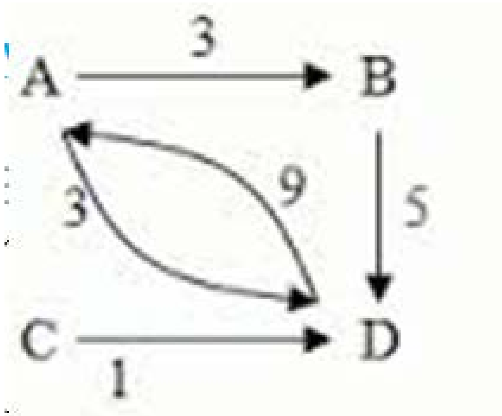
เมทริกซ์ประชิด (Adjacency Matrix)

- มีอาเรย์ขนาด |V| x |V|
- ถ้ามีเส้นเชื่อม จาก u ไป v ให้ใส่ค่าที่ช่อง [u][v] เป็น 1
- อย่างไรก็ดี การใส่ค่าในอาเรย์ ก็ขึ้นอยู่ว่าเราทำอะไร บนอะไร
- เช่น simple unweight ก็ใส่เป็น 0 กับ 1 (0 คือไม่มี edge ถึงกัน / 1 คือมี edge ถึงกัน)
- unweight ใส่จำนวน edge ที่เชื่อมระหว่าง u v
- weight เก็บน้ำหนัก

ตัวอย่างเมทริกซ์ประชิด (Adjacency Matrix)







$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}.$$

$$\begin{bmatrix} 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \\ 9 & 0 & 0 & 0 \end{bmatrix}$$

การกำหนดค่าเส้นเชื่อม (u,v) ในเมทริกซ์อาจกำหนดเป็นอย่างอื่นก็ได้ ขึ้นกับการนิยามของเรา

รายการประชิด (Adjacency List)

- มีรายการโยง V ตัว มาเก็บ
- สำหรับ รายการโยง **U** ถ้า **U** ติดกับใครก็ใส่เข้าไปในรายการโยง **U**
- อาจมีข้อมูลอื่นติดไปด้วย เช่น น้ำหนัก หรือหมายเลขของเส้นเชื่อม เพื่อไปทำงานอย่างอื่นต่อ

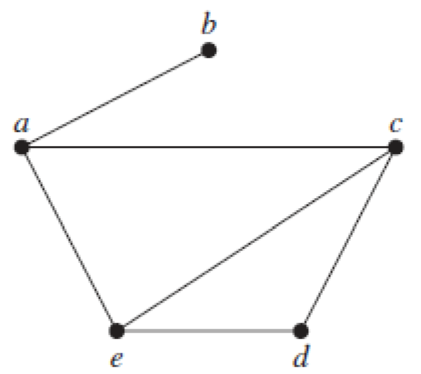


FIGURE 1 A Simple Graph.

TABLE 1 An Adjacency List for a Simple Graph.

Vertex	*Adjacent Vertices*
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

ตัวอย่างรายการประชิด (Adjacency List)

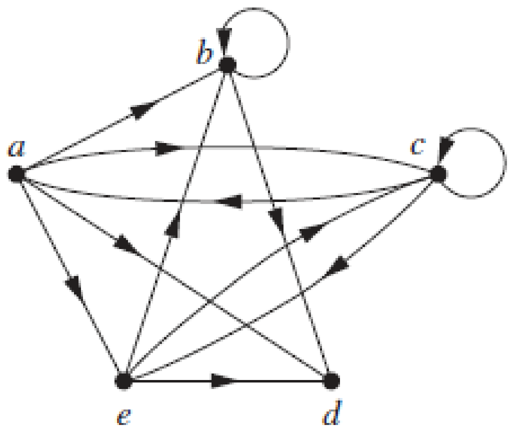
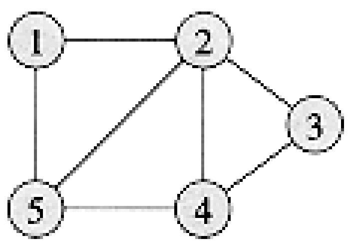


FIGURE 2 A Directed Graph.

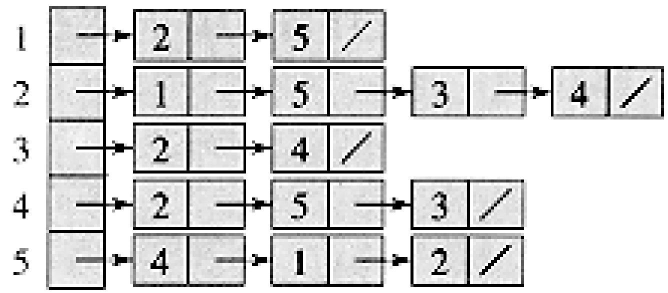
TABLE 2 An Adjacency List for a Directed Graph.

Initial Vertex	*Terminal Vertices*
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

ตัวอย่างการเก็บกราฟอีกนิด



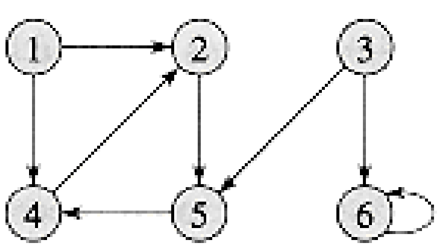
(a)



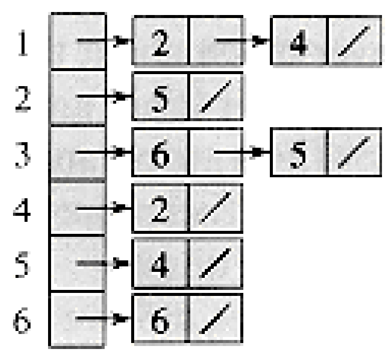
(b)

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

(c)



(a)



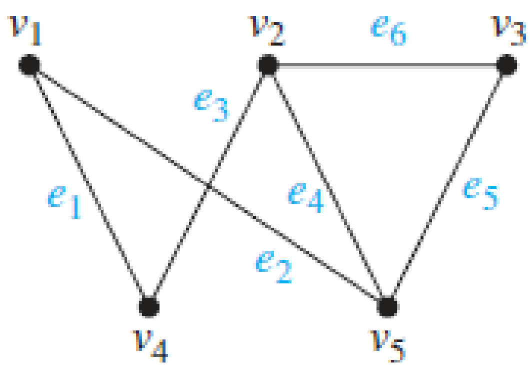
(b)

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

(c)

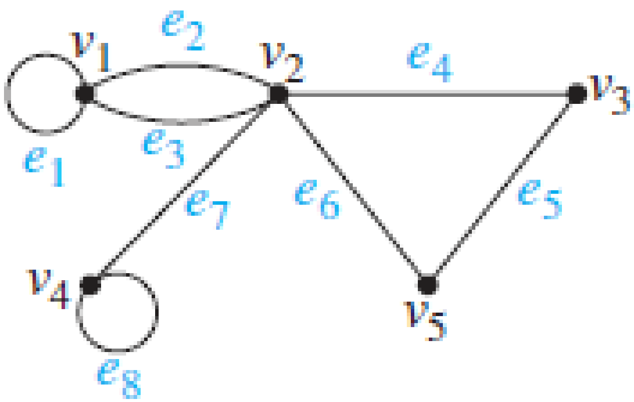
เมทริกซ์เกิดกับ (Incident Matrix)

- มีเมทริกซ์ขนาด |V| x |E|
- สำหรับแต่ละเส้นระบุว่า เส้นนี้เกิดกับใคร
- ใช้กับ undirected graph เท่านั้น



$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{array}{c}
\begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}.
$$

ตัวอย่างเมทริกซ์เกิดกับ (Incident Matrix)



$$
\begin{array}{c|cccccccc}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
\hline
v_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
v_2 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
v_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
v_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
v_5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}.
$$

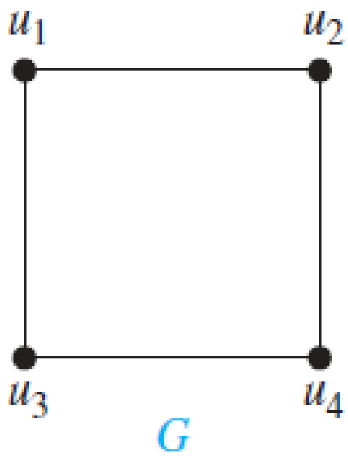
พักทำโจทย์

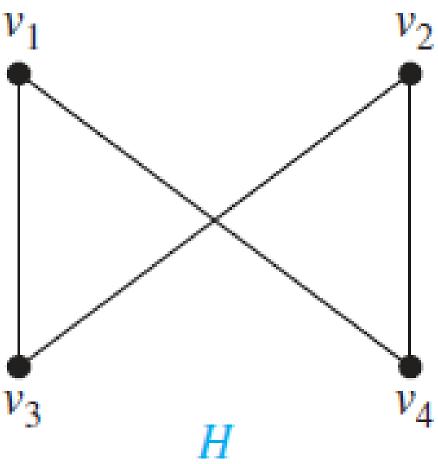
- UVa – 11550 Demanding Dilemma
  - มี Incident Matrix มาให้
  - ถามว่า เป็น simple undirected graph หรือไม่
- Uva – 10895 Matrix Transpose
  - ได้เมทริกซ์มาอันหนึ่ง โดยเมทริกซ์นี้ตัด 0 ออกหมดแล้ว
  - Input จะระบุ n m ตามด้วย 2n บรรทัด
  - สำหรับแต่ละบล็อก จะระบุ r คือ จำนวนที่ไม่เป็น 0 ในแถวนั้น ตามด้วยเลข r ตัว ระบุตำแหน่งที่ไม่เป็น 0 และอีก r ตัว คือ เลขในตำแหน่งตามลำดับ
  - ให้ Output matrix transpose ตามรูปแบบ Input

คำศัพท์ (Terminology) IX

- กราฟอย่างง่าย $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ จะ สมสัณฐาน (isomorphic) $[G_1 \simeq G_2]$ กัน ถ้ามีฟังก์ชันหนึ่งต่อหนึ่งทั่วถึง $\mathrm{f}: V_1 \rightarrow V_2$ โดย a กับ b ติดกัน ก็ต่อเมื่อ f(a) กับ f(b) ติดกัน
- เรียก ฟังก์ชัน f นี้ว่า isomorphism
- พูดง่ายๆ คือ สามารถวาดกราฟใหม่ให้เหมือนกันทุกประการได้
- ถ้า G กับ G′ isomorphic กัน เรียก self-complementary
- กราฟอย่างง่ายที่ไม่สมสัณฐาน เรียก กราฟไม่สมสัณฐาน (nonisomorphic)
- สามารถขยายผลนิยามบน nonsimple graph รวมถึง directed graph ได้

ตัวอย่าง

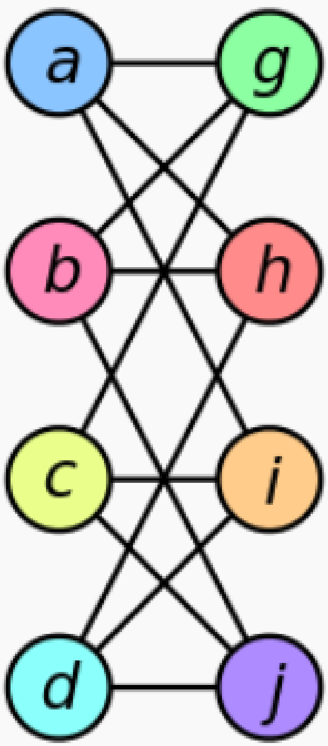
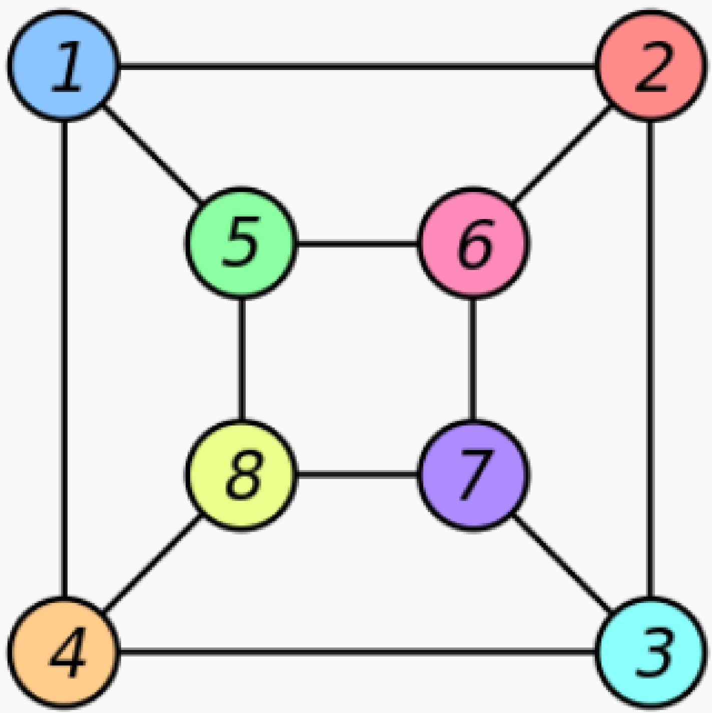




G กับ H สมสัณฐานกัน

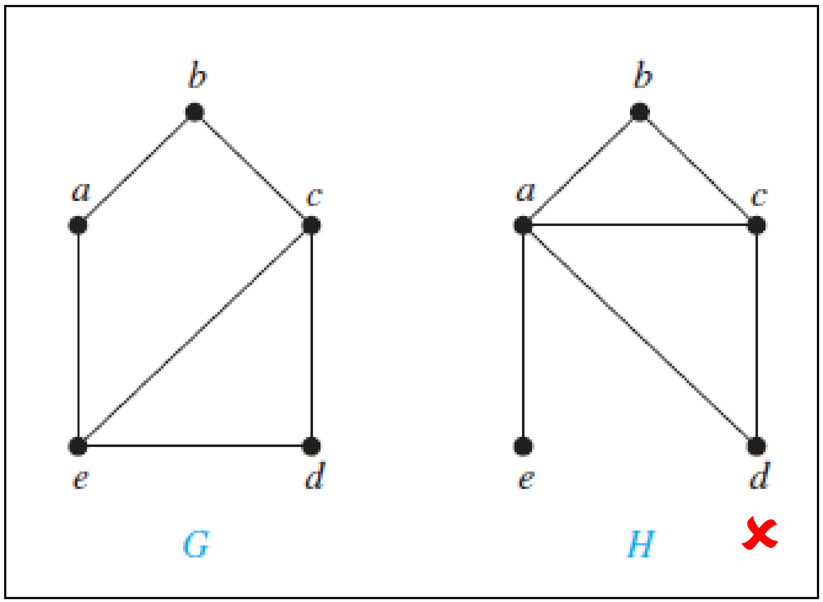
ลองบิดกราฟด้านล่างดู มันก็จะได้กราฟบน

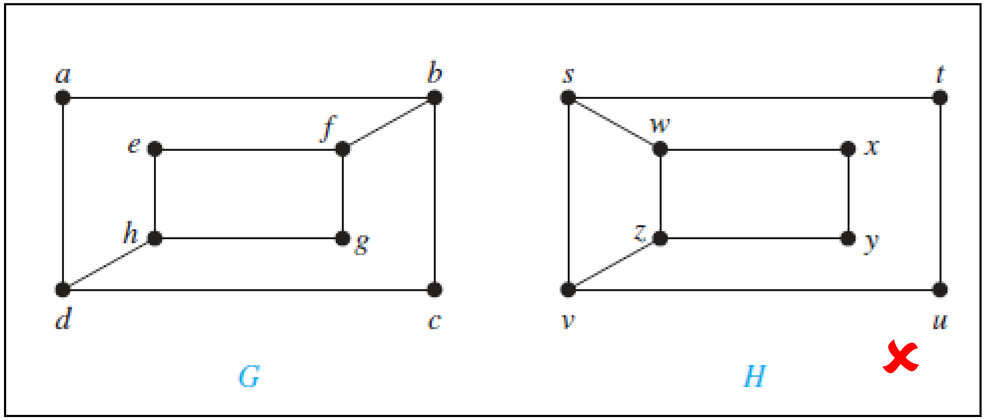
ตัวอย่าง (ต่อ)

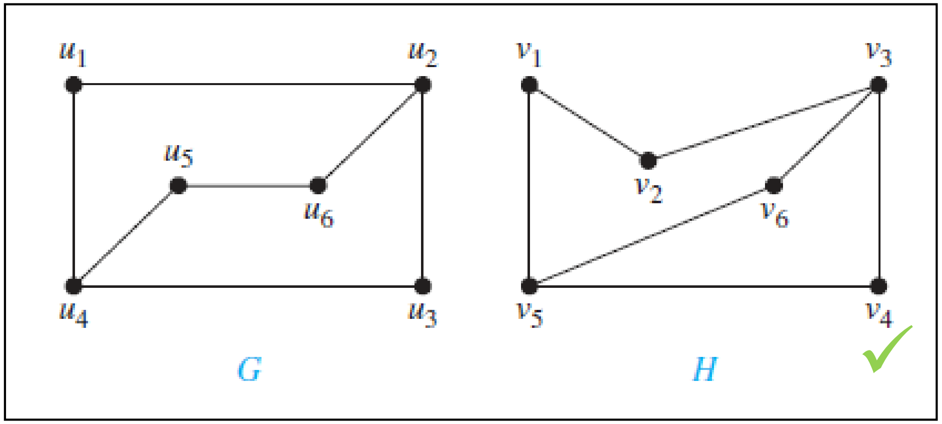
Graph G	Graph H	An isomorphism between G and H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

ตัวอย่าง (ต่อ)

- แต่ละคู่ สมสัณฐานกันหรือไม่







วิธีหาว่า กราฟสมสัณฐานกันหรือไม่

- ทำไงดี ?
- เขียนกราฟเป็น adjacency matrix แล้วไปหาต่อว่า match จุดยอดอย่างไร
- Time complexity คือเท่าไหร่ ?
- O(N!) ด้วยวิธีนี้เต็มที่ก็ optimize ได้แค่เป็น exponential
- มีวิธีที่ดีกว่า (เร็วกว่า) นี้มั้ย
- **ก็ไม่รู้สินะ**
- ถึงตอนนี้ยังไม่รู้ว่า มีวิธีเป็น polynomial time หรือไม่ (NP)

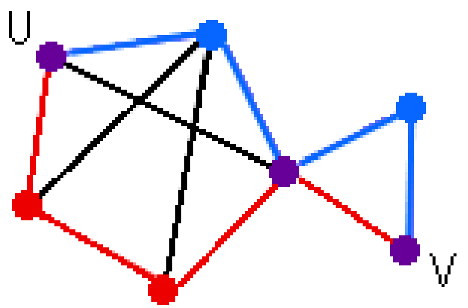
คำศัพท์ (Terminology) X

- แนวเดิน (walk) คือ ลำดับการเดิน โดยระบุเป็นจุดยอดสลับกับเส้นเชื่อม $v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n$ โดย v_{i-1} กับ v_i คือ จุดปลายของเส้นเชื่อม e_i
- แนวเดินปิด (closed walk) คือ แนวเดินที่ $v_0 = v_n$
- ถ้าไม่ใช่แนวเดินปิด ก็คือแนวเดินเปิด (open walk)
- ความยาว (length) ของแนวเดิน คือ จำนวนเส้นเชื่อมในแนวเดิน
- รอยเดิน (trail) คือ แนวเดินที่**ไม่มีเส้นเชื่อมซ้ำ**
- รอยเดินปิด (closed trail) เรียกชื่ออื่นว่า tour, circuit

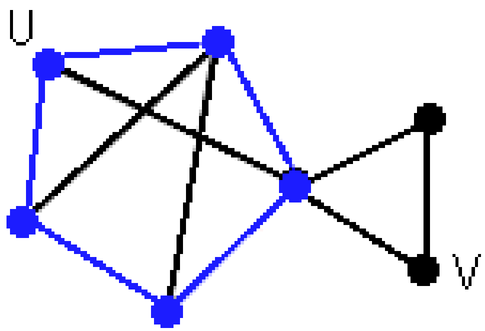
คำศัพท์ (Terminology) XI

- วิถี (path) คือ แนวเดินที่**ไม่มีจุดยอดซ้ำ** ยกเว้นจุดเริ่มกับจบที่อาจซ้ำกันได้
- วัฏจักร (cycle) คือ วิถีที่มีจุดเริ่มกับจบเป็นจุดเดียวกัน
- เนื่องจากวิถีไม่มีการใช้ edge ซ้ำ จึงเขียนแต่จุดยอดเพื่อบอกวิถีที่เดินได้
- แต่สำหรับ กราฟที่ไม่ใช่กราฟอย่างง่าย ต้องระบุเส้นเชื่อมให้ชัดเจน อาจเกิดความกำกวมได้หากระบุแบบนี้
- บางตำรา เรียก circuit = cycle และเรียกวัฏจักรที่ไม่มีเส้นเชื่อมซ้ำว่า simple circuit
- circuit และ cycle ต้องมีความยาวอย่างน้อย 3

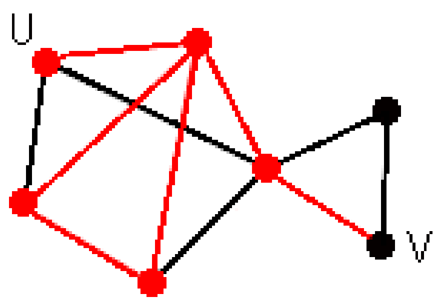
ตัวอย่าง



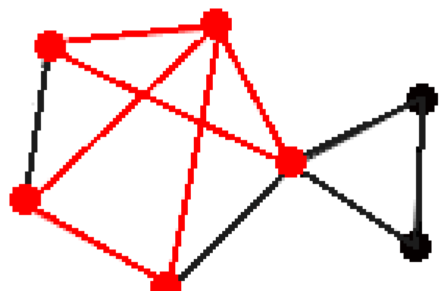
2 paths u-to v



cycle



trail u-to v



circuit

ทฤษฎีบท

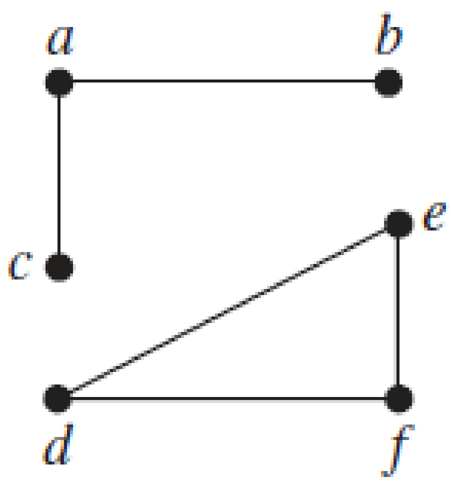
- สำหรับจุดยอด **U, V** ทุกคู่ในกราฟเชื่อมโยง จะมีวิถีถึงกัน

พิสูจน์

- เพราะว่าเป็นกราฟเชื่อมโยง จึงได้ว่ามีเส้นทางถึงกัน
- แต่แน่ใจได้อย่างไรว่าเป็น วิถี
- สมมติว่าเป็นแนวเดิน (มีการใช้จุดยอดซ้ำ)
- เราจะได้ว่าแทนที่จะวนกลับมาให้เกิดจุดยอดซ้ำ เราก็ข้ามไปเลยก็ได้
- นั่นคือ มีวิถีทุกคู่ **U, V**

คำศัพท์ (Terminology) XII

- องค์ประกอบ (component) ของกราฟ คือ กราฟย่อยที่ใหญ่ที่สุดที่เชื่อมโยงกัน
- กราฟที่มีแค่ 1 component เรียกว่า connected graph

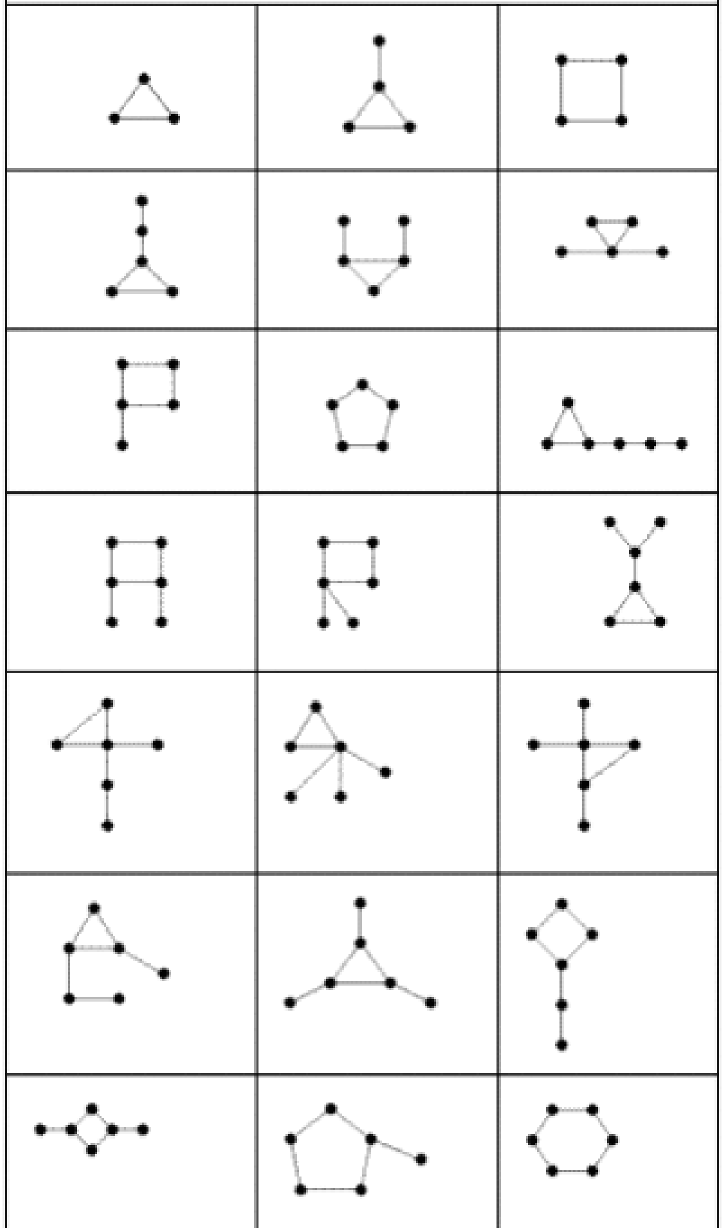


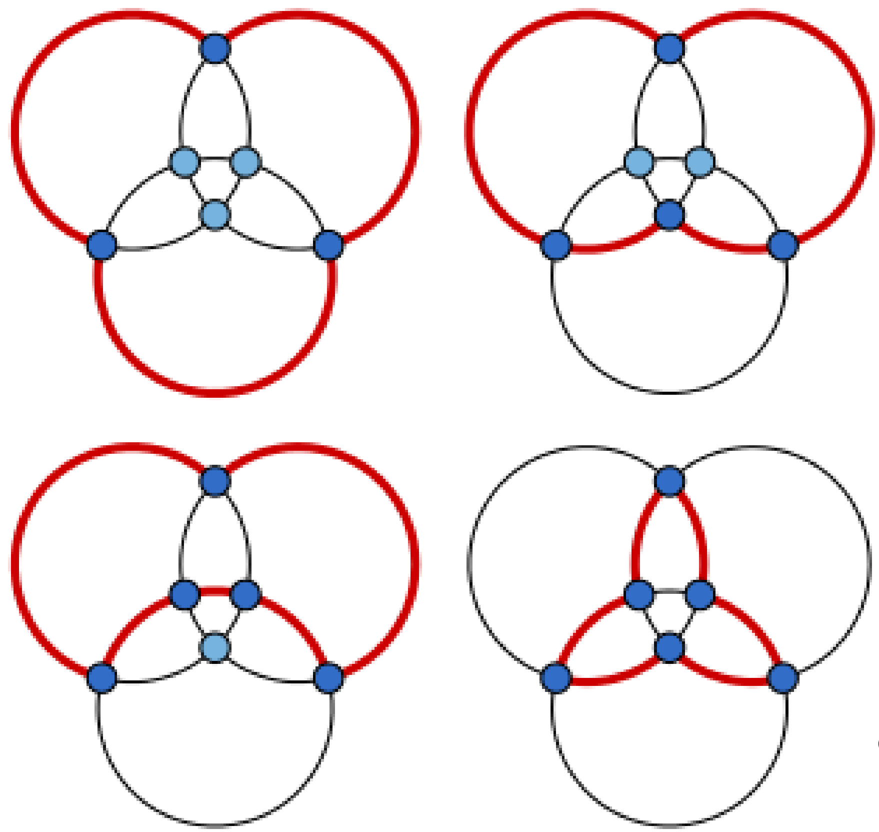
กราฟนี้มี 2 องค์ประกอบ

คำศัพท์ (Terminology) XIII

- กราฟที่ไม่มี cycle เรียก acyclic graph
- กราฟมีทิศทางที่ไม่มี cycle เรียก directed acyclic graph (DAG) ซึ่งสำคัญมากในเรื่องกำหนดการเชิงพลวัต (dynamic programming)
- กราฟที่มี cycle แค่หนึ่ง cycle เรียก unicyclic graph
- กราฟที่มี cycle ตั้งแต่ขนาด 3 ถึง อันดับของกราฟ เรียก pancyclic graph

The twenty one unicycles with 3, 4, 5, and 6 vertices





ตัวอย่าง

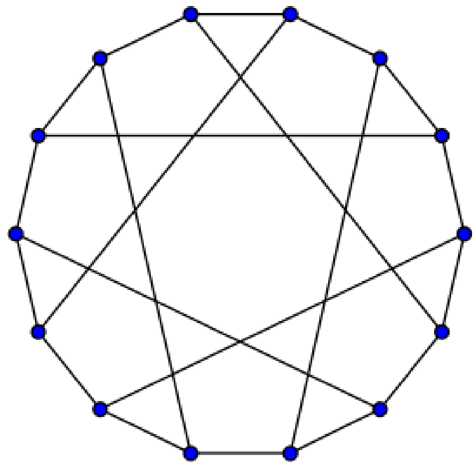
Pancyclic graph

คำศัพท์ (Terminology) XIV

- girth ของกราฟ G (g(G)) คือ ขนาดของ cycle ที่เล็กที่สุดในกราฟ
- ถ้ากราฟไม่มี cycle จะได้ว่า girth ของกราฟ คือ อนันต์
- กราฟที่มี girth > 3 เรียก กราฟปลอดสามเหลี่ยม (Triangle-free graph)
- circumference ของกราฟ G คือ ขนาดของ cycle ที่ใหญ่ที่สุดในกราฟ

ตัวอย่าง

Graph	Girth	Circumference
K_n	3 เมื่อ n>2	n เมื่อ n>2
$K_{n,m}$	3 เมื่อ min(n,m) > 1	2 min(n,m) เมื่อ min(n,m) > 1
C_n	n	n
W_n	3	n-1
Q_n	4 เมื่อ n>1	2^n เมื่อ n>1

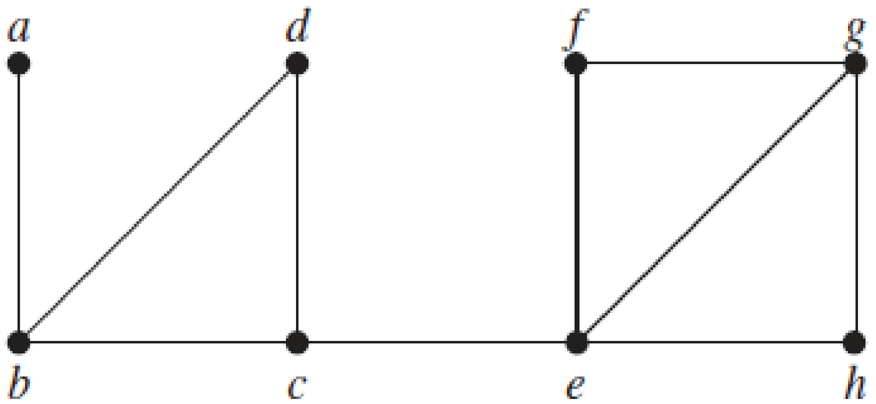


Girth = 6
Circumference = 14

คำศัพท์ (Terminology) XV

- cut vertex / articulation point คือ จุดยอด ที่เมื่อเอาออกแล้วทำให้กราฟไม่เชื่อมโยงกัน
- cut edge / bridge คือ เส้นเชื่อม ที่เมื่อเอาออกแล้วทำให้กราฟไม่เชื่อมโยงกัน
- กราฟที่ไม่มี cut vertex เรียกว่า nonseparable graph
- มี $G = (V, E)$ ให้ $V' \subseteq V$ จะเรียก V' ว่า vertex cut / vertex separator / separating set ถ้า G – V' เป็นกราฟไม่เชื่อมโยง
- vertex connectivity ของกราฟ G ($\kappa(G)$) คือ ขนาดของ vertex cut ที่เล็กที่สุดของ G

ตัวอย่าง



G_1

b, c, e เป็น articulation point

(เรียกแบบนี้ดีกว่า จะไม่ได้งงกับ vertex cut)

ab, ce เป็น bridge

$\kappa(G) = 1$

ตัวอย่าง vertex cut

- พิจารณา K_n จะเห็นว่าไม่มี vertex cut เพราะตัดอย่างไรกราฟก็ไม่ขาด
- นิยาม $\kappa(G)$ เพิ่มเติมคือ เป็นขนาดของ vertex cut ที่เล็กที่สุด หรือตัดจนกราฟเหลือจุดยอดหนึ่งจุด
- จะเห็นได้ว่า
- ถ้า G มันไม่เชื่อมโยงอยู่แล้ว หรือ มีจุดยอดจุดเดียว จะมี $\kappa(G) = 0$
- กราฟที่มี articulation point หรือเป็น K_2 จะมี $\kappa(G) = 1$
- กราฟที่ไม่มี articulation point แต่สามารถขาดจากกันได้โดยตัดจุดยอดสองจุด หรือเป็น K_3 จะมี $\kappa(G) = 2$
- กราฟ K_n จะมี $\kappa(G) = n - 1$

คำศัพท์ (Terminology) XVI

- กราฟนั้นเป็นกราฟ k-connected (k-vertex-connected) ถ้า $\kappa(G) \geq k$
- 1-connected ก็คือ connected graph
- 2-connected (biconnected) ก็คือ กราฟที่ไม่มี articulation point (nonseparable graph)

คำศัพท์ (Terminology) XVII

- มี $G = (V, E)$ ให้ $E' \subseteq E$ จะเรียก E′ ว่า edge cut ถ้าหาก G – E′ เป็นกราฟไม่เชื่อมโยง
- edge connectivity ของกราฟ G ($\lambda(G)$) คือ ขนาดของ edge cut ที่เล็กที่สุดของ G
- $\lambda(G)$ = 0 ถ้าหาก G ไม่ connected หรือ G มีจุดยอดเดียว

คำถาม

ถ้า G คือ k-connected graph แล้ว G เป็น j **-connected graph** สำหรับทุก j ที่ $0 \leq j \leq k$

- ถูก

ถ้า G เป็นกราฟอย่างง่ายที่มีจุดยอด n จุด แล้ว $0 \leq \lambda(G) \leq n - 1$

- ถูก

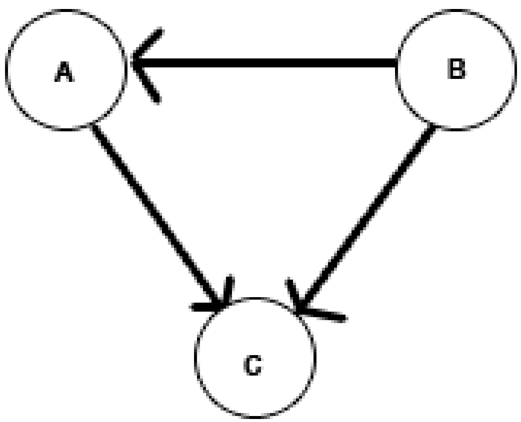
ให้ G เป็นกราฟอย่างง่ายที่มีจุดยอด n จุด จะมี $\lambda(G) = n - 1$ ก็ต่อเมื่อ $G = K_n$

- ถูก

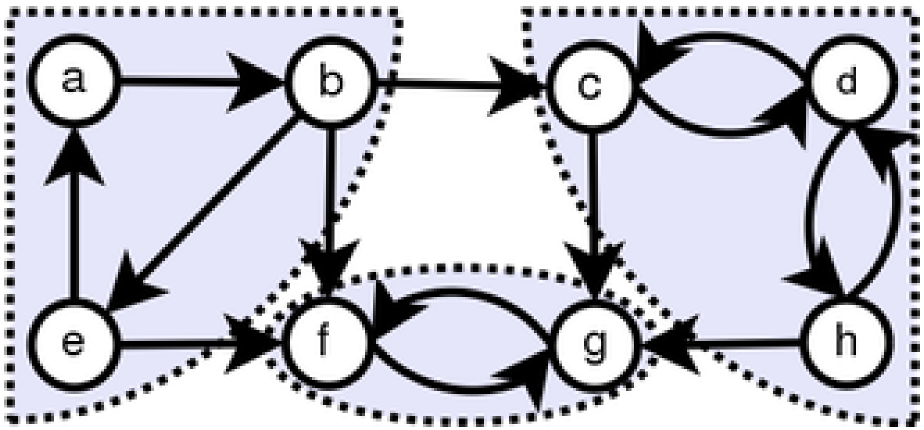
คำศัพท์ (Terminology) XVIII

- Directed graph มีการ **เชื่อมโยงแบบเข้ม** (strongly connected) ถ้าทุกคู่จุด (u,v) มีวิถีถึงกัน
- Directed graph มีการ **เชื่อมโยงแบบอ่อน** (weakly connected) ถ้าแปลงเป็น underlying undirected graph แล้วทุกคู่จุด (u,v) มีวิถีกัน
- strongly connected component คือ subgraph ที่ใหญ่ที่สุดที่มีสมบัติ strongly connected

ตัวอย่าง



weakly connected graph



กราฟนี้มี 3 SCC
(strongly connected component)

ทฤษฎีบท

- จำนวนวิธีในการเดินทางจาก u ไป v ที่มีความยาวแนวเดินเป็น n บนกราฟ G คือ สมาชิกของ A^n ตัวที่อยู่ในแถวที่ u หลักที่ v เมื่อ A คือ adjacency matrix ของ G

พิสูจน์

- กำหนดให้ weight ทุก edge เป็น 1
- สมมติว่า A^r คือ จำนวนแนวเดินความยาว r จาก v_i ไป v_j
- $A^{r+1} = A^r A$ จะได้ว่า A^{r+1} ที่ตำแหน่ง (i,j) คือ $b_{i1}a_{1j} + b_{i2}a_{2j} +\ldots+b_{in}a_{nj}$
- โดย การอุปนัยเชิงคณิตศาสตร์ จะได้ว่า ทฤษฎีบทเป็นจริง

ตัวอย่างการหาจำนวนแนวเดิน

```
The graph:    0->1 with length 1: 0->1 (only 1 path)
              0->1 with length 2: impossible
 0--1         0->1 with length 3: 0->1->2->1 (and 0->1->0->1)
    |         0->1 with length 4: impossible
    2--3      0->1 with length 5: 0->1->2->3->2->1 (and 4 others)
```

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} M^2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} M^3 = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{bmatrix} M^5 = \begin{bmatrix} 0 & 5 & 0 & 3 \\ 5 & 0 & 8 & 0 \\ 0 & 8 & 0 & 5 \\ 3 & 0 & 5 & 0 \end{bmatrix}$$

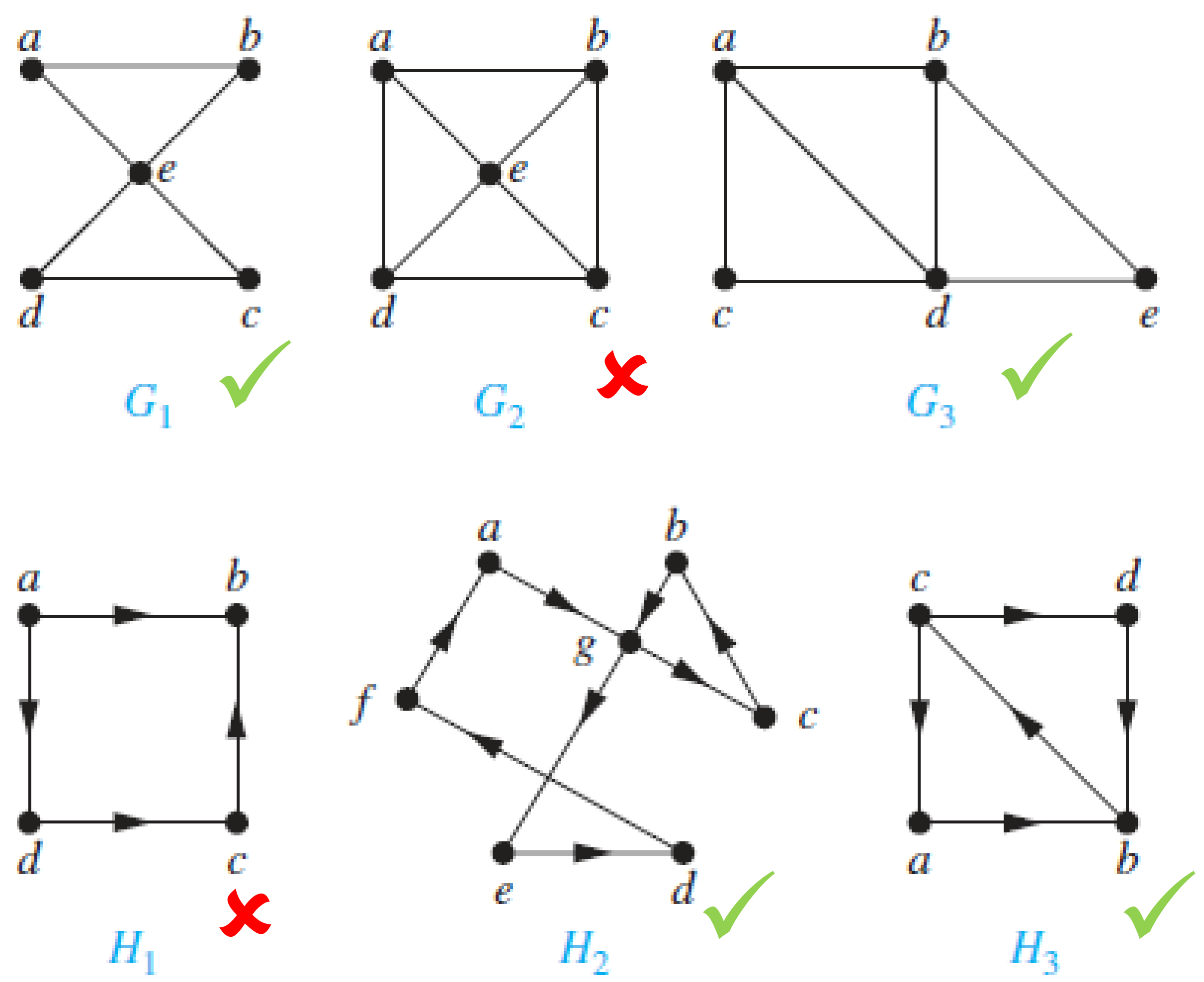
คำศัพท์ (Terminology) XIX

- วัฏจักรออยเลอร์ (Euler circuit) คือ **วัฏจักร**ที่**ผ่านทุกเส้นเชื่อม**ในกราฟ เส้นละครั้ง
- วิถีออยเลอร์ (Euler path) คือ **วิถี**ที่**ผ่านทุกเส้นเชื่อม**ในกราฟ เส้นละครั้ง
- วัฏจักรแฮมิลตัน (Hamilton circuit) คือ **วัฏจักร**ที่**ผ่านทุกจุดยอด**ในกราฟ จุดละครั้ง
- วิถีแฮมิลตัน (Hamilton path) คือ **วิถี**ที่**ผ่านทุกจุดยอด**ในกราฟ จุดละครั้ง

ทฤษฎีบท

- กราฟไม่มีทิศทาง ที่เชื่อมโยง และเป็นจุดยอดคู่ทุกจุด จะมี Euler circuit
- กราฟไม่มีทิศทาง ที่เชื่อมโยง และมีจุดยอดคี่สองจุด จะมี Euler path
- กราฟมีทิศทาง ที่เชื่อมโยง และทุกจุดมี in-deg = out-deg จะมี Euler circuit
- กราฟมีทิศทาง ที่เชื่อมโยง และมีจุดยอดจุดหนึ่งมี in-deg มากกว่า out-deg อยู่ 1 ,มีจุดยอดอีกหนึ่งจุดที่ out-deg มากกว่า in-deg อยู่ 1 และจุดยอดที่เหลือ มี in-deg = out-deg จะมี Euler path

ตัวอย่าง วัฏจักรออยเลอร์ / วิถีออยเลอร์



หยุดตรวจ

- กราฟออยเลอร์ (Euler graph) คือกราฟที่จุดยอดทุกจุด เป็นจุดยอดคู่
- กราฟออยเลอร์ มีวัฏจักรออยเลอร์ เสมอ ใช่หรือไม่ ?
- ไม่ใช่ เพราะ กราฟออยเลอร์ไม่จำเป็นต้อง เป็นกราฟเชื่อมโยง
- กราฟที่มีวัฏจักรออยเลอร์ เรียก กราฟออยเลอร์เลี่ยน (Eulerian graph)

ทฤษฎีบท เกี่ยวกับ แฮมิลตัน

- ยังไม่มี ทฤษฎีที่ใช้หา Hamilton circuit/path ได้กับกราฟใดๆ
- ตัวอย่าง ทฤษฎีที่ใช้บอกว่ามี วัฏจักรแฮมิลตัน
- Direc's Theorem
- ถ้า G เป็นกราฟอย่างง่าย ที่มีจุดยอด n จุด และ $n \geq 3$ และทุกจุดมีดีกรีอย่างน้อย n/2 จะมี วัฏจักรแฮมิลตัน
- Ore's Theorem
- ถ้า G เป็นกราฟอย่างง่าย ที่มีจุดยอด n จุด และ $n \geq 3$ และ $deg(u)+deg(v) \geq n$ สำหรับทุกคู่ u ,v ที่ไม่ติดกัน จะมี วัฏจักรแฮมิลตัน
- จะพิสูจน์ในหน้าถัดไป

พิสูจน์

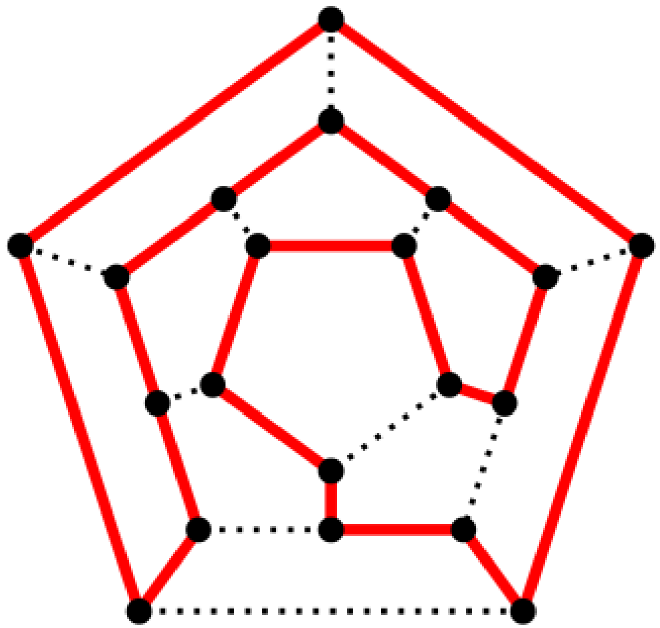
- จะพิสูจน์โดย พิสูจน์แย้ง
- กราฟ G คือกราฟที่สอดคล้องกับ $deg(u)+deg(v) \geq n$ สำหรับทุก u,v ที่ไม่ติดกัน และ G ไม่มีวัฏจักรแฮมิลตัน
- ให้ H คือกราฟที่เติมเส้นเชื่อมบน G ให้มากที่สุดที่เป็นไปได้โดยยังไม่ให้เกิดวัฏจักรแฮมิลตัน
- นั่นคือ ถ้าใส่เส้นเชื่อมไปอีก จะทำให้เกิดวัฏจักร ฉะนั้นกราฟ H มีวิถีแฮมิลตัน
- ให้ $v_1,v_2,\ldots,v_n$ คือ วิถีแฮมิลตันใน H แน่นอนว่า v_1 กับ v_n ไม่ติดกัน เพราะ H ไม่มีวัฏจักรแฮมิลตัน จากที่กำหนดในตอนแรก จะได้

$$deg(v_1)+deg(v_n) \geq n$$

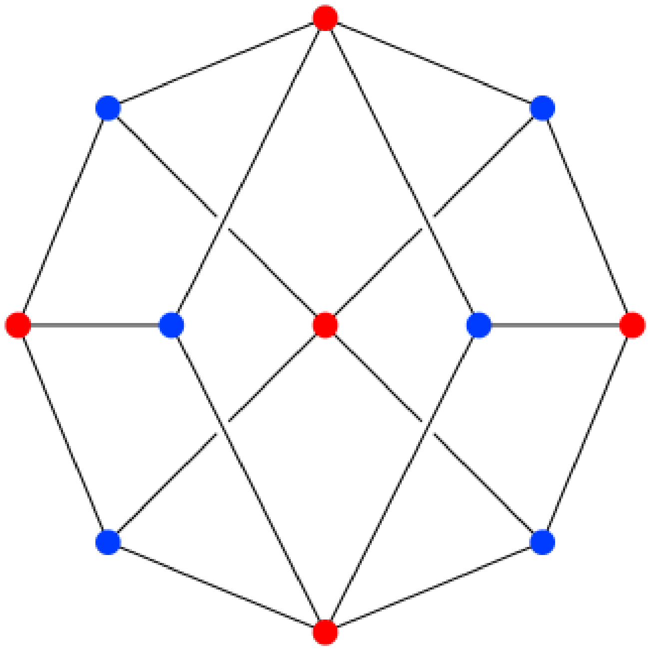
พิสูจน์ (ต่อ)

- $deg(v_1)+deg(v_n) \geq n \rightarrow deg(v_1) \geq n - deg(v_n)$
- $n - deg(v_n)$ คือจำนวนจุดยอดที่ไม่ติดกับ v_n ซึ่งมีอย่างมาก $deg(v_1)$ จุด (สังเกตว่าค่านี้ นับว่า v_n ไม่ติดกับ v_n)
- v_1 โยงไปหาจุดอื่น (เรียกว่าเซต S) เป็นจำนวน $deg(v_1)$ จุด (สังเกตเซตนี้ ไม่มี v_n เพราะถ้ามีจะได้ว่ามีวัฏจักรทันที)
- เพราะว่า มีจุดยอดอยู่อย่างมาก $deg(v_1) - 1$ จุดที่ไม่ติดกับ v_n และมีจุดใน S เป็นจำนวน $deg(v_1)$ จุด โดยทฤษฎีบทรังนกพิราบจะได้ว่า **มี**จุดใน S เชื่อมไปยัง v_n อย่างน้อยหนึ่งจุด (เรียกจุด v_x) แสดงว่ามีเส้นเชื่อม v_x v_n และ v_{x+1} v_1 (**)
- ทำให้ได้ วัฏจักร $v_1, v_2, \ldots, v_x, v_n, v_{n-1}, v_{n-2}, \ldots, v_{x+1}, v_1$
- เงื่อนไขการสร้างของ H เป็นเท็จ นั่นคือที่บอก G นั้นไม่มีวัฏจักรแฮมิลตันเป็นเท็จ

ตัวอย่าง วัฏจักรแฮมิลตัน



กราฟที่มีวัฏจักรแฮมิลตัน
เรียก กราฟแฮมิลโทเนี่ยน
(Hamiltonian graph)



กราฟนี้ไม่มีวัฏจักรแฮมิลตัน
(วิถีแฮมิลตันก็ไม่มี)

คำศัพท์ (Terminology) XX

- กราฟเชิงระนาบ (planar graph) คือ กราฟที่สามารถวาดให้เส้นเชื่อมไม่ตัดกันได้
- วิธีการวาดกราฟให้เป็นกราฟเชิงระนาบ เรียก planar representation

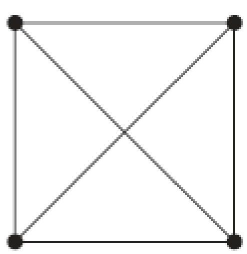


FIGURE 2 The Graph K_4.

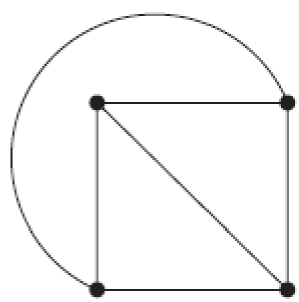


FIGURE 3 K_4 Drawn with No Crossings.

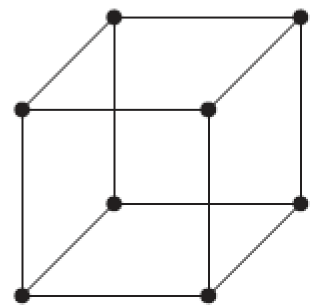


FIGURE 4 The Graph Q_3.

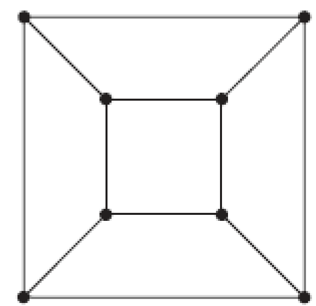


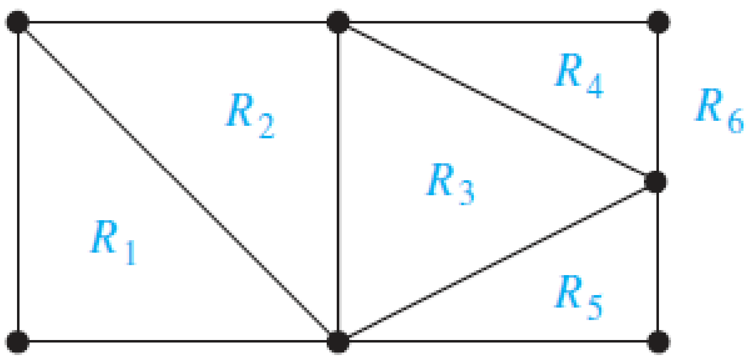
FIGURE 5 A Planar Representation of Q_3.

ทฤษฎีบท

- Euler's Formula

G เป็น กราฟเชิงระนาบที่เชื่อมโยง ที่มี e edge และ v vertex จะได้ว่า planar representation ของ G จะมี บริเวณ r เป็น

- r = e – v + 2



e = 11

v = 7

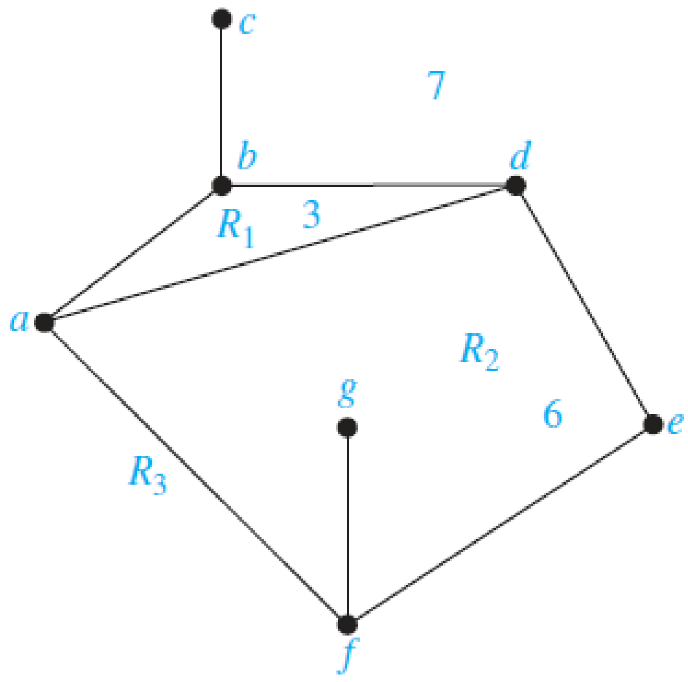
จะได้ว่า r = 11 – 7 + 2 = 6

พิสูจน์ Euler's Formular

- $r = e - v + 2 \rightarrow r - e + v = 2$
- จะพิสูจน์โดยอุปนัยเชิงคณิตศาสตร์ บน e
- Basis step: $e = 0$ นั่นคือ $v = 1$ และ $r = 1$ ถูก
- Indecutive step: ให้ เงื่อนไขเป็นจริง ทุก $e > 0$ ถึง $k - 1$
- ถ้ากราฟไม่มี cycle เลย (เป็นต้นไม้) จะได้ว่ามี $v = k$ และมี $r = 1$ นั่นคือ $1 - (k - 1) + k = 2$
- ถ้ากราฟมี cycle สมมติให้ eo เป็น edge ที่ทำให้มี cycle
- พิจารณากราฟที่ถูกนำเส้นนี้ออก จะเห็นว่ามี v จุด $e - 1$ เส้น และ $r - 1$ พื้นที่ (เดิมกราฟมี r พื้นที่ พอเอา cycle ออก พื้นที่ก็ลดลง 1)
- $(r - 1) - (e - 1) + v = 2$ (จากที่กำหนดให้)
- นั่นคือ $r - e + v = 2$ จบการพิสูจน์

คำศัพท์ (Terminology) XXI

- Degree of region คือ จำนวนเส้นเชื่อมที่ปิดล้อมพื้นที่นั้น



นับไต่ขอบจริงๆ เลย

บทแทรก

- บทแทรก 1
- ถ้า G เป็น connected planar simple graph ที่มี e edges และ v vertices โดย $v \geq 3$ จะได้ว่า $e \leq 3v - 6$
- พิสูจน์
- เพราะว่า เมื่อนับดีกรีของพื้นที่ เส้นเชื่อมทุกเส้นเชื่อมถูกนับ 2 ครั้ง และแต่ละพื้นที่ มีดีกรีอย่างน้อย 3
- จะได้ 2e = ผลรวมดีกรีของพื้นที่ $\geq 3r \rightarrow (2/3)e \geq r$
- และจาก $re - v + 2 = r \rightarrow e - v + 2 \leq (2/3)e$
- นั่นคือจะได้ว่า $e \leq 3v - 6$

บทแทรก

- บทแทรก 2
- ถ้า G เป็น connected planar simple graph และเป็น triangle free ที่มี e edges และ v vertices โดย $v \geq 3$ จะได้ว่า $e \leq 2v - 4$
- พิสูจน์
- เพราะว่า เมื่อนับดีกรีของพื้นที่ เส้นเชื่อมทุกเส้นเชื่อมถูกนับ 2 ครั้ง และแต่ละพื้นที่ มีดีกรี**อย่างน้อย 4 (ตามเงื่อนไขโจทย์)**
- จะได้ 2e = ผลรวมดีกรีของพื้นที่ $\geq 4r \rightarrow (2/4)e \geq r$
- และจาก $re - v + 2 = r \rightarrow e - v + 2 \leq (2/4)e$
- นั่นคือจะได้ว่า $e \leq 2v - 4$

ข้อสังเกต

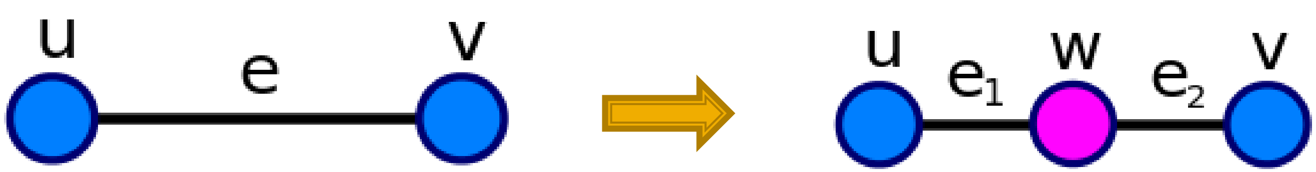
- จากบทแทรกที่ 1 จะได้ว่า K_5 ไม่เป็นกราฟเชิงระนาบ
  - กราฟ K_5 มีจุดยอด 5 จุด เส้นเชื่อม 10 เส้น $10 > 3*5 – 6 = 9$
- จากบทแทรกที่ 2 จะได้ว่า $K_{3,3}$ ไม่เป็นกราฟเชิงระนาบ
  - กราฟ $K_{3,3}$ มีจุดยอด 6 จุด เส้นเชื่อม 9 เส้น $9 > 2*6 – 4 = 8$
- กราฟ K_5 เป็นกราฟไม่เชิงระนาบที่มี จุดยอดน้อยสุด
- กราฟ $K_{3,3}$ เป็นกราฟไม่เชิงระนาบที่มี เส้นเชื่อมน้อยสุด

หยุดตรวจ

- ให้พิสูจน์บทแทรก 3 อันมีเนื้อหาต่อไปนี้
- ถ้า G เป็น connected planar simple graph แล้วต้องมีจุดยอดใน G มีดีกรีไม่เกิน 5 อย่างน้อยหนึ่งจุด
- เฉลย
- สมมติว่า ไม่มีจุดยอดที่มีดีกรี 5 เลย
- จาก ทฤษฎีบทการจับมือ จะได้ว่า ผลรวมดีกรีคือ 2e
- และจากที่สมมติในตอนต้น จึงได้ว่า 2e ≥ 6v (แต่ละจุดมีดีกรีอย่างน้อย 6 ดังนั้น 6v คือขอบล่าง)
- จากบทแทรก 1 $e \le 3v - 6 \rightarrow 2e \le 6v - 12$
- แต่ 6v > 6v – 12 นั่นคือขัดแย้งกัน ฉะนั้นบทแทรก 3 เป็นจริง

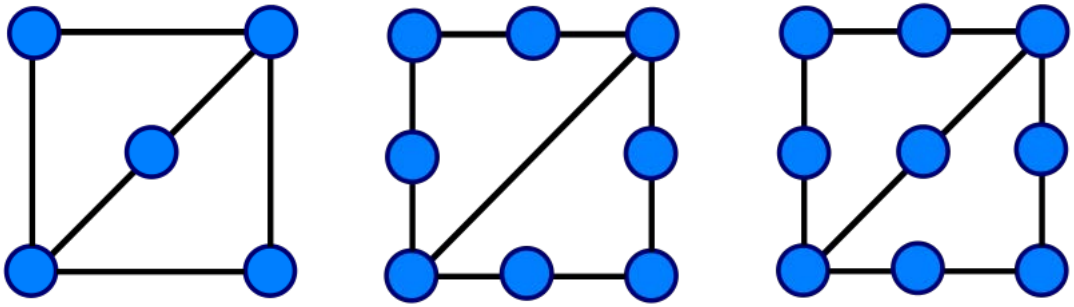
คำศัพท์ (Terminology) XXII

- กราฟ $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ จะ สมานสัณฐาน (homeomorphic) กัน ถ้ากราฟสองสามารถสมสัณฐานกันได้โดยการทำ subdivision บนกราฟทั้งสอง
- subdivision คือ การนำเอา vertex แทรกลงไปบน edge



- การทำย้อนกลับ นี้เรียกว่า smoothing

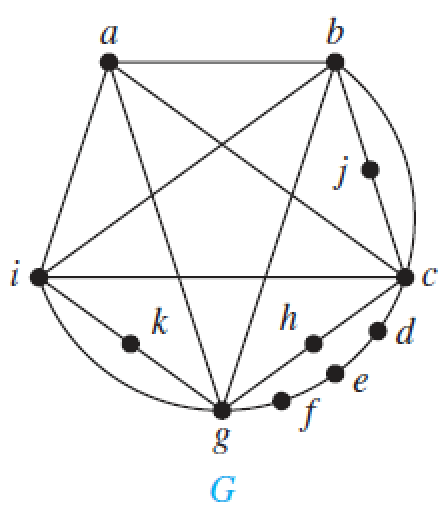
ตัวอย่าง



ทั้ง 3 กราฟนี้ สมานสัณฐานกัน

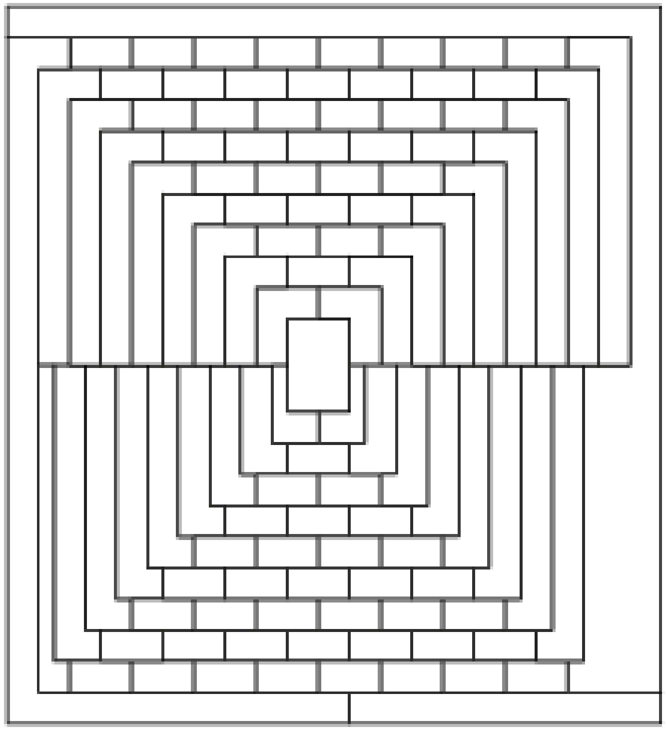
ทฤษฎีบท

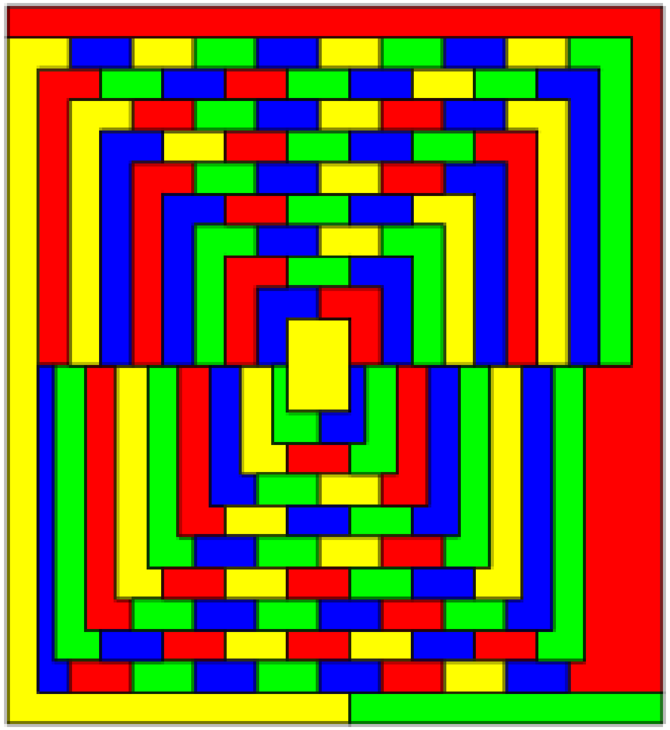
- กราฟ G จะเป็น กราฟไม่เชิงระนาบ ก็ต่อเมื่อ มี subgraph ของ G ซึ่งสมานสัณฐานกับ K_5 หรือ $K_{3,3}$



ทฤษฎีบท

- Four color theorem
- กราฟเชิงระนาบ G ใดๆ มีเลขโครมาติก χ(G) ไม่เกิน 4





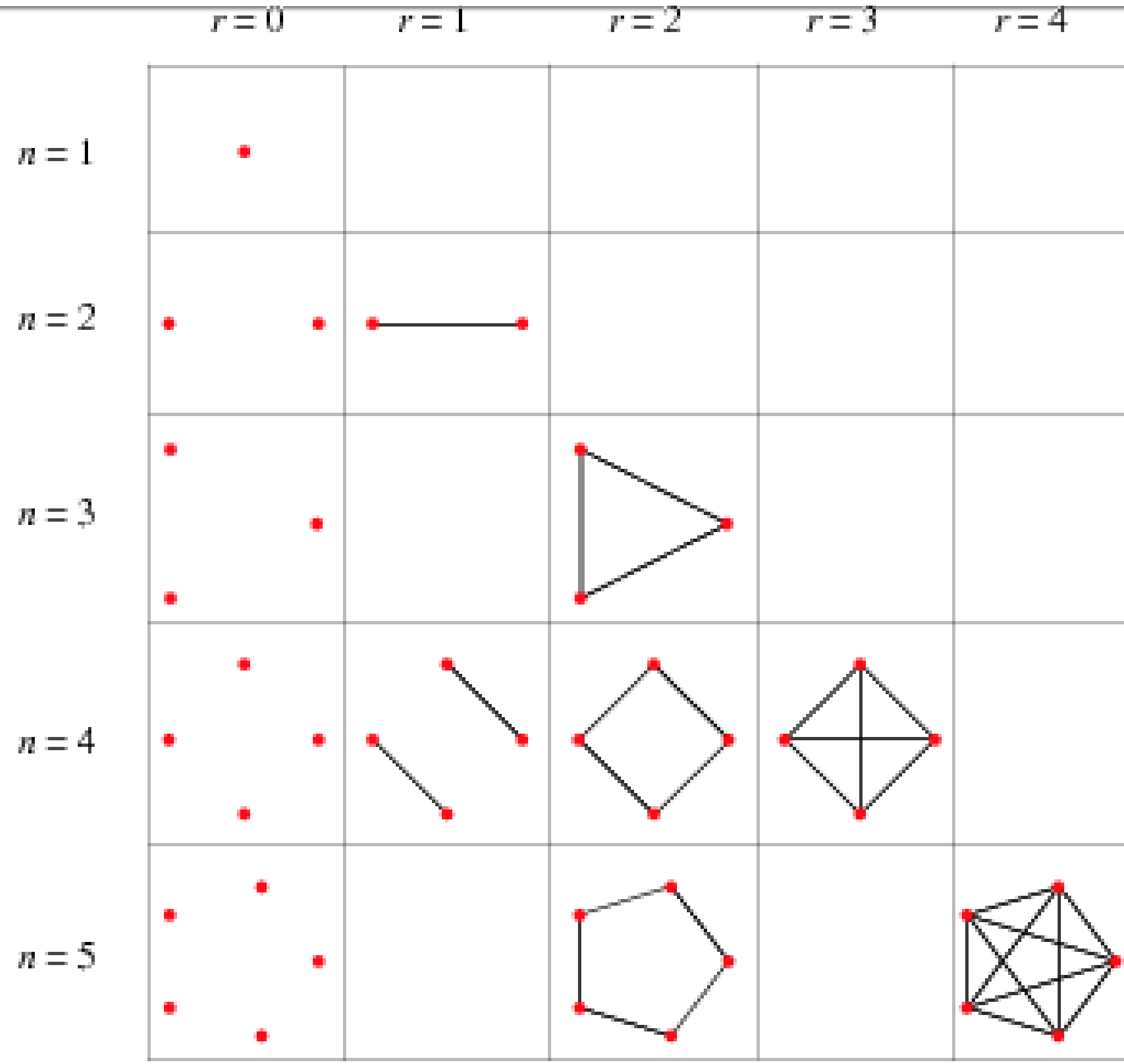
Did you know

- Martin Gardner (1975) กล่าวว่า เสนอกราฟหน้าที่แล้ว ซึ่งมีพื้นที่ 110 ส่วน มันต้องใช้ 5 สี ในวัน April Fool's
- แล้วมีคนมาระบายสีให้มันได้ ในปี 1998
- ทฤษฎีสี่สี เป็นข้อคาดเดาที่ตั้งกันมาตั้งแต่ปี 1852
- และ proof ทฤษฎีนี้ได้ในปี 2005 ด้วยคอมพิวเตอร์

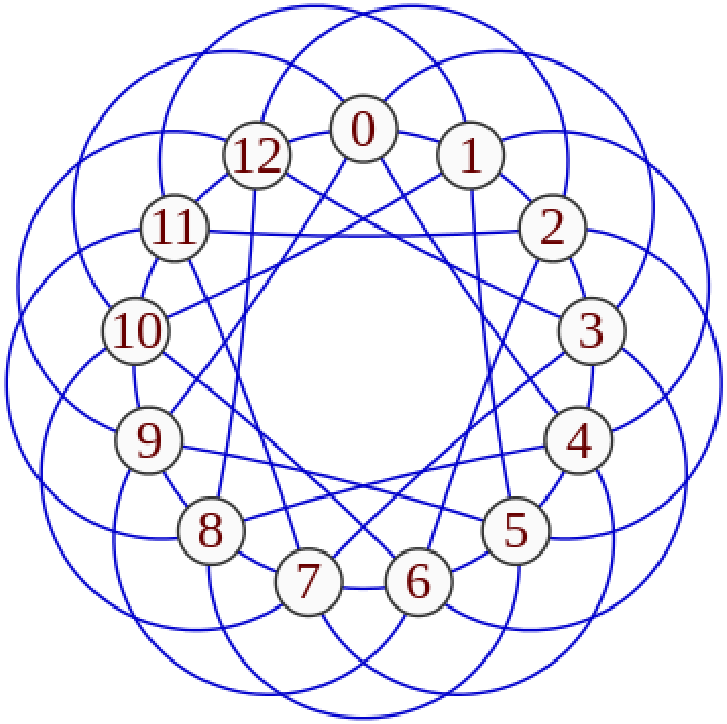
คำศัพท์ (Terminology) XXIII

- regular graph คือ กราฟที่จุดยอดทุกจุดมีดีกรีเท่ากัน
- k-regular graph คือ กราฟที่จุดยอดทุกจุดมีดีกรี k
- strongly regular graph คือ กราฟ k-regular ซึ่งมี v จุดยอด และสำหรับทุกคู่จุดยอดที่ติดกันจะมีเพื่อนบ้านร่วมกันเท่ากันคือ λ และไม่ติดกันมีเพื่อนบ้านร่วมกันคือ μ (สัญลักษณ์ srg(v,k,λ,μ))

ตัวอย่าง regular graph



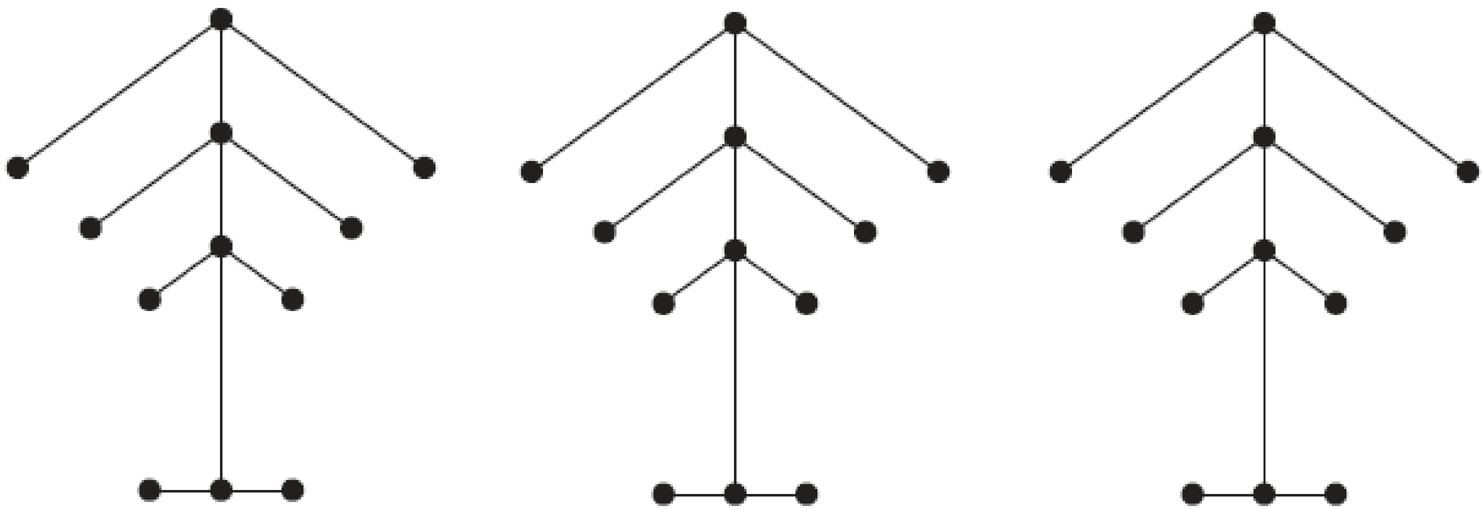
ตัวอย่าง strong regular graph



The Paley graph of order 13, a strongly regular graph with parameters srg(13,6,2,3).

คำศัพท์ (Terminology) XXIV

- ต้นไม้ (tree) คือ กราฟเชื่อมโยงที่ไม่มีวัฏจักร
- ป่า (forest) คือ กราฟที่ประกอบไปด้วยต้นไม้หลายต้น
- จุดยอดในต้นไม้ มีอีกชื่อว่า ปม (node)



ทฤษฎีบท

- กราฟไม่มีทิศทาง T จะเป็นต้นไม้ ก็ต่อเมื่อ ทุกจุดใน T มีวิถีไปถึงกันเพียงแบบเดียวเท่านั้น

พิสูจน์

→ ให้ T เป็นต้นไม้ จากนิยามจะได้ว่าทุกจุดต้องมีวิถีไปถึงกัน และจากนิยามเช่นกันจะได้ว่ามีวิถีแค่แบบเดียวด้วย เพราะถ้ามีมากกว่าหนึ่งแบบแสดงว่ามีวัฏจักร

← จากที่กำหนดแสดงว่า กราฟเชื่อมโยงกันแน่นอน เพราะทุกคู่ไปถึงกันได้ และได้ว่ากราฟนี้ไม่มีวัฏจักร เพราะถ้ามีวัฏจักรต้องมีวิถีไปได้มากกว่าหนึ่งแบบ

ทฤษฎีบท

- ต้นไม้ที่มีจุดยอด n จุด จะมีเส้นเชื่อม n – 1 เส้น

พิสูจน์

จะพิสูจน์โดยอุปนัยเชิงคณิตศาสตร์

Basis step: P(1) จริง

Inductive step: ให้ P(k) เป็นจริงพิสูจน์ P(k+1)

เพิ่มจุดยอดหนึ่งจุด แล้วเพิ่มเส้นเชื่อมเข้าไปใน k จุดยอดนั้น

เราจะได้ว่า สามารถหาวิถีจากจุดยอดใหม่นี้ กับจุดยอดเก่าทุกจุดได้ และมีแบบ

เดียว นั่นคือ P(k+1) เป็นจริง

นั่นคือ P(n) เป็นจริง

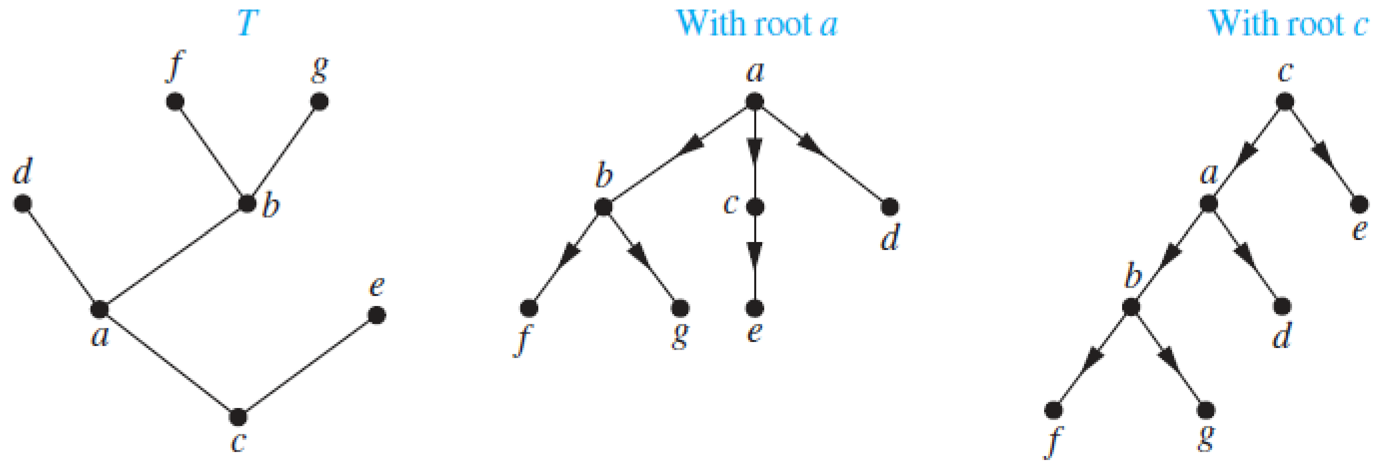
Need to know

กราฟเชื่อมโยงที่มีสมบัติต่อไปนี้ คือต้นไม้

- ไม่มีวัฏจักร
- ทุกคู่จุดยอดมีวิถีถึงกันเพียงแบบเดียว
- มีเส้นเชื่อม $n - 1$ เส้น (ซึ่งเป็นจำนวนเส้นเชื่อมน้อยสุดในการทำให้กราฟมันเชื่อมกัน)

คำศัพท์ (Terminology) XXV

- ต้นไม้มีราก (rooted tree) คือ กราฟมีทิศทาง ที่มีเส้นทางจากจุดยอดจุดหนึ่ง ซึ่งเรียกว่า ราก (root) ไปยังจุดยอดจุดอื่นๆ ทุกจุดในกราฟ และมีเส้นทางแบบเดียวเท่านั้น
- บางครั้ง เราก็วาดต้นไม้แบบมีรากด้วยเส้นเชื่อมไม่มีทิศทาง



คำศัพท์ (Terminology) XXVI

- สำหรับ จุดยอด u ,v ,w ในต้นไม้
- ถ้า ใต้ u เป็น v ,w จะได้ว่า u คือ พ่อ (parent) ของ v,w / v,w เป็นลูก (child) ของ u และ v w เป็นพี่น้องกัน (siblings=มีพ่อเป็นคนเดียวกัน)
- จุดยอดทุกจุด บนเส้นทางจากราก ไปยัง u จะเรียกจุดยอดเหล่านั้นเป็น บรรพบุรุษ (ancestor) ของ u (u ไม่เป็นบรรพบุรุษของตัวเอง) และ จุดยอดทั้งหลายที่อยู่ต่อจาก u จะเรียกว่าลูกหลาน (descendent) ของ u
- จุดยอดที่ไม่มีลูก เรียกว่า ใบ (leaf) หรือจุดปลาย (terminal vertex) และจุดยอดที่มีลูกเรียกจุดยอดภายใน (internal vertex)
- ถ้า x เป็นจุดยอดในต้นไม้แล้ว ต้นไม้ย่อย (subtree) ที่มี x เป็นราก คือ ต้นไม้ใหม่ที่เกิดจากการนำลูกหลานทั้งหมดของ x (เหมือนตอนกิ่ง)

ตัวอย่าง

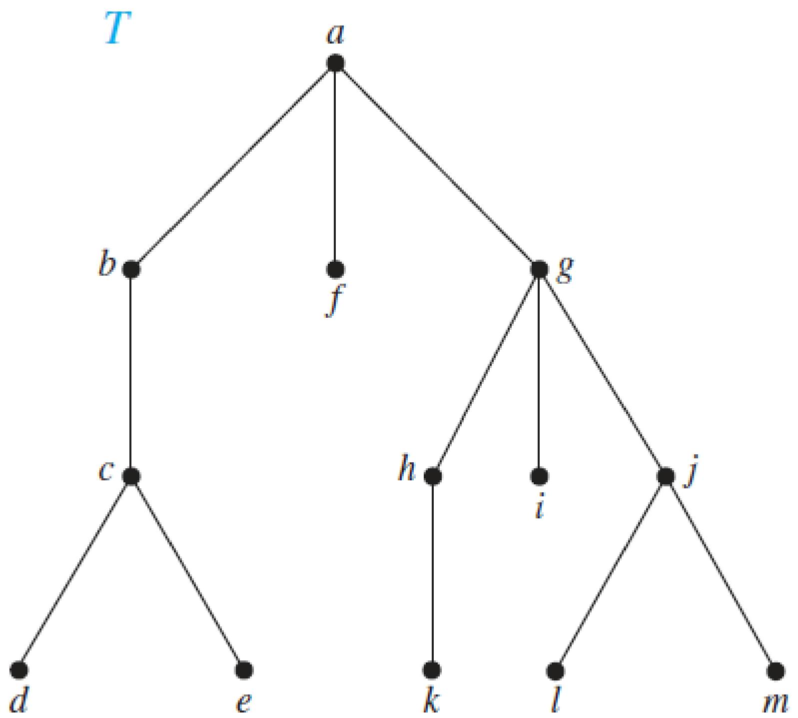


FIGURE 5 **A Rooted Tree T.**

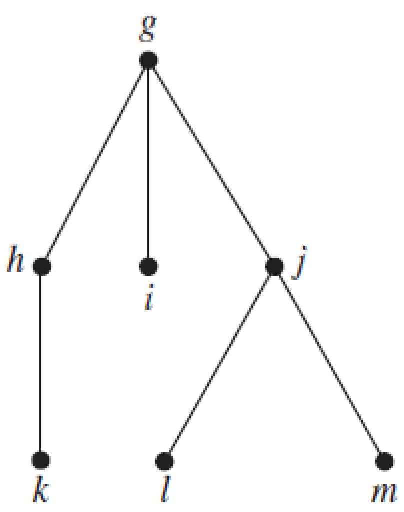


FIGURE 6 **The Subtree Rooted at g.**

คำศัพท์ (Terminology) XXVII

- จะเรียก ต้นไม้มีราก ว่า ต้นไม้ m ภาค (m-ary tree) ถ้าจุดยอดภายในทุกจุด มีลูกไม่เกิน m
- full m-ary tree คือ ต้นไม้ที่ จุดยอดภายใน มีลูกเป็น m
- complete m-ary tree คือ full m-ary tree ซึ่งมีใบเต็มชั้น
- m=2 เรียกว่า ต้นไม้ทวิภาค (binary tree)
- ต้นไม้มีราก แบบมีอันดับ (ordered rooted tree) คือ ต้นไม้ที่มีการระบุอันดับของลูกด้วย ว่าเป็นลูกตัวที่เท่าใด
- ในกรณี ordered binary tree จะเรียก ลูกทางซ้าย (left child) ลูกทางขวา (right child) และ ต้นไม้ย่อยทางซ้าย (left subtree) ต้นไม้ย่อยทางขวา (right subtree)

ตัวอย่าง

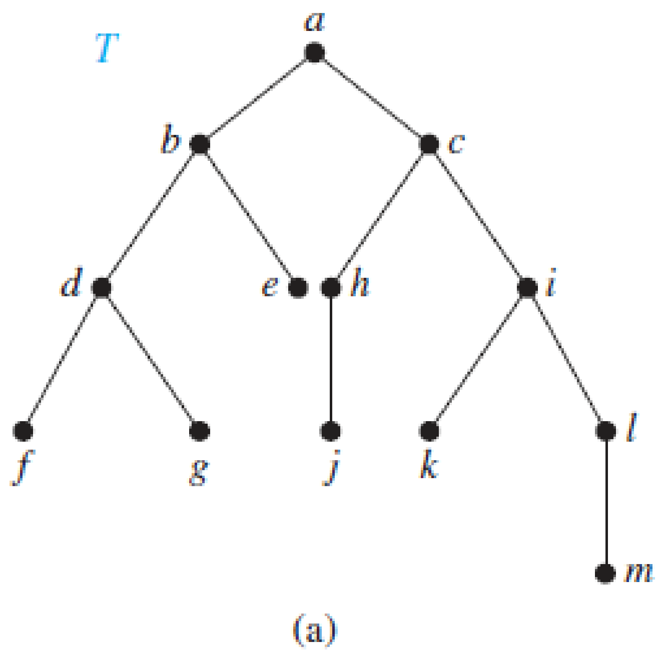


FIGURE 8 A Binary Tree T and Left and Right Subtrees of the Vertex c.

คำถาม

- full m-ary tree ที่มีจุดยอดภายใน i จุด จะมีจุดยอดกี่จุดจุด

เฉลย

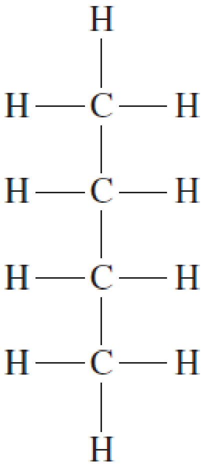
- mi + 1 จุด
- เพราะว่า ทุกจุดยกเว้นราก จะเป็นลูกของจุดยอดภายใน
- และจุดยอดภายในต้องมีลูกเป็น m
- จึงได้ว่า มีจุดยอดรวม mi + 1

คำถาม

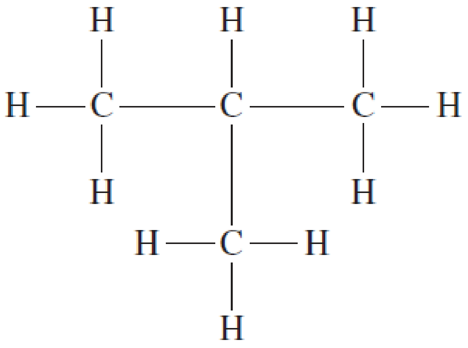
- full m-ary tree …
- ที่มีจุดยอด n จุดยอด จะมี จุดยอดภายใน และ ใบ เท่าใด
  - $i = \frac{n-1}{m}, l = \frac{[(m-1)n+1]}{m}$
- ที่มีจุดยอดภายใน i จุด จะมี จุดยอด และ ใบ เท่าใด
  - $n = mi + 1, l = (m-1)i + 1$
- ที่มีใบ l ใบ จะมี จุดยอด และ จุดยอดภายใน เท่าใด
  - $n = \frac{ml-1}{m-1}, i = \frac{l-1}{m-1}$

Did you know

- สามารถใช้ต้นไม้วาด ไฮโดรคาร์บอนอิ่มตัว (C_nH_{2n+2} Saturated hydrocarbons: alkane) โดยมี internal vertex เป็น C ซึ่งมีดีกรี 4 และมีใบเป็น H ซึ่งมีดีกรี 1
- รูปแบบต้นไม้ที่ไม่สมสัณฐานกัน คือจำนวน isomer ของมัน



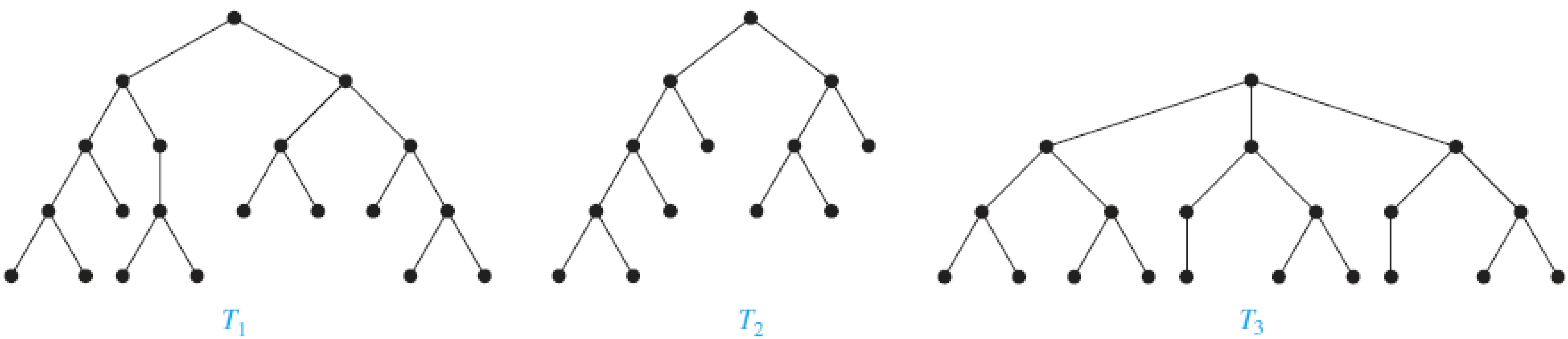
Butane



Isobutane

คำศัพท์ (Terminology) XXVIII

- ระดับชั้น (level) ของ u คือ ความยาววิถีจากรากถึงโหนด u
- ความสูง (height) ของต้นไม้ คือ ค่าของ ระดับชั้นที่มากที่สุดในต้นไม้
- ต้นไม้ได้ดุล (balanced tree) คือ ต้นไม้ที่ใบทุกใบมีระดับขั้นเป็น h หรือ h – 1



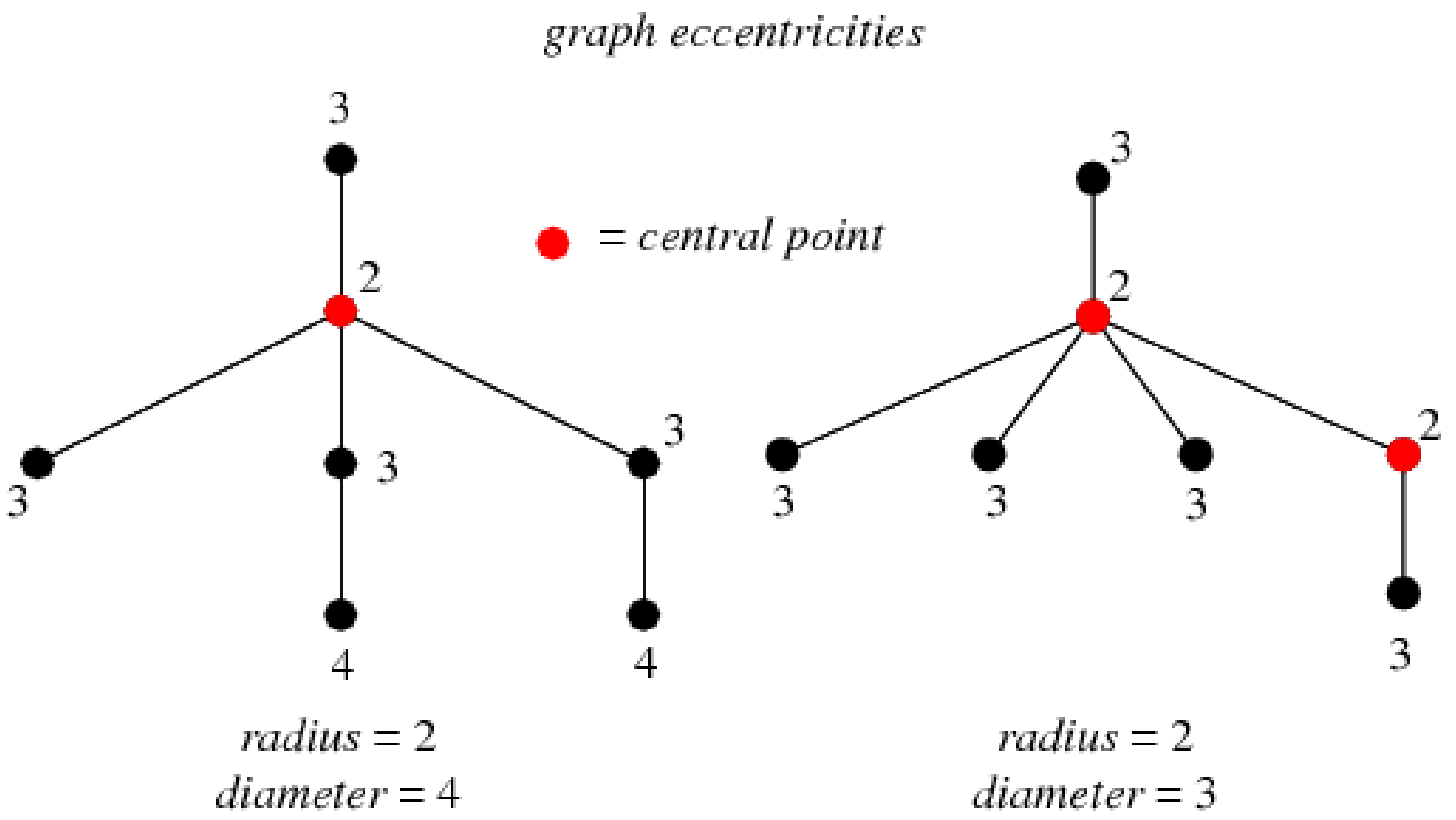
คำศัพท์ (Terminology) XXIX

- วิถีสั้นสุด (shortest path) จาก u ไป v คือ วิถีจาก u ไป v ที่มีความยาวสั้นที่สุด
- graph geodesic คือ วิถีสั้นสุดจาก u ไป v โดยเส้นเชื่อมทุกเส้นมีน้ำหนักเท่ากัน
- ความยาวของ graph geodesic ดังกล่าวเรียก ระยะทาง (graph distance) [d(u,v)]

คำศัพท์ (Terminology) XXX

- ความเยื้องศูนย์กลางของ v (graph eccentricity) $\epsilon(v)$ บนกราฟเชื่อมโยง คือ ระยะทางที่มีค่ามากที่สุดของ v ในการไปยังจุดอื่นๆ (ถ้าไม่เชื่อมโยงค่านี้คือ อนันต์)
- ค่าความเยื้องศูนย์กลางที่**มากที่สุด**ของกราฟ หรือ ระยะทางที่มากสุดในกราฟ คือ เส้นผ่านศูนย์กลาง ของกราฟ (graph diameter : d)
- ค่าความเยื้องศูนย์กลางที่**น้อยที่สุด**ของกราฟ หรือ หรือ ระยะทางที่น้อยสุดในกราฟ คือ รัศมีของกราฟ (graph radius : r)
- จุดที่ $\epsilon(v) = d$ เรียก จุดขอบ (peripheral vertex)
- จุดที่ $\epsilon(v) = r$ เรียก จุดศูนย์กลาง (central vertex)

ตัวอย่าง


graph eccentricities
3
2
= central point
3
3
3
3
4
4
radius = 2
diameter = 4
3
2
2
3
3
3
3
radius = 2
diameter = 3

ทฤษฎีบท

- ต้นไม้จะมีจุดศูนย์กลางได้อย่างมาก 2 จุด

พิสูจน์

- พิจารณาต้นไม้ซึ่งมี V เป็นจุดศูนย์กลางจุดหนึ่ง
- หากลบจุดขอบของต้นไม้ออกทั้งหมด ก็ยังคงได้ว่า V เป็นจุดศูนย์กลางอยู่
- เพราะว่า จุดยอดทุกจุดจะมีความเยื้องศูนย์กลางลดลงเท่าๆ กันหมด
- หากดำเนินการไปเรื่อยๆ ตราบเท่าที่เมื่อลบแล้วยังได้ต้นไม้ไม่ว่าง
- จะได้ว่า ท้ายสุดก็จะเหลือ จุดยอดจุดหนึ่ง หรือ เป็นกิ่งหนึ่งกิ่งเท่านั้น
- นั่นคือมีจุดศูนย์กลางได้อย่างมากสองจุด

ทฤษฎีบท

- ต้นไม้ m ภาค ที่สูง h จะมีใบอย่างมาก m^h ใบ

พิสูจน์

- พิสูจน์โดยอุปนัยเชิงคณิตศาสตร์
- Basis step: P(1) จริง เพราะรากก็มีลูกอย่างมากเป็น m
- Inductive step: ให้ P(k) จริง พิสูจน์ P(k+1)
- เพราะว่า P(k) มีใบอย่างมาก m^k ซึ่งแต่ละใบจะมีลูกได้อย่างมากคือ m นั่นคือจะได้จะเกิดใบได้อย่างมาก $m^k \times m = m^{k+1}$
- P(k+1) เป็นจริง ดังนั้น P(n) เป็นจริง

บทแทรก

- ต้นไม้ m ภาค สูง h และมีใบ l ใบ จะได้ว่า $h \geq \lceil \log_m l \rceil$

พิสูจน์

- จากทฤษฎีบทที่ว่า ต้นไม้ m ภาค ที่สูง h จะมีใบอย่างมาก m^h ใบ
- นั่นคือ $l \leq m^h \rightarrow \log_m l \leq h$
- เนื่องจาก h เป็นจำนวนเต็ม จึงสรุปว่า $h \geq \lceil \log_m l \rceil \geq \log_m l$

บทแทรก

- ต้นไม้ m ภาค ถ้าเต็ม และได้ดุล จะได้ว่า $h = \lceil \log_m l \rceil$

พิสูจน์

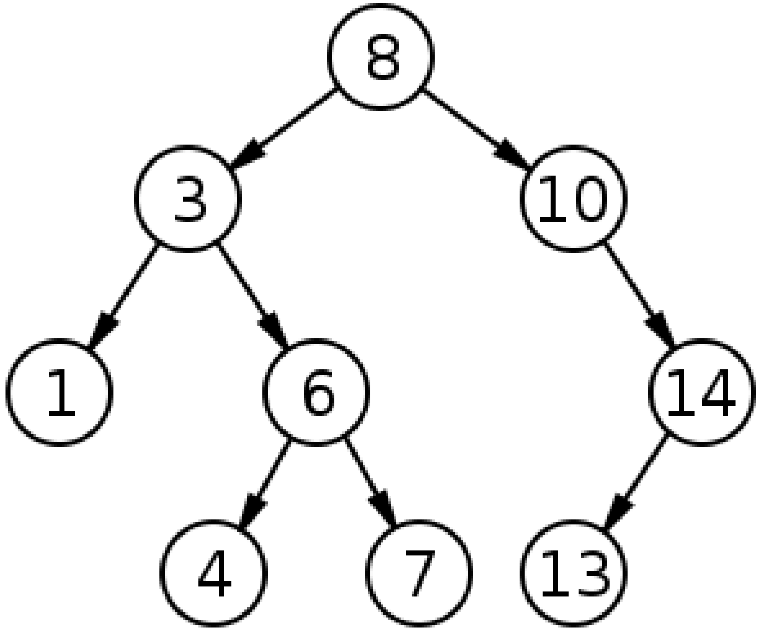
- จากทฤษฎีบทที่ว่า ต้นไม้ m ภาค ที่สูง h จะมีใบอย่างมาก m^h ใบ
- และ ต้นไม้ที่เต็ม หรือได้ดุลนั้น จะมีใบมากกว่า m^{h-1} ใบ [พิสูจน์ ?]
- นั่นคือ$m^{h-1} < l \leq m^h \rightarrow h - 1 < \log_m l \leq h$
- เนื่องจาก h เป็นจำนวนเต็ม จึงสรุปว่า $h = \lceil \log_m l \rceil$

ต้นไม้แบบต่างๆ

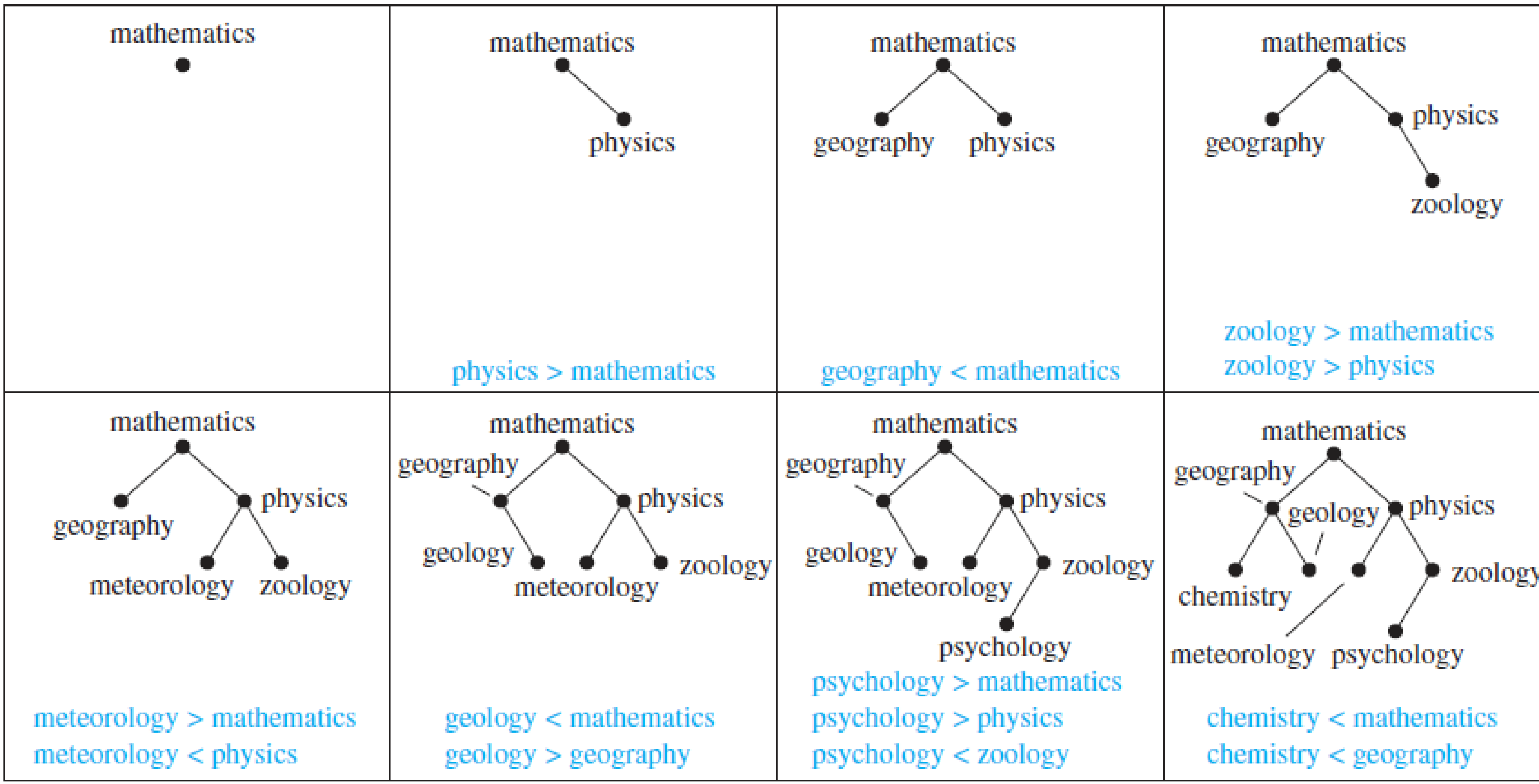
ต้นไม้ค้นหาแบบทวิภาค (Binary Serach Tree: BST)

- คือ ต้นไม้ทวิภาค ซึ่งมีข้อมูลอยู่ที่โหนด และสอดคล้องเงื่อนไข ดังนี้
- สมมติ X คือโหนดหนึ่งในต้นไม้
- ทุกข้อมูลที่อยู่ในต้นไม้ย่อยทาง**ซ้าย**ของ X ต้องมีค่า**น้อยกว่า** X
- ทุกข้อมูลที่อยู่ในต้นไม้ย่อยทาง**ขวา**ของ X ต้องมีค่า**มากกว่า** X
- ต้นไม้ย่อยทั้งสองด้าน ต้องมีสมบัตินี้อยู่
- บางครั้งเรียก ordered binary tree / sorted binary tree
- กรณีมีข้อมูลซ้ำในต้นไม้นี้ จะนิยามให้ ต้นไม้ย่อยทางซ้าย มีค่า**น้อยกว่าหรือเท่ากับ** X และต้นไม้ย่อยทางขวา มีค่า**มากกว่า** X

ตัวอย่าง Binary Search Tree



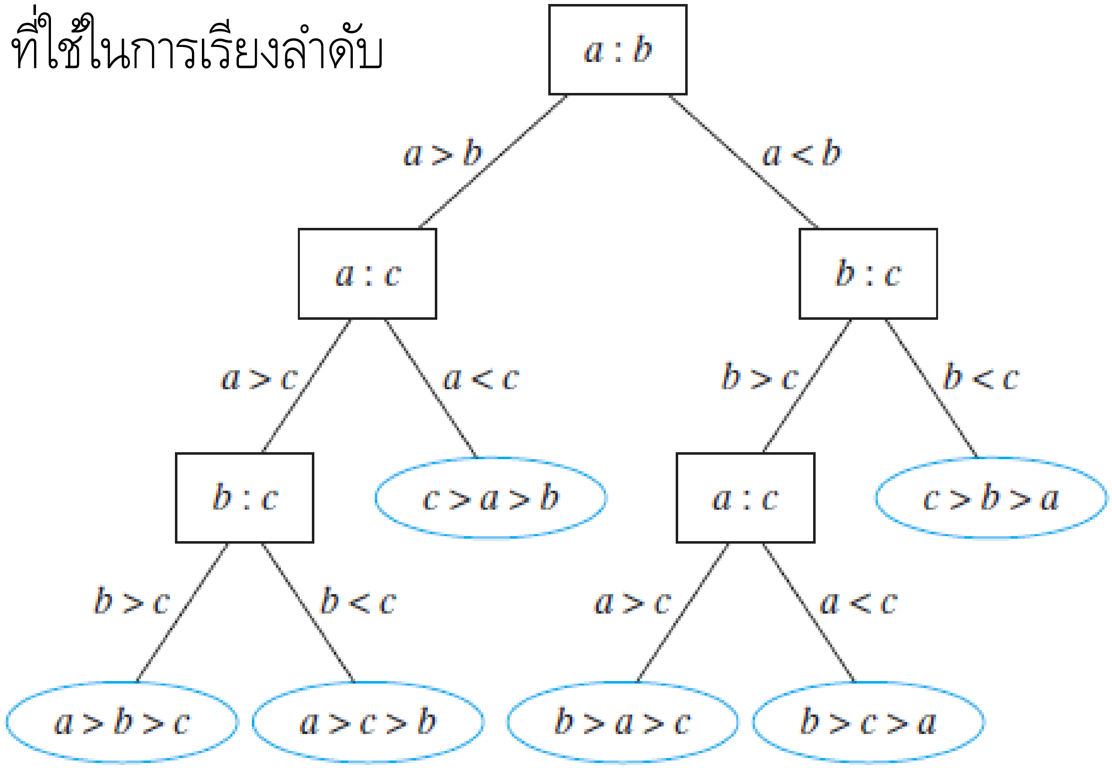
ตัวอย่าง Binary Search Tree



ต้นไม้การตัดสินใจ (Decision Tree)

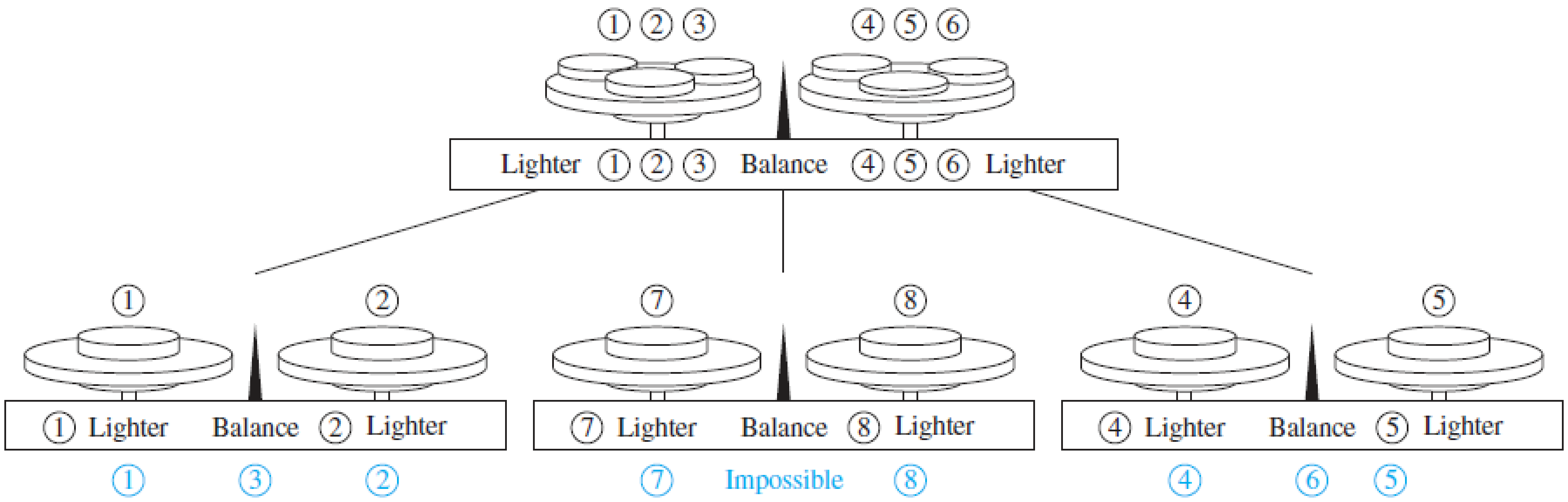
- คือ โครงสร้างการตัดสินใจที่มีลักษณะอย่างต้นไม้

ต้นไม้ตัดสินใจ ที่ใช้ในการเรียงลำดับ



ตัวอย่าง ปัญหาชั่งเหรียญ

- มีเหรียญ 7 เหรียญหนักเท่ากัน และมีอีกเหรียญหนึ่งเบากว่าเหรียญอื่น จะใช้ตาชั่งสองแขนหาเหรียญที่เบากว่านั้นได้อย่างไร โดยใช้การชั่งน้อยครั้งสุด



คำถาม

- การเรียงข้อมูล ซึ่งใช้การเปรียบเทียบระหว่างสองตัว จะต้องเปรียบเทียบกันอย่างน้อยกี่ครั้ง จึงการันตีว่าจะเรียงข้อมูลได้ถูกต้อง
- อย่างน้อย $\lceil \lg n! \rceil$ ครั้ง
- เพราะว่า ต้นไม้การตัดสินใจที่ใช้ในการเรียงข้อมูลมีใบอยู่ **n!** ใบ
- และเรารู้ว่าต้นไม้ต้นนี้ สูงอย่างน้อย $\lceil \lg n! \rceil$
- นั่นคือ ต้องใช้การเปรียบเทียบอย่างน้อย $\lceil \lg n! \rceil$
- นั่นคือ การเรียงข้อมูลที่ใช้วิธีการเปรียบเทียบทีละสองตัวนี้ จะมีประสิทธิภาพ $\Omega(n \log n)$

ต้นไม้เกม (Game Tree)

- คือ ต้นไม้ที่แสดงสถานะ (state) ของเกม ซึ่งจะมีค่ากำกับเพื่อเลือกกลยุทธ์ในการเล่น

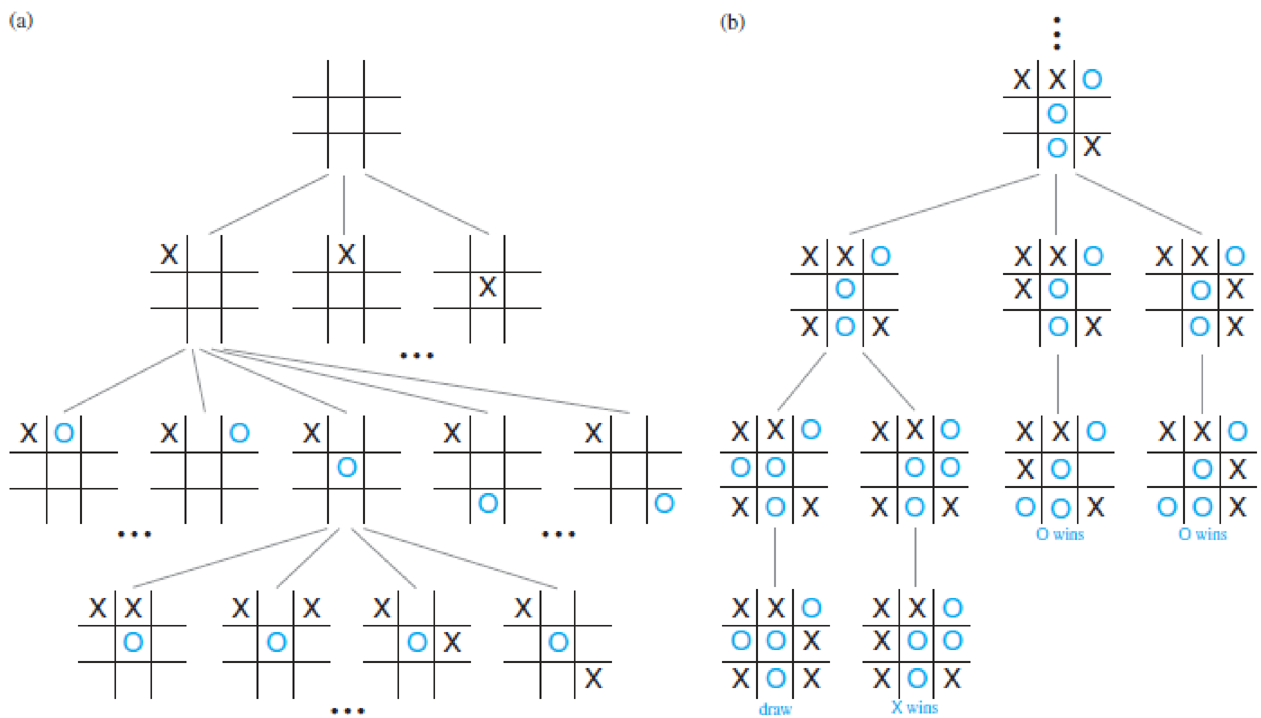


FIGURE 8 Some of the Game Tree for Tic-Tac-Toe.

ค่าของจุดยอดในต้นไม้เกม

- นิยามแบบความสัมพันธ์เวียนเกิด ดังนี้
- กำหนดค่าที่ใบของต้นไม้ (ค่านี้อาจกำหนดตามผู้เล่นคนแรก)
- สำหรับจุดยอดภายใน ค่าของมันจะเท่ากับ ค่าที่มากที่สุดของลูกของมัน

- จากการกำหนดค่าเช่นนี้ ทำให้มีกลยุทธ์ในการเล่นเกมเพื่อให้ชนะ

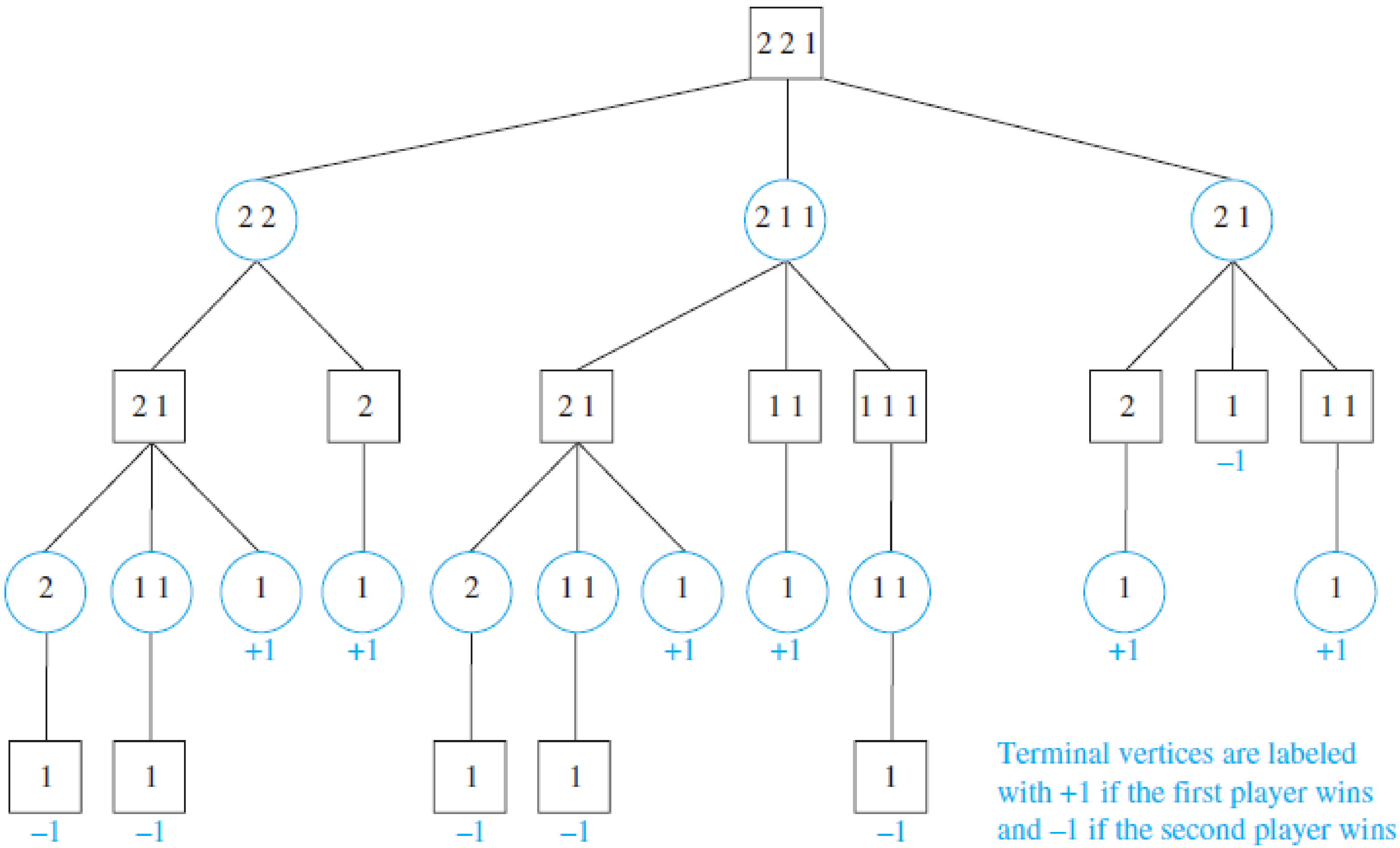
minimax strategy

- หลักการ คือ เมื่อ**เราเล่น**จะพยายาม**เลือกค่าที่มาก**ที่สุด
- ถ้าเขาเล่น เขาจะเลือกในกรณีที่เราแย่ที่สุด

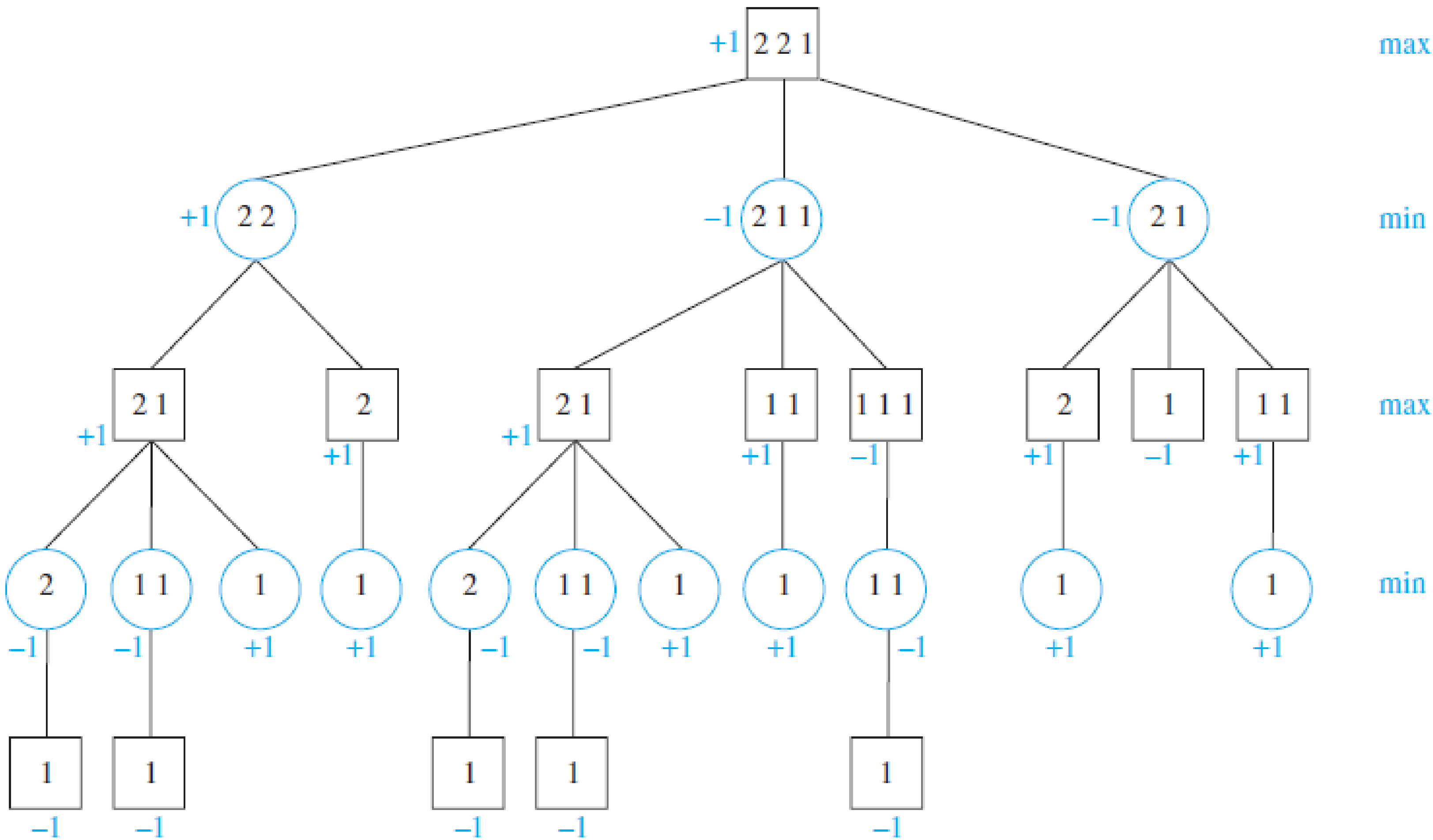
ตัวอย่าง เกมนิมกฎสลับ

- มีกองหินอยู่ **n** กอง แต่ละกองมีหินอยู่ต่างๆ กัน
- ในแต่ละตาเล่น ผู้เล่นเลือกกองหินมากองหนึ่ง แล้วหยิบหินออก กี่ก้อนจากกองนั้นก็ได้
- ใครหยิบหินก้อนสุดท้ายในเกมได้ แพ้

ตัวอย่าง เกมนิมกฎสลับ (ต้นไม้เกม)



ตัวอย่าง เกมนิมกฎสลับ (ต้นไม้เกมที่ใส่ค่าแล้ว)



การท่องไปในต้นไม้ (Tree Traversal)

- กำหนด T คือต้นไม้ ที่มีรากเป็น r แล้วมีต้นไม้ย่อยคือ $T_1,T_2,\ldots,T_n$ เรียงจากซ้ายไปขวา การท่องไปในต้นไม้ นิยามแบบเรียกตัวเอง มีลักษณะดังนี้
- ก่อนลำดับ (preorder) จะเข้าถึงข้อมูลในลักษณะ $r,T_1,T_2,\ldots,T_n$
- ตามลำดับ (inorder) จะเข้าถึงข้อมูลในลักษณะ $T_1,r,T_2,\ldots,T_n$
- หลังลำดับ (postorder) จะเข้าถึงข้อมูลในลักษณะ $T_1,T_2,\ldots,T_n,r$

คำถาม

Programming Practice

- UVa – 10004 Bicoloring
- UVa – 10596 Morning Walk
- UVa – 11080 Place the Guards
- UVa – 10054 The Necklace
- UVa – 11396 Claw Decomposition
- UVa – 117 The Postal Worker Rings Once

จบเนื้อหา

สวัสดี