

Algorithm Design and Analysis

วิชาบังคับก่อน: 204251 หรือ 204252; และ 206183 หรือ 206281

ผู้สอน: ตอน 1 ผศ. เบญจมาศ ปัญญางาม

 ตอน 2 ผศ. ดร. จักริน ชวชาติ

วันสอบปลายภาค : วันพฤหัสบดี ที่ 26 ต.ค. 66

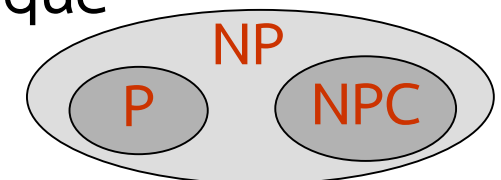
เวลา 12:00 - 15:00 น. (ตามประกาศมหาวิทยาลัย)

บทที่ 11

NP-Completeness Part III

P NP และ NP-complete

- ❑ P คือ กลุ่มของปัญหาที่มีอัลกอริทึมที่หาคำตอบ (Solution) ของปัญหาได้ภายใน Polynomial time ($\leq n^k$)
 - ▶ เช่น ปัญหา Searching, Sorting, MST(Minimum Spanning Tree)
- ❑ NP คือ กลุ่มของปัญหาที่มีอัลกอริทึมที่ทวนสอบ (Verify) คำตอบของปัญหาได้ภายใน Polynomial time
- ❑ เช่น ปัญหา Searching, Sorting, MST, SAT, Clique
- ❑ NPC คือ กลุ่มของปัญหาที่เป็น NP และทุกปัญหาอื่นๆ ใน NP สามารถลดรูปไปเป็นปัญหานั้นได้ภายใน polynomial time (for all $B \in NP$, $B \leq_p A$) เช่น ปัญหา SAT, Clique



The problem MST-Decision

- กำหนดกราฟ $G=(V,E)$ เป็นกราฟแบบมีน้ำหนักและไม่ระบุทิศทางที่ประกอบด้วย n โหนด
- Decision Problem : กราฟ G มี spanning tree ที่มีน้ำหนักรวมไม่เกินค่า k หรือไม่?
- ปัญหานี้อยู่ในกลุ่ม P หรือไม่
 - ▶ มีอัลกอริทึมที่หาคำตอบของปัญหานี้ (Solve) ภายใน polynomial time หรือไม่
- ปัญหานี้อยู่ในกลุ่ม NP หรือไม่
 - ▶ มีอัลกอริทึมที่ทวนสอบ (Verify) คำตอบของปัญหานี้ภายใน polynomial time หรือไม่

The problem MST-Decision $\in P$

กราฟ G มี spanning tree ที่มีน้ำหนักรวมไม่เกินค่า k หรือไม่?

- ปัญหานี้อยู่ในกลุ่ม P หรือไม่
 - ▶ หาอัลกอริทึมในการหาคำตอบ

SolveDecisionMST ($G(V,E),k$) {

1) เรียกใช้อัลกอริทึม Kruskal ในการหา MST T

$T = \text{KruskalAlgorithm}(G(V,E))$

2) ตรวจสอบน้ำหนักรวมใน T ที่ได้

ถ้าผลรวมน้ำหนักของทุกเส้นทางใน $T \leq k$ จริง

ให้คืนค่าจริง (True)

หากไม่จริง ให้คืนค่าเท็จ (False)

} อ. ดร. จักริน ชวชาติ
อ. เบญจมาศ ปัญญางาม

The problem MST-Decision $\in P$

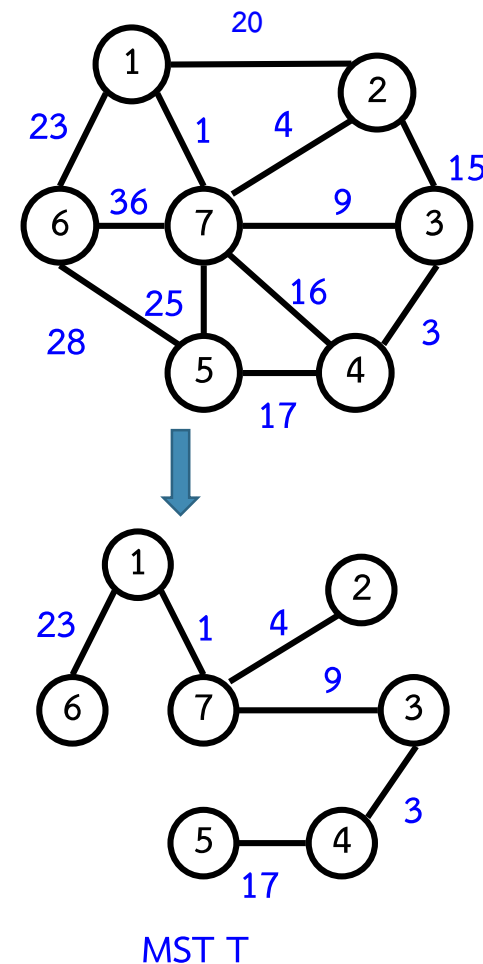
กราฟ G มี spanning tree ที่มีน้ำหนักรวมไม่เกินค่า k หรือไม่?

□ เนื่องจากอัลกอริทึม Kruskal จะคืนค่า MST T ที่มีค่าผลรวมน้ำหนัก (cost) น้อยที่สุดอยู่แล้ว

- ดังนั้นหาก cost ของ T ไม่เกินค่า k แสดงว่าเราสามารถหา spanning tree จากกราฟ G โดยที่มีน้ำหนักรวมไม่เกินค่า k ได้แน่ๆ

- อัลกอริทึม SolveDecisionMST จะคืนค่าจริง
- เช่น หากกำหนดกราฟ G ดังรูป และ k เท่ากับ 65

จะพบว่าอัลกอริทึม Kruskal จะคืนค่า MST T ที่ได้ cost เท่ากับ 57 ซึ่ง $< k$ ดังนั้นอัลกอริทึม SolveDecisionMST **จะคืนค่าจริง**



The problem MST-Decision $\in P$

กราฟ G มี spanning tree ที่มีน้ำหนักรวมไม่เกินค่า k หรือไม่?

- แต่หาก MST T ที่ได้มีค่า cost เกินค่า k แสดงว่าเราจะไม่สามารถหา spanning tree ในกราฟ G ที่มีน้ำหนักรวมไม่เกินค่า k
 - ▶ อัลกอริทึม SolveDecisionMST จะคืนค่าเท็จ
- นอกจากนี้ อัลกอริทึม SolveDecisionMST สามารถทำงานได้ภายใน Polynomial Time ($O(E \lg E)$) ดังนั้นปัญหานี้เป็นปัญหาในกลุ่ม P

The problem MST-Decision \in NP

กราฟ G มี spanning tree ที่มีน้ำหนักรวมไม่เกินค่า k หรือไม่?

□ ปัญหานี้อยู่ในกลุ่ม NP หรือไม่

▶ หาอัลกอริทึมในการ Verify คำตอบของปัญหา

VerifyDecisionMST($G(V,E)$, k , $T(V',E')$) {

1) ตรวจสอบทุกโหนดใน V' ว่ามีชื่อไม่ซ้ำกันและเป็นเซตเดียวกับโหนดในเซต V
หากไม่ใช่ให้คืนค่าเท็จ (False)

2) ตรวจสอบทุกเส้นเชื่อมระหว่างโหนดใน $T(E')$ ว่าไม่มี Cycle ทุกเส้นใน E' มีอยู่ใน E
หากไม่ใช่ให้คืนค่าเท็จ

3) ตรวจสอบผลรวมของน้ำหนักจากทุกเส้นเชื่อมระหว่างโหนดใน T ว่ามีค่าไม่เกิน k
หากไม่ใช่ให้คืนค่าเท็จ

คืนค่าจริง (True)

อ. ดร. จักริน ขวชาติ

อ. เบญจมาศ ปัญญางาม

The problem MST-Decision \in NP

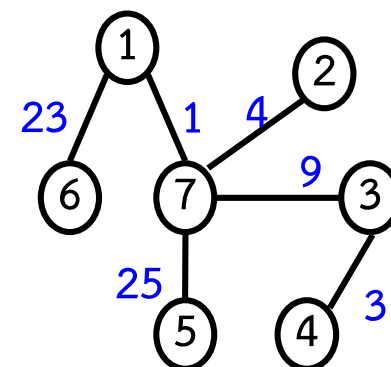
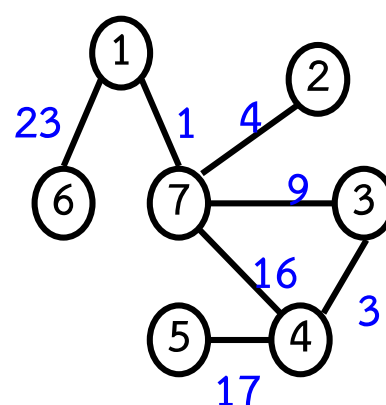
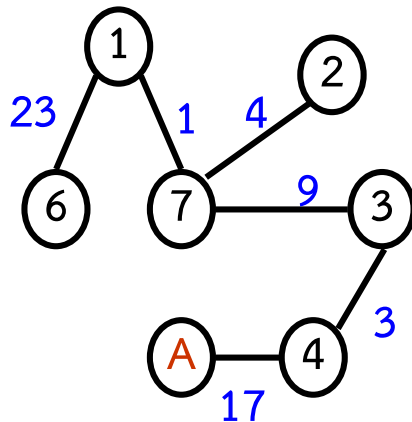
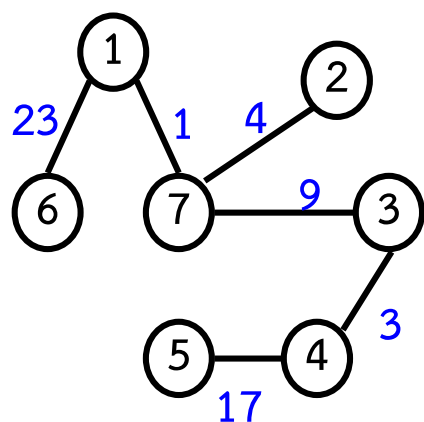
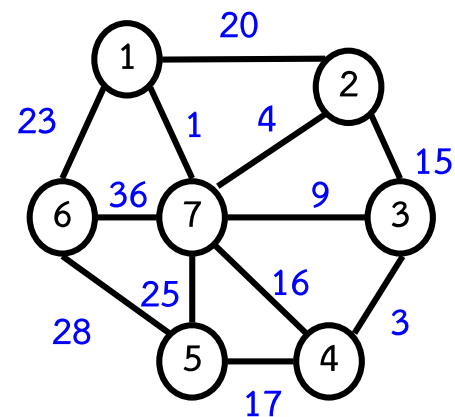
กราฟ G มี spanning tree ที่มีน้ำหนักรวมไม่เกินค่า k หรือไม่?

□ หากกำหนดกราฟ $G(V,E)$ ดังรูป และ $k = 65$

และกำหนด spanning tree $T(V',E')$ ที่ต้องการให้

อัลกอริทึม VerifyDecisionMST ทำงานเพื่อ Verify

ว่าเป็นคำตอบของปัญหาหรือไม่ ดังนี้



- จะพบว่าอัลกอริทึมให้จะคืนค่าจริง เท็จ เท็จ และจริง ตามลำดับ

อ. ดร. จักริน ชวชาติ

อ. เบญจมาศ ปัญญางาม

The problem MST-Decision \in NP

กราฟ G มี spanning tree ที่มีน้ำหนักรวมไม่เกินค่า k หรือไม่?

- นอกจากนี้อัลกอริทึม $\text{VerifyDecisionMST}(G(V,E), k, T(V',E'))$ สามารถ verify คำตอบของปัญหานี้ได้โดยใช้เวลาไม่เกิน $|V|^2$
 - ▶ ซึ่งทำงานภายใน Polynomial Time
- ดังนั้นสามารถสรุปได้ว่าปัญหานี้อยู่ใน NP

The problem VC-Decision (Vertex Cover)

- กำหนดกราฟ $G=(V,E)$ เป็นกราฟแบบไม่มีน้ำหนักและไม่ระบุทิศทางที่ประกอบด้วย n โหนด
- Decision Problem : กราฟ G มีเซตย่อย $V_{vc} \subseteq V$ ที่ครอบคลุมทุกเส้นเชื่อมใน G โดยที่ $|V_{vc}| = k$ หรือไม่

The problem VC-Decision $\in P$

กราฟ G มีเซตย่อย $V_{vc} \subseteq V$ ที่ครอบคลุมทุกเส้นเชื่อมใน G โดยที่ $|V_{vc}| = k$ หรือไม่
ปัญหานี้อยู่ในกลุ่ม P หรือไม่ จะต้องหาอัลกอริทึมในการหาคำตอบ

SolveDecisionVC ($G(V,E),k$) { // brute force

สำหรับทุกเซตของ V_i โดย $|V_i| = k$ และ $V_i \subseteq V$

ตรวจสอบว่าจำนวนเส้นเชื่อมทั้งหมดที่มีปลายด้านหนึ่งของเส้น เป็นโหนดในเซต V_i
เท่ากับขนาดของ E ในกราฟ G หรือไม่

หากใช่ ให้คืนค่าจริง (เพราะมีเซต V_i เป็น vertex cover ที่มีขนาดเท่ากับ k)

หากไม่ใช่ ให้วนตรวจสอบ V_{i+1} ต่อไปจนครบทุกเซต

คืนค่าเท็จ (เนื่องจากไม่พบ V_i ที่เป็น vertex cover)

}

The problem VC-Decision $\in P$

กราฟ G มีเซตย่อย $V_{vc} \subseteq V$ ที่ครอบคลุมทุกเส้นเชื่อมใน G โดยที่ $|V_{vc}| = k$ หรือไม่

- เนื่องจากจำนวนเซตของ V_i โดย $|V_i| = k$ และ $V_i \subseteq V$ มี 2^k แบบ โดยที่ $k \leq n$
 - ▶ ดังนั้นอัลกอริทึมจะหาคำตอบของปัญหานี้ (Solve) ภายใน exponential time
- ดังนั้นยังไม่สามารถสรุปได้ว่าปัญหานี้เป็นปัญหาในกลุ่ม P หรือไม่

The problem VC-Decision \in NPC

กราฟ G มีเซตย่อย $V_{vc} \subseteq V$ ที่ครอบคลุมทุกเส้นเชื่อมใน G โดยที่ $|V_{vc}| = k$ หรือไม่

ปัญหานี้อยู่ในกลุ่ม NPC หรือไม่

- 1) พิสูจน์ว่า ปัญหานี้เป็น NP
- 2) พิสูจน์ว่า มีปัญหา $B \leq_p$ ปัญหา VC-Decision โดยที่ B เป็นปัญหาในกลุ่ม NP-Complete

The problem VC-Decision \in NPC

□ กราฟ G มีเซตย่อย $V_{vc} \subseteq V$ ที่ครอบคลุมทุกเส้นเชื่อมใน G โดยที่ $|V_{vc}| = k$ หรือไม่

1) ปัญหานี้อยู่ในกลุ่ม NP หรือไม่

VerifyDecisionVC (Input : $G(V,E)$, k , V') {

1. ตรวจสอบว่าแต่ละโหนดใน V' มีชื่อต่างกันและ $|V'| = k$

หากไม่ใช่ ให้คืนค่าเท็จ

2. ตรวจสอบแต่ละเส้นเชื่อมระหว่างคู่โหนด $\langle u,v \rangle \in E$ ($u,v \in V$)

- หากเส้นเชื่อม $\langle u,v \rangle$ มีปลายด้านหนึ่งเป็นโหนดใดๆ ในเซต V'

(พบว่า $u \in V'$ หรือ $v \in V'$) ให้ตรวจสอบเส้นเชื่อมถัดไป

- หากไม่ใช่ ให้คืนค่าเท็จ

3. คืนค่า จริง //ตรวจสอบว่าพบครบทุกเส้นเชื่อมใน E

}

The problem VC-Decision \in NPC

กราฟ G มีเซตย่อย $V_{vc} \subseteq V$ ที่ครอบคลุมทุกเส้นเชื่อมใน G โดยที่ $|V_{vc}| = k$ หรือไม่

- 1) ปัญหานี้อยู่ในกลุ่ม NP หรือไม่
- นอกจากนี้พบว่าอัลกอริทึม $\text{VerifyDecisionVC}(G(V,E), k, V')$ สามารถ verify คำตอบของปัญหานี้ได้โดยใช้เวลาไม่เกิน $|E|$
- ซึ่งเท่ากับ n^2 ($n=|V|$) แสดงว่าอัลกอริทึมนี้ทำงานภายใน Polynomial Time
- ดังนั้นสามารถสรุปได้ว่าปัญหานี้อยู่ใน NP

The problem VC-Decision \in NPC

- กราฟ G มีเซตย่อย $V_{vc} \subseteq V$ ที่ครอบคลุมทุกเส้นเชื่อมใน G โดยที่ $|V_{vc}| = k$ หรือไม่

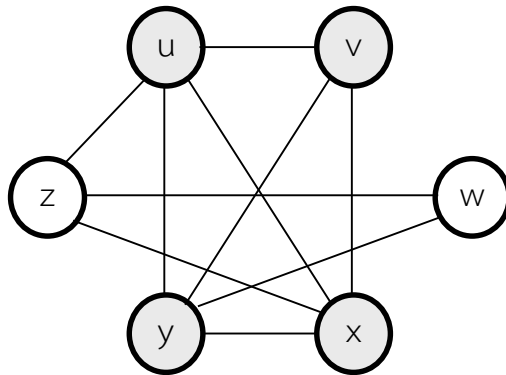
2) พิสูจน์ว่า มีปัญหา $B \leq_p$ ปัญหา VC-Decision โดยที่ B เป็นปัญหาในกลุ่ม NP-Complete

มีอัลกอริทึมในการลดรูปปัญหา Clique (สมมติว่าเรารู้ว่า Clique เป็น NPC) ไปเป็นปัญหา VC โดยใช้เวลาในการทำงานไม่เกิน polynomial time

The problem VC-Decision \in NPC

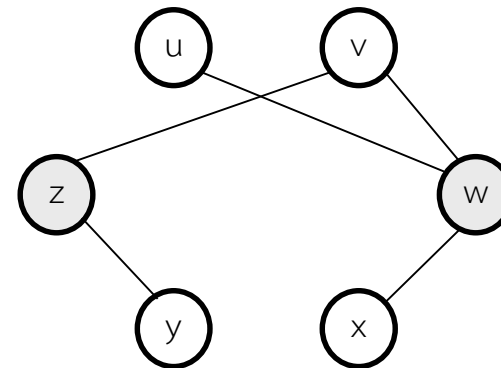
- กำหนดกราฟ $G=(V,E)$ เป็นกราฟแบบไม่มีน้ำหนักและไม่ระบุทิศทางที่ประกอบด้วย n โหนด
- ปัญหา VC-Decision : กราฟ G มีเซตย่อย $V_{vc} \subseteq V$ ที่ครอบคลุมทุกเส้นเชื่อมใน G โดยที่ $|V_{vc}| = k$ หรือไม่
- ปัญหา Clique-Decision : กราฟ G มี clique (complete subgraph) ขนาด k หรือไม่

IDEA



กราฟ G มี clique ขนาด 4

อ. ดร. จักริน ชวชาติ
อ. เบลูจมาศ ปัญญางาม



กราฟ G' มี VC ขนาด 2

The problem VC-Decision \in NPC

หาอัลกอริทึมในการลดรูปปัญหา Clique ไปเป็นปัญหา VC

- เราพิสูจน์แล้วว่า VC เป็น NP นั่นคือมีอัลกอริทึมในการ verify คำตอบของปัญหา VC ภายใน polynomial time

VerifyClique (Input : $G(V,E)$, k , V_{clique}) {

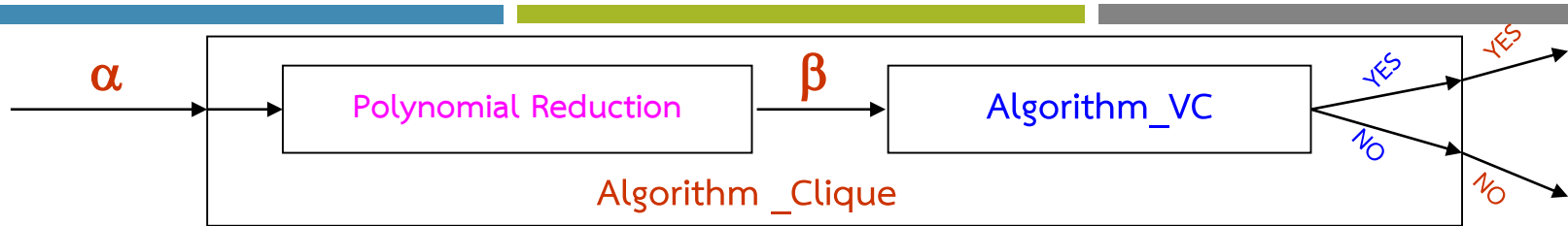
1. ขั้นตอนการ ReduceCliqueToVC

1.1 สร้างกราฟ $G'(V,E')$ โดย $E' = \{(u, v) : u, v \in V, \text{ และ } (u, v) \notin E\}$

1.2 สร้าง $V_{vc} = \{v : v \in V \text{ และ } v \notin V_{clique}\}$

2. Return(VerifyDecisionVC($G'(V,E')$, $|V| - k$, V_{vc})

• พบว่า จากข้อ 1 สามารถลดรูปปัญหา Clique ไปเป็นปัญหา VC ภายใน polynomial time (ใช้เวลาเท่ากับ $= O(|V|^2)$)

Reduction : Clique \leq_p VC

- ตัวอย่าง Instance สำหรับปัญหา Clique (α)

$$V = \{u, v, w, x, y, z\}$$

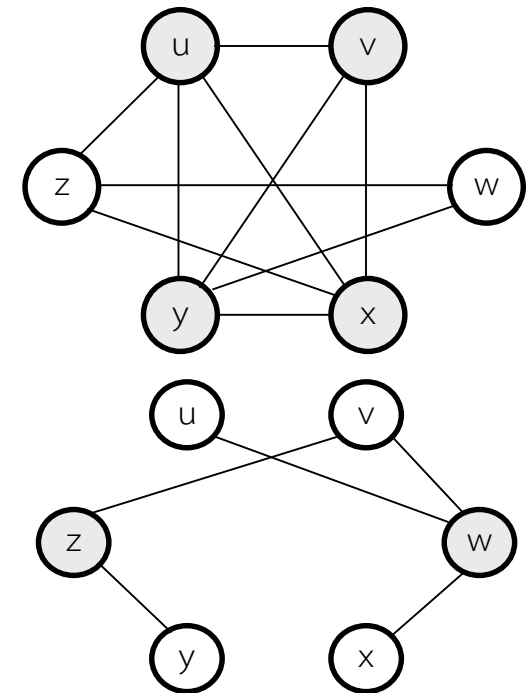
$$E = \{\{u, v\}, \{u, x\}, \{u, y\}, \{u, z\}, \{v, x\}, \{v, y\}, \{w, y\}, \{w, z\}, \{x, y\}, \{x, z\}\}$$

$$k = 4 \quad V_{\text{clique}} = \{u, v, y, x\}$$

- เปลี่ยนเป็น Instance สำหรับปัญหา VC (β)

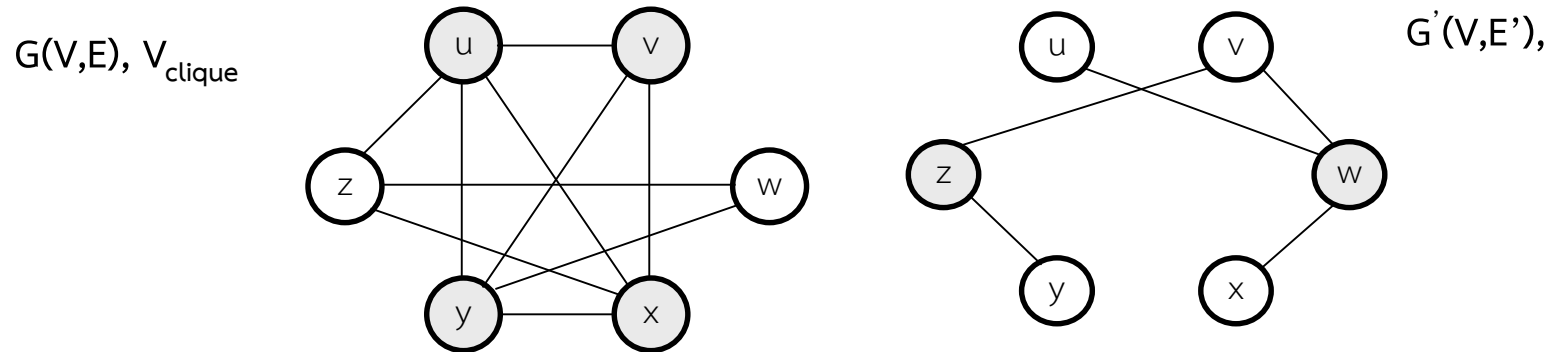
$$V = \{u, v, w, x, y, z\} \quad E = \{\{u, w\}, \{v, w\}, \{v, z\}, \{w, x\}, \{y, z\}\}$$

$$k = 2 \quad V_{\text{vc}} = \{w, z\}$$



จะพบว่า Algorithm_VC ให้คำตอบเป็น Yes สำหรับ instance β

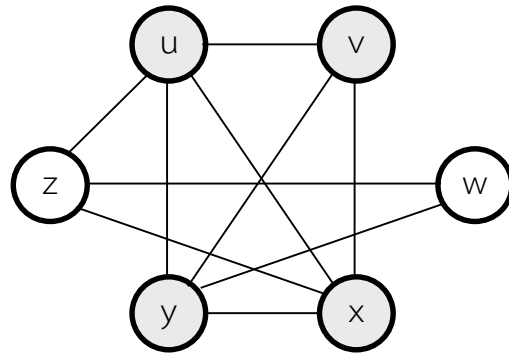
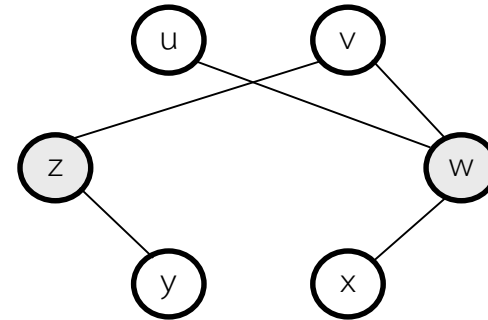
ซึ่งคำตอบสำหรับ instance α จะให้คำตอบเป็น Yes เช่นกัน

Reduction : Clique \leq_p VC

- ถ้า G มี clique ขนาด k ($V_{\text{clique}} \subseteq V, |V_{\text{clique}}| = k$) แล้วเราจะพบว่าเซต $V - V_{\text{clique}}$ เป็น VC ในกราฟ G' เสมอ
- กำหนดให้ $(u, v) \in E'$ นั่นคือ $(u, v) \notin E$ (ไม่มีเส้นเชื่อมระหว่างโหนด u และ v ในกราฟ G)
- แสดงว่าโหนด u หรือ v หรือทั้งสองโหนดจะไม่อยู่ในเซต V_{clique} แต่จะอยู่ในเซต $V - V_{\text{clique}}$ โดยที่เส้นเชื่อมระหว่างโหนด (u, v) ใน E' จะครอบคลุมโดยโหนดในเซต $V - V_{\text{clique}}$

Reduction : Clique \leq_p VC

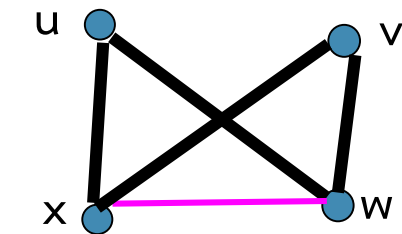
- หากเราพบ vertex cover $V_{vc} \subseteq V$ ในกราฟ G' , โดยที่ $|V_{vc}| = |V| - k$ แล้วเราจะพบ clique ขนาด k ในกราฟ G เสมอ

 $G(V,E), V_{clique}$  $G'(V,E'), V_{vc}$ 

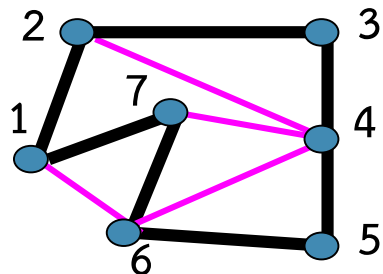
- พิจารณาแต่ละเส้นเชื่อม $(u, v) \in E'$ จะพบว่า $u \in V_{vc}$ หรือ $v \in V_{vc}$ หรือ $u, v \in V_{vc}$ แสดงว่าถ้า $u \notin V_{vc}$ และ $v \notin V_{vc}$
- นั่นคือจะไม่มีเส้นเชื่อมระหว่างโหนด u และ v ในเซต E'
 - แสดงว่า $(u, v) \in E$
 - หรือในทางกลับกัน ทุกโหนดในเซต $V - V_{vc}$ จะมีเส้นเชื่อม
- ดังนั้นเซตของโหนดใน $V - V_{vc}$ จะเป็น clique ขนาด k นั่นเอง

Hamiltonian Cycle

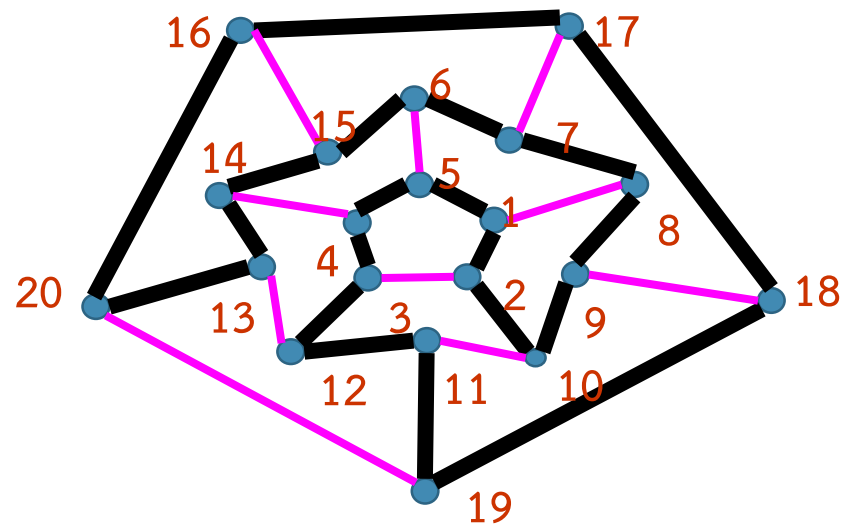
- กำหนด $G=(V,E)$ เป็นกราฟแบบไม่ระบุทิศทาง
- มีวงจร C ซึ่งผ่านทุกโหนดใน V โดยผ่านแต่ละโหนดเพียง 1 ครั้ง หรือไม่



$$C = \{u, x, v, w, u\}$$



$$C = \{1, 2, 3, 4, 5, 6, 7, 1\}$$



$$C = \{ 1, 2, 10, 9, 8, 7, 6, 15, 14, 13, 20, 16, 17, 18, 19, 11, 12, 3, 4, 5, 1 \}$$

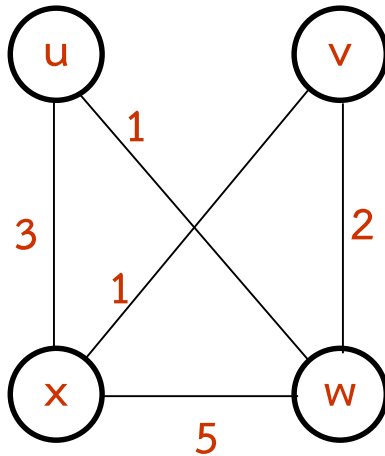
ปัญหา Hamiltonian Cycle เป็นปัญหาในกลุ่ม NPC หรือไม่

Hamiltonian Cycle \in NPC

- ปัญหา Hamiltonian Cycle เป็นปัญหาในกลุ่ม NPC หรือไม่

Assignment#8 :Traveling Salesman Problem

- กำหนด $G=(V,E)$ เป็นกราฟแบบมีน้ำหนักและไม่ระบุทิศทาง
- มีวงจรในกราฟ G ซึ่งผ่านโนดแต่ละโนดเพียง 1 ครั้ง โดยผลรวมน้ำหนักของวงจรนี้มีค่าไม่เกิน k หรือไม่



คำตอบคือ $\langle u, w, v, x, u \rangle$

โดยผลรวมน้ำหนักคือ $1 + 2 + 1 + 3 = 7$

คำถาม:

ปัญหานี้เป็น NPC หรือไม่ ?