

Data Structures and Algorithms

Chapter 1 – Introduction to Data Structures and Algorithms

COZ1302 Data Structures and Algorithms

Department of Innovation and Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University



Pasawut Cheerapakorn
Suan Sunandha Rajabhat University

Course Description:

การอธิบายขั้นตอนวิธี การเขียนผังงาน การแก้ปัญห
การคิดเชิงตรรกะ ชนิดข้อมูล ตัวแปร โครงสร้างควบคุม การนำเข้า
ข้อมูล การแสดงผล หลักการเขียนโปรแกรม วิเคราะห์ปัญหาโดยวิธี
การโปรแกรม ฝึกปฏิบัติการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์
สร้างโปรแกรมแก้ปัญหางานขนาดเล็กได้

Explanation procedure, flowchart writing, problem
solving: logical thinking; data type; variable, control structure,
input; output; principle in program writing; analyze problem by
program; practice in program writing via computer languages;
create basic problem solving program.

Algorithm Design

Design Thinking

Flowchart Design Standard

Flowgorithm

Computer Programming

Course Outline:

- **Chapter 1 – Introduction to Data Structures and Algorithms**
- Chapter 2 – Flowchart and Pseudocode
- Chapter 3 – Decision or Conditional
- Chapter 4 – Loop or Repeat
- Chapter 5 – Variables and Data Types
- Chapter 6 – Introduction to Computer Programming
- Chapter 7 – Basic Computer Programming
- Chapter 8 – Project Design

Measurement and Evaluation:

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- การมอบหมายงาน
- แบบทดสอบย่อย
- การมีส่วนร่วมในชั้นเรียน

20%

15%

5%

2. การสอบกลางภาค (Midterm Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

3. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ

20%

4. การสอบปลายภาค (Final Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดีเยี่ยม
82 – 85	A-	ดีเยี่ยม
78 – 81	B+	ดีมาก
74 – 77	B	ดี
70 – 73	B-	ค่อนข้างดี
66 – 69	C+	ปานกลางค่อนข้างดี
62 – 65	C	ปานกลาง
58 – 61	C-	ปานกลางค่อนข้างอ่อน
54 – 57	D+	ค่อนข้างอ่อน
50 – 53	D	อ่อน
46 – 49	D-	อ่อนมาก
0 – 45	F	ตก

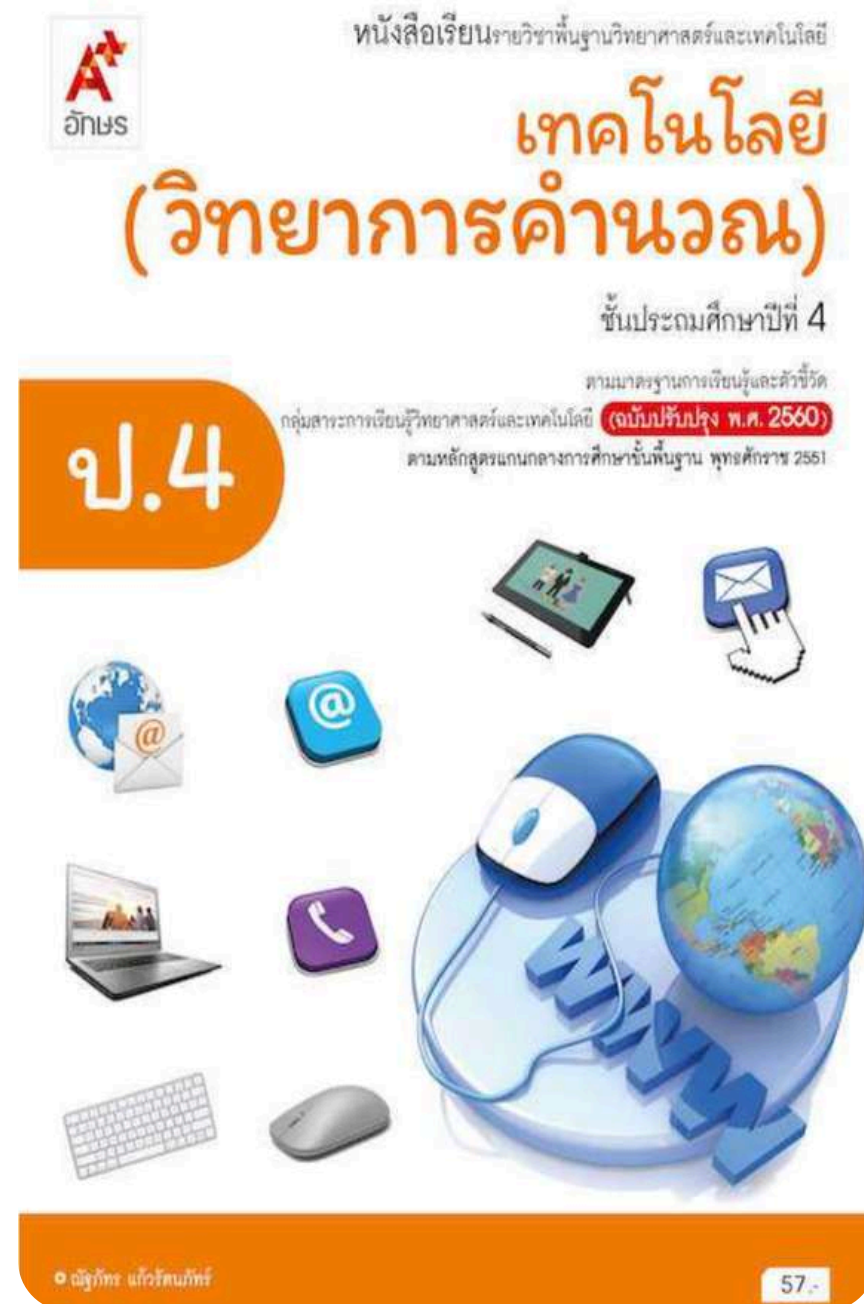
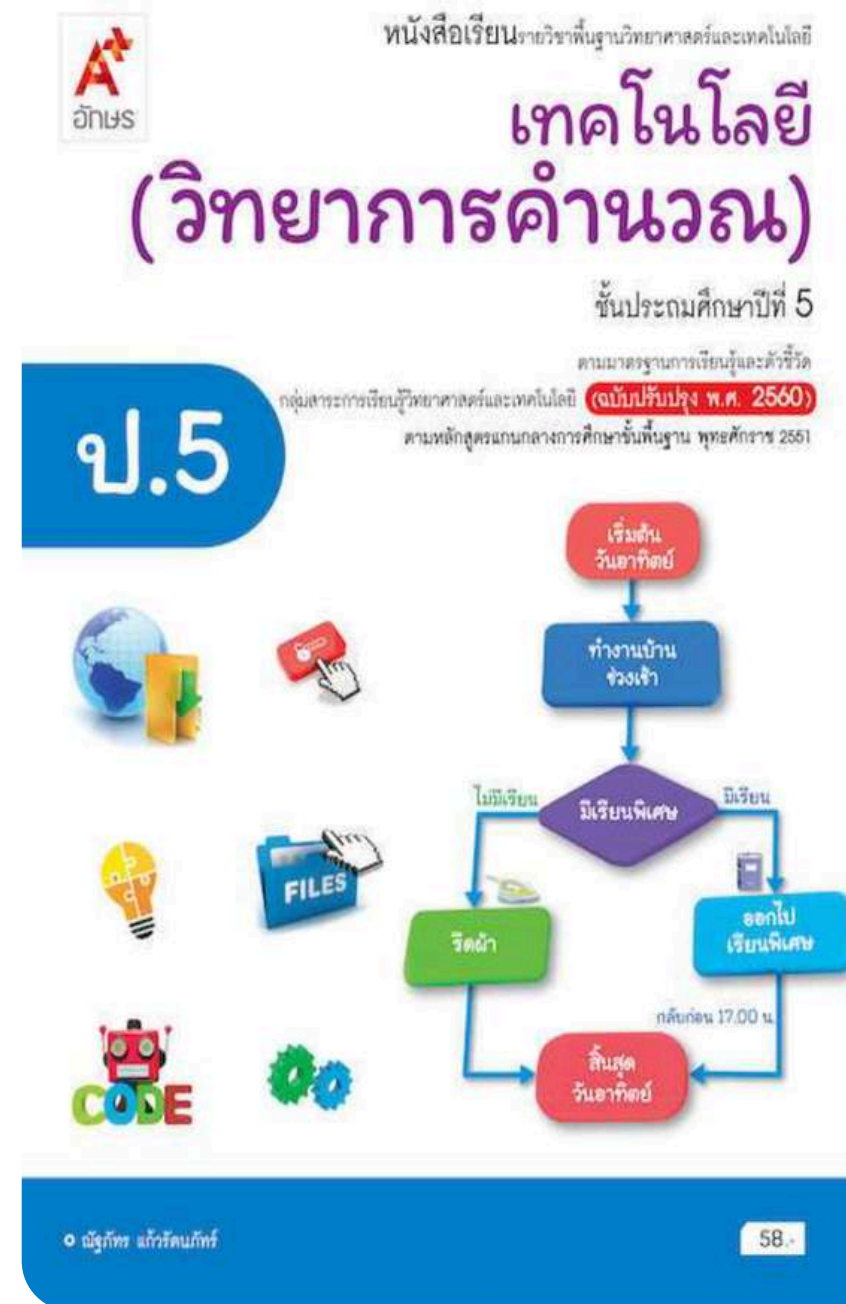
Measurement and Evaluation:

ครั้งที่ / สัปดาห์	บทเรียน / หัวข้อ
1	แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ (Introduction to Course)
2	บทที่ 1 พื้นฐานโครงสร้างข้อมูลและอัลกอริทึม (Introduction to Data Structures and Algorithms)
3	บทที่ 2 ผังงานและรหัสเทียม (Flowchart and Pseudocode)
4	บทที่ 3 การตัดสินใจหรือเงื่อนไข (Decision or Conditional)
5	บทที่ 3 การตัดสินใจหรือเงื่อนไข (Decision or Conditional) [ต่อ]
6	บทที่ 4 การวนลูปหรือทำซ้ำ (Loop or Repeat)
7	บทที่ 4 การวนลูปหรือทำซ้ำ (Loop or Repeat) [ต่อ]
8	บทที่ 5 ตัวแปรและประเภทของข้อมูล (Variables and Data Types)

Measurement and Evaluation:

ครั้งที่ / สัปดาห์	บทเรียน / หัวข้อ
9	สอบกลางภาค (Midterm Examination)
10	บทที่ 6 การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น (Introduction to Computer Programming)
11	บทที่ 6 การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น (Introduction to Computer Programming) [ต่อ]
12	บทที่ 7 พื้นฐานการเขียนโปรแกรมคอมพิวเตอร์ (Basic Computer Programming)
13	บทที่ 8 การออกแบบโครงงานทางเทคโนโลยีเพื่อการศึกษา (Project Design)
14	สอบปลายภาค (Final Examination)
15	นำเสนอและส่งโครงงาน (Project Pitching and Presentation)
16	

Learning Materials Suggestion:



เขียนและเรียบเรียงโดย: ผศ.ดร.ณัฐภัทร แก้วรัตนภัทร์

Learning Materials Suggestion:

คู่มือครู

เทคโนโลยี (วิทยาการคำนวณ) ป.1

ชั้นประถมศึกษาปีที่ 1
กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (ฉบับปรับปรุง พ.ศ. 2560)
ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551

ผู้เขียนบทนำสื่อเรียน	ผู้ตรวจบทนำสื่อเรียน	บรรณาธิการนำสื่อเรียน
นางสมศรี พงษ์ทอง	ดร.จางรุ่ง พงษ์ทอง นางณิษฐา สุริยะฉาย นางสาวพิชญ์ภาณี อังศิริพิชญ์กุล	นางณิษฐา แก้วรัตนภัทร์
ผู้เรียบเรียงคู่มือครู	บรรณาธิการคู่มือครู	
นางเอกสิริชัย ชูชาติ นางสาวระชนกชญาณี บุญตา	นางสาวกัญญาภา วัฒนพงษ์ นางสาวศศิธร คงอยู่	

หน้าปก : 1
หน้าปกหลัง : 2

อักษรส

หนังสือเรียน รายวิชาพื้นฐานวิทยาศาสตร์

เทคโนโลยี (วิทยาการคำนวณ) ป.1

ชั้นมัธยมศึกษาปีที่ 1

ตามหลักสูตรการเรียนรู้และตัวชี้วัด กลุ่มสาระการเรียนรู้วิทยาศาสตร์ (ฉบับปรับปรุง พ.ศ. 2560)
ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551

ผู้เรียบเรียง	ผู้ตรวจ	บรรณาธิการ
นางณิษฐา โอธนากรทิพย์ นางศุภนิภากร แฉิมอยู่ นางเอกสิริชัย อังศิริ	นางสมศรี พงษ์ทอง นางณิษฐา แก้วรัตนภัทร์ นางณิษฐา สุริยะฉาย	นางฉัตรกัญญา พิศมุต

หน้าปก : 1
หน้าปกหลัง : 2

อักษรส

คู่มือครู

เทคโนโลยี (วิทยาการคำนวณ) ป.2

ชั้นประถมศึกษาปีที่ 2
กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (ฉบับปรับปรุง พ.ศ. 2560)
ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551

ผู้เรียบเรียงหนังสือเรียน	ผู้ตรวจหนังสือเรียน	บรรณาธิการหนังสือเรียน
นางสมศรี พงษ์ทอง นายอาทิตย์ นามแก้ว	ดร.จางรุ่ง พงษ์ทอง นายณิษฐา สุริยะฉาย นางสาวพิชญ์ภาณี อังศิริ นางณิษฐา อังศิริ	ดร.ณิษฐา แก้วรัตนภัทร์
ผู้เรียบเรียงคู่มือครู	บรรณาธิการคู่มือครู	
ทีมวิชาการ STEM	นางสาวกัญญาภา วัฒนพงษ์ นางสาวศศิธร คงอยู่	

หน้าปก : 1
หน้าปกหลัง : 2

อักษรส

คำแนะนำการใช้

คู่มือครู รายวิชาพื้นฐาน เทคโนโลยี (วิทยาการคำนวณ) ป.2
จัดทำขึ้นเพื่อให้ครูผู้สอนใช้เป็นแนวทางวางแผนการจัดการเรียนการสอน เพื่อพัฒนาผลสัมฤทธิ์ทางการเรียนและประเมินคุณภาพผู้เรียนตามนโยบายของสำนักงานคณะกรรมการการศึกษาขั้นพื้นฐาน (สพฐ.)

Chapter Overview
โครงสร้างแผนและแนวทางการประเมินผู้เรียนประจำหน่วยการเรียนรู้

Chapter Concept Overview
สรุปสาระสำคัญประจำหน่วยการเรียนรู้

โซน 1 ช่วยครูจัดการเรียนการสอน
แนวทางการจัดการเรียนการสอนให้แก่มัธยมศึกษาตอนต้นและนำขึ้นเรียนการสอน และการจัดการเรียนการสอนอย่างยืดหยุ่นเพื่อให้ได้นักเรียนบรรลุผลสัมฤทธิ์ตามตัวชี้วัด

โซน 2 ช่วยครูเตรียมสอบ
ประกอบด้วยข้อสอบประเภทต่าง ๆ ที่เป็นประโยชน์เพื่อช่วยพัฒนาการเรียนการสอนของครูผู้สอน

เก็ธแถม
ความรู้เสริมสำหรับครู ข้อเสนองานและข้อสังเกต แนวทางการจัดการเรียนการสอนเพื่อประโยชน์ในการจัดการเรียนการสอน

บันทึกความรู้
ความรู้เพิ่มเติมจากเนื้อหา เพื่อให้ครูนำไปใช้อธิบายให้นักเรียน

สื่อ Digital
การแนะนำแหล่งที่หาสื่อ Digital ต่าง ๆ

แนวทางการวัดและประเมินผล
การเสนอแนะแนวทางในการวัดและประเมินผลนักเรียนเพื่อสอดคล้องกับแผนการสอน

โซน 1
การแก้ปัญหาอย่างมีประสิทธิภาพ

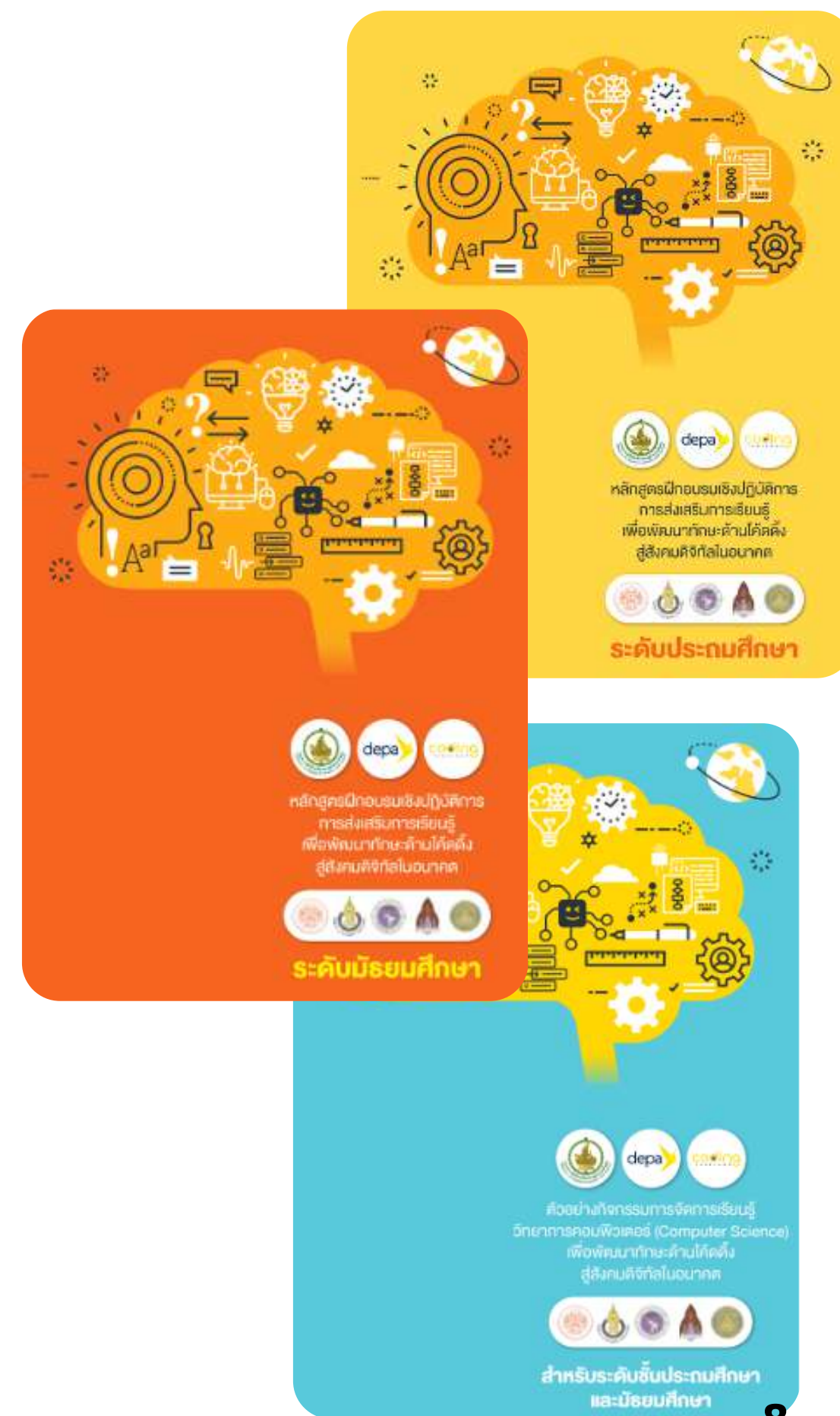
โซน 2

โซน 3

บรรณาธิการหนังสือเรียน: ผศ.ดร.ณิษฐภัทร แก้วรัตนภัทร์

Learning Materials:

1. หลักสูตรฝึกอบรมเชิงปฏิบัติการการส่งเสริมการเรียนรู้เพื่อพัฒนาเพื่อพัฒนาทักษะด้านโค้ดดิ้งสู่สังคมดิจิทัลในอนาคต สำหรับชั้นประถมศึกษา ดาวนีโหลดที่ <https://www.depa.or.th/storage/app/media/file/coding1.pdf>
2. หลักสูตรฝึกอบรมเชิงปฏิบัติการการส่งเสริมการเรียนรู้เพื่อพัฒนาเพื่อพัฒนาทักษะด้านโค้ดดิ้งสู่สังคมดิจิทัลในอนาคต สำหรับชั้นมัธยมศึกษา ดาวนีโหลดที่ <https://www.depa.or.th/storage/app/media/file/coding2.pdf>
3. ตัวอย่างกิจกรรมการจัดการเรียนรู้วิทยาการคอมพิวเตอร์ (Computer Science) เพื่อพัฒนาทักษะด้านโค้ดดิ้งสู่สังคมดิจิทัลในอนาคต สำหรับชั้นประถมศึกษาและมัธยมศึกษา ดาวนีโหลดที่ <https://www.depa.or.th/storage/app/media/file/coding3.pdf>



วิทยาการคำนวณเรียนเกี่ยวกับอะไร?

- วิทยาการคำนวณ เรียนรู้เกี่ยวกับการพัฒนาผู้เรียนให้มีความรู้ความเข้าใจ มีทักษะการคิดเชิงคำนวณ การคิดวิเคราะห์ แก้ปัญหาเป็นขั้นตอนและเป็นระบบ ประยุกต์ใช้ความรู้ด้านวิทยาการคอมพิวเตอร์ และเทคโนโลยีสารสนเทศสื่อสารในการแก้ปัญหาที่พบในชีวิตจริงได้อย่างมีประสิทธิภาพ

(อ้างอิงตามตัวชี้วัดและสาระการเรียนรู้แกนกลาง กลุ่มสาระการเรียนรู้วิทยาศาสตร์ (ฉบับปรับปรุง พ.ศ. 2560) ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551, สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สสวท.) กระทรวงศึกษาธิการ)

พื้นฐานการคิดเชิงคำนวณ
Computational Thinking

พื้นฐานเทคโนโลยีดิจิทัล
Digital Technology

พื้นฐานการรู้เท่าทันสื่อและดิจิทัล
Media and Digital Literacy

วิทยาการคำนวณเรียนเกี่ยวกับอะไร?

- การคิดคำนวณและการให้เหตุผล (**Computational Thinking**)
- เรียบเรียงความคิดให้คิดเป็นขั้นตอน (**Algorithm Design**)
- คิดเชิงออกแบบ (**Design Thinking**) เพื่อหาวิธีการที่ดีที่สุด เช่น ใช้เวลาสั้นที่สุด, ใช้คำสั่งน้อยที่สุด, ใช้ทรัพยากรและพลังงานน้อย (**Less but More**)
- หาวิธีจัดการที่สะดวกที่สุด เช่น การปรับค่าตัวแปรแทนที่จะป้อนคำสั่งแบบ Manual, การแก้ไขค่าของตัวแปร, การใช้สูตรทางคณิตศาสตร์
- กระบวนการคิดเชิงสร้างสรรค์ (**Creativity Thinking**)
- ประโยชน์ มูลค่า ของขั้นตอนที่คิดขึ้น (โปรแกรมที่พัฒนามาขึ้น)

การคิดเชิงคำนวณ - Computational Thinking



1. การคิดย่อยปัญหา (Decomposition)

- คือ การพิจารณาแบ่งปัญหาให้มีความซับซ้อนน้อยลง เพื่อช่วยให้การวิเคราะห์และออกแบบวิธีการแก้ปัญหาคำทำได้สะดวก
- ปรับมุมมองปัญหา
- จากมองภาพใหญ่ เป็นมองภาพย่อย
- จากมองรวมกัน เป็นมองแยกจากกัน

การคิดเชิงคำนวณ - Computational Thinking



2. การคิดเป็นนามธรรม (Abstraction)

- คือ การคัดแยกคุณลักษณะที่สำคัญออกจากรายละเอียดปลีกย่อย เพื่อให้ได้ข้อมูลที่จำเป็นและเพียงพอในการแก้ปัญหา
- ลองทำจัดสิ่งที่ไม่จำเป็นออกไป
- ลดสิ่งลวงตา มองเห็นสิ่งที่จำเป็นจริงๆ

การคิดเชิงคำนวณ - Computational Thinking



3. การรู้จำรูปแบบ (Pattern Recognition)

- คือ การวิเคราะห์และแสดงความสัมพันธ์ของสิ่งต่างๆ ที่มีลักษณะสำคัญร่วมกันอย่างมีเงื่อนไข
- สังเกตสิ่งที่ปรากฏชัดเจน และ ช้อนเร้น
- สังเกตสิ่งที่ปรากฏ มีอะไรคล้ายกัน
- สังเกตสิ่งที่ปรากฏ มีอะไรต่างกัน
- เลือกรูปแบบที่สังเกตได้และมีความเป็นไปได้ที่จะใช้ในการแก้ปัญหา

การคิดเชิงคำนวณ - Computational Thinking



4. การออกแบบอัลกอริทึม (Algorithm Design)

- เริ่มต้นขั้นตอน ขั้นตอนถัดไป และสิ้นสุดที่ขั้นตอนใด
- ลองคิดออกแบบหลายๆ ขั้นตอน และเปรียบเทียบ เพื่อหาขั้นตอนที่เหมาะสมที่สุด

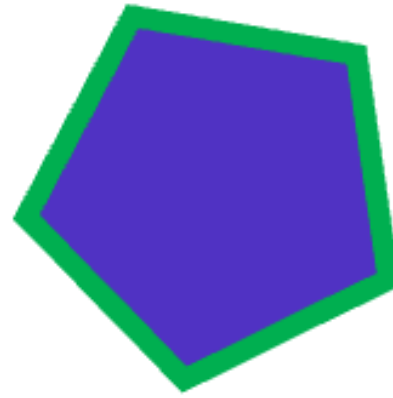
ปัญหา: เรียงลำดับรูปภาพต่อไปนี้ ยังไงดี?



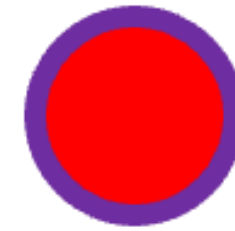
A



B



C

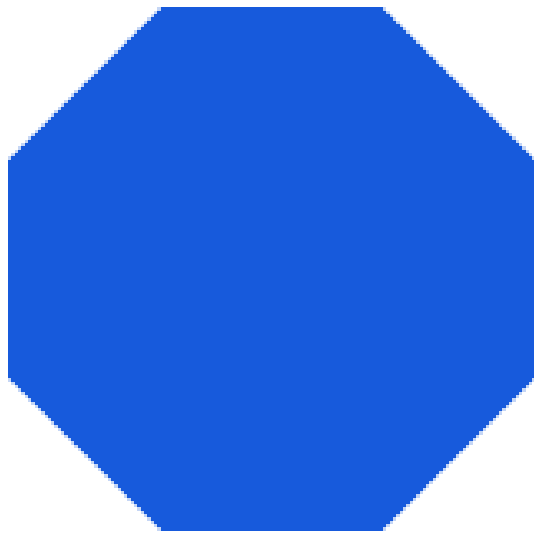


D

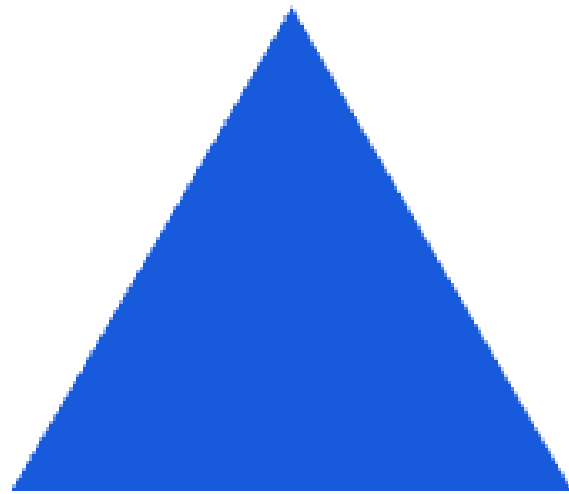


E

ปัญหา: เรียงลำดับรูปภาพต่อไปนี้ ยังไงดี?



A



B



C

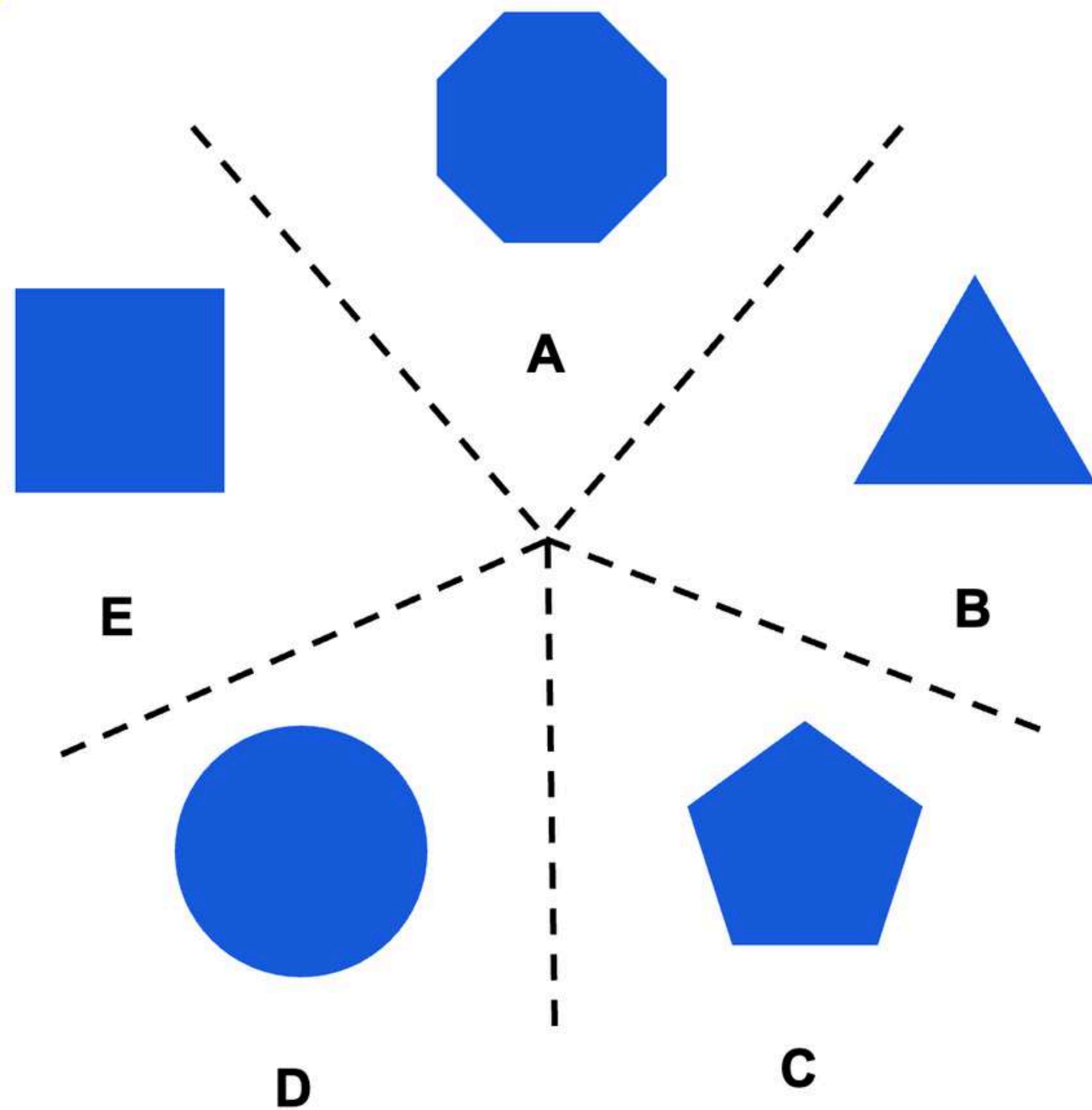


D



E

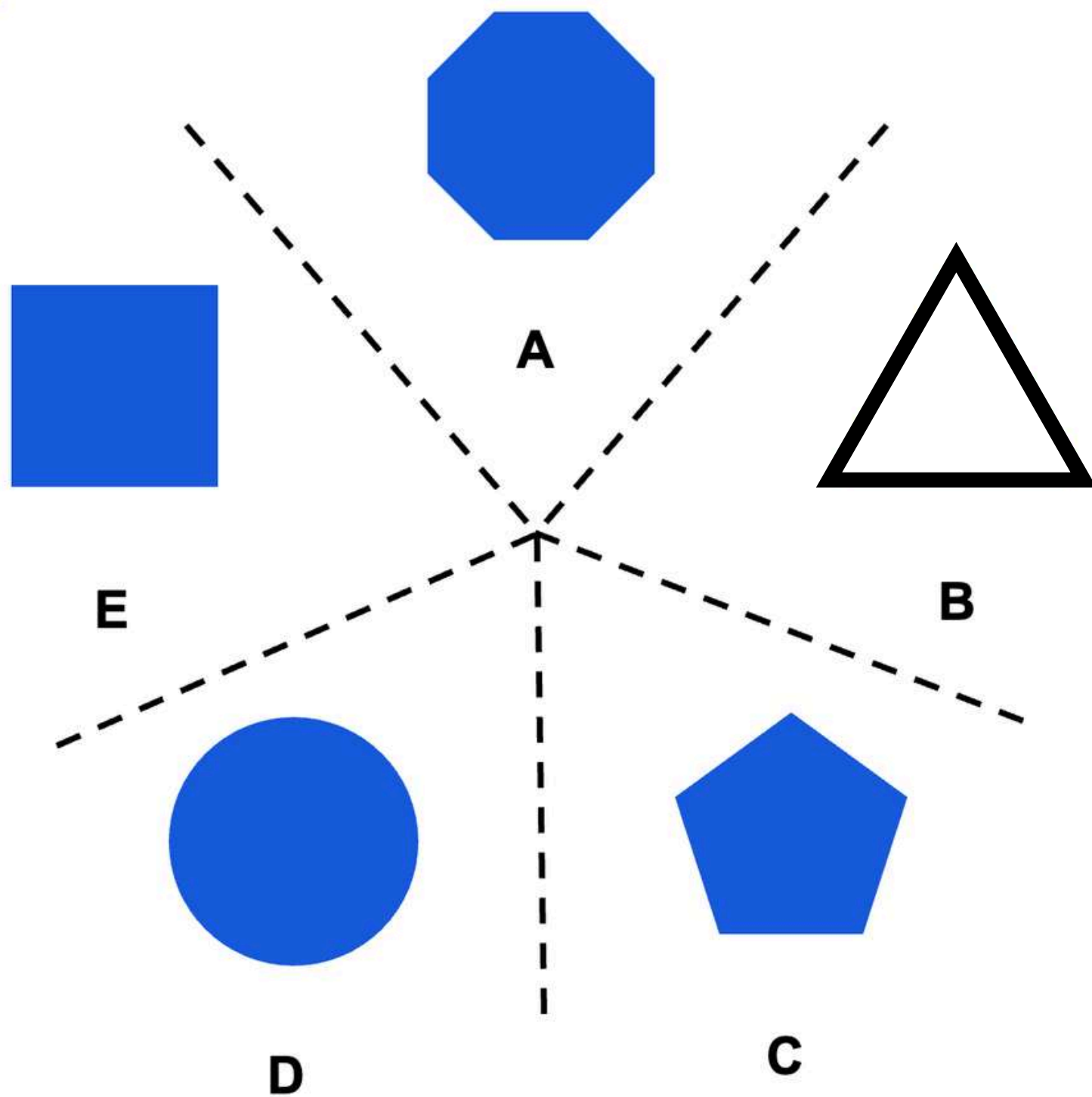
ปัญหา: เรียงลำดับรูปภาพต่อไปนี้ ยังไงดี?



1. การคิดย่อยปัญหา (Decomposition)

- คือ การพิจารณาแบ่งปัญหาปัญหาให้มีความซับซ้อนน้อยลง เพื่อช่วยให้การวิเคราะห์และออกแบบวิธีการแก้ปัญหาคำทำได้สะดวก
- ปรับมุมมองปัญหา
- จากมองภาพใหญ่ เป็นมองภาพย่อย
- จากมองรวมกัน เป็นมองแยกจากกัน

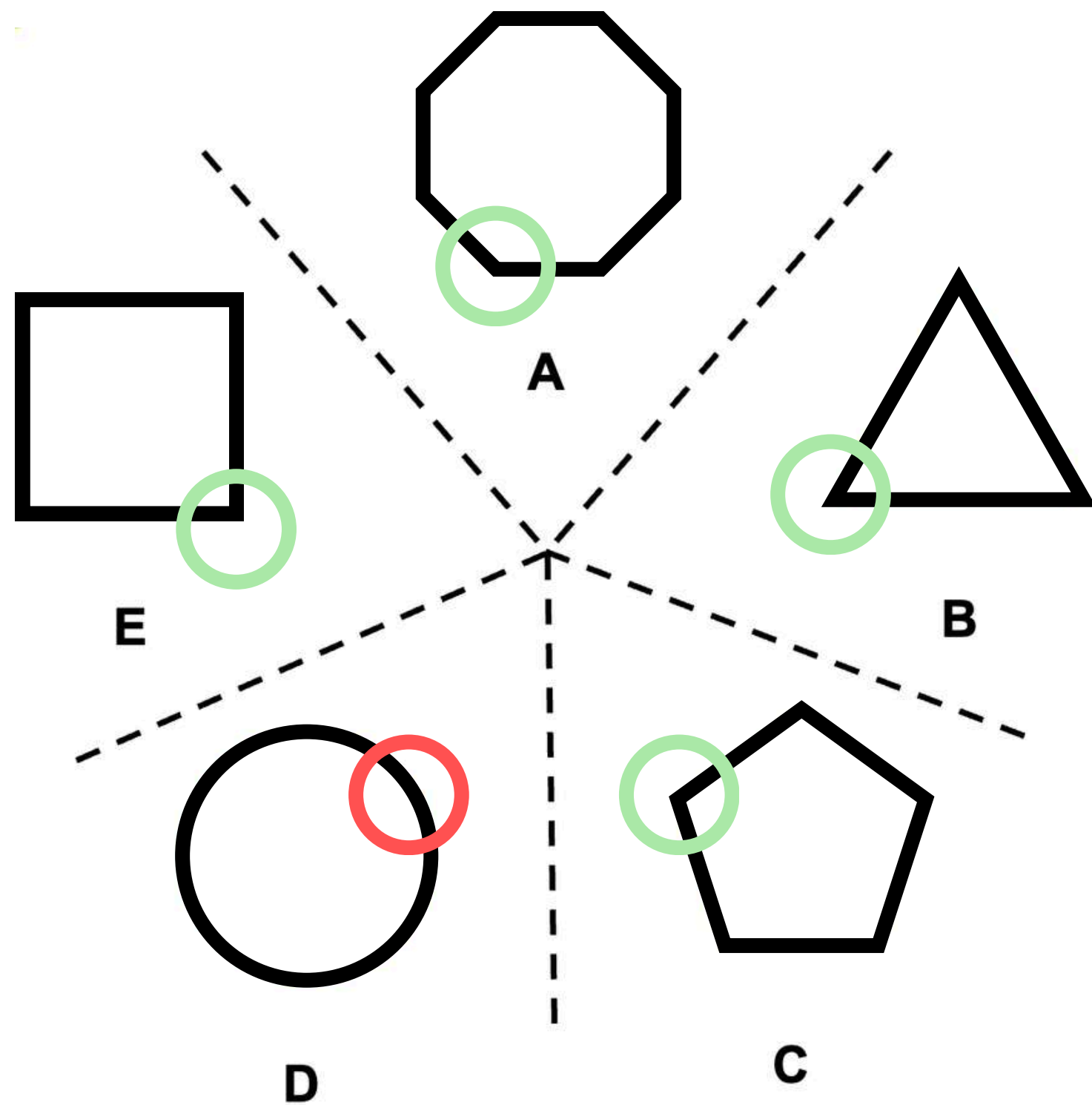
ปัญหา: เรียงลำดับรูปภาพต่อไปนี้ ยังไงดี?



2. การคิดเป็นนามธรรม (Abstraction)

- คือ การคัดแยกคุณลักษณะที่สำคัญออกจากรายละเอียดปลีกย่อย เพื่อให้ได้ข้อมูลที่จำเป็นและเพียงพอในการแก้ปัญหา
- ลองทำจัดสิ่งที่ไม่จำเป็นออกไป
- ลดสิ่งลวงตา มองเห็นสิ่งที่จำเป็นจริงๆ

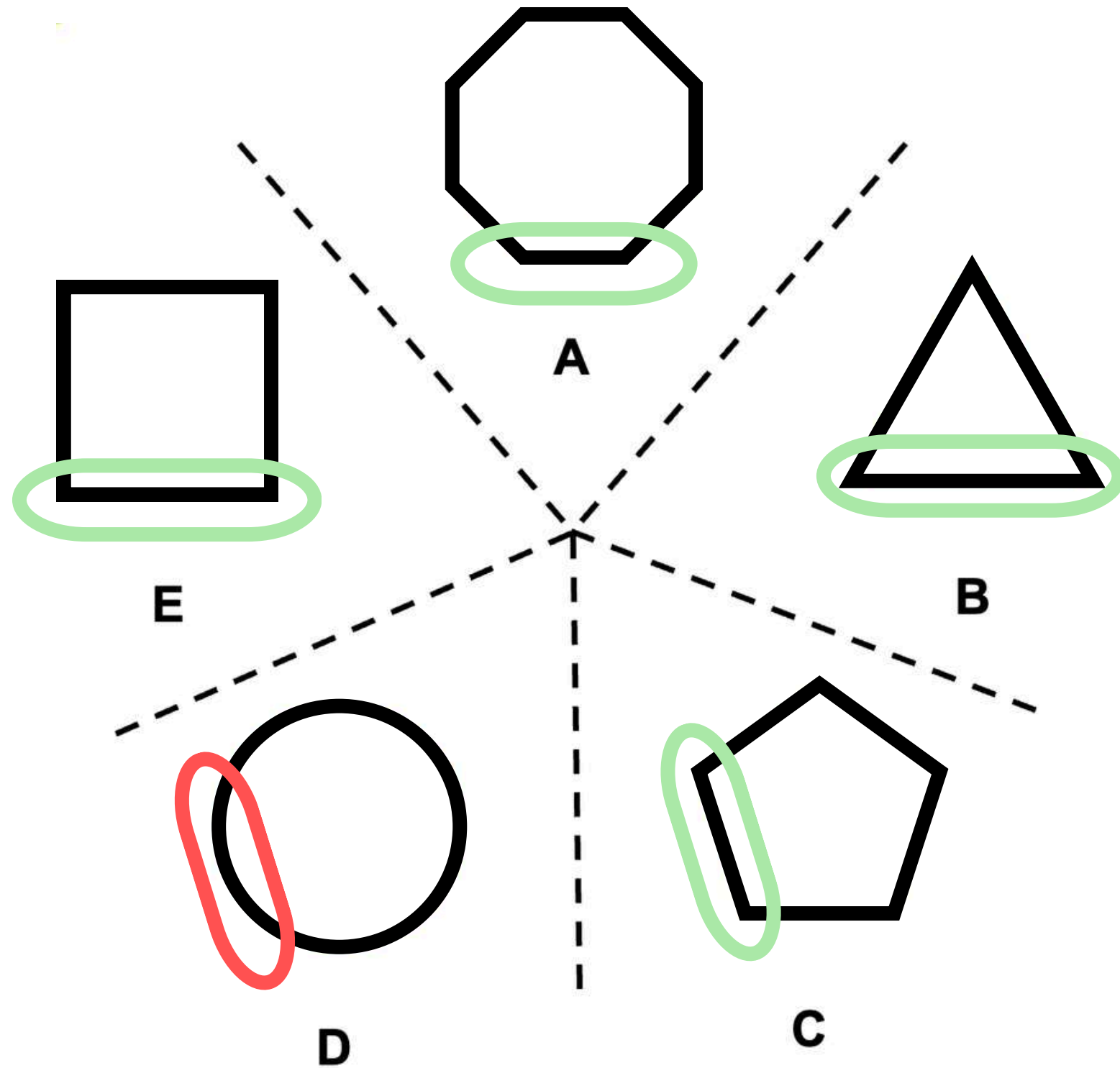
ปัญหา: เรียงลำดับรูปภาพต่อไปนี้ ยังไงดี?



3. การรู้จำรูปแบบ (Pattern Recognition)

- คือ การวิเคราะห์และแสดงความสัมพันธ์ของสิ่งต่างๆ ที่มีลักษณะสำคัญร่วมกันอย่างมีเงื่อนไข
- สังเกตสิ่งที่ปรากฏชัดเจน และ ช้อนเร้น
- สังเกตสิ่งที่ปรากฏ มีอะไรคล้ายกัน
- สังเกตสิ่งที่ปรากฏ มีอะไรต่างกัน
- เลือกรูปแบบที่สังเกตได้และมีความเป็นไปได้ที่จะใช้ในการแก้ปัญหา

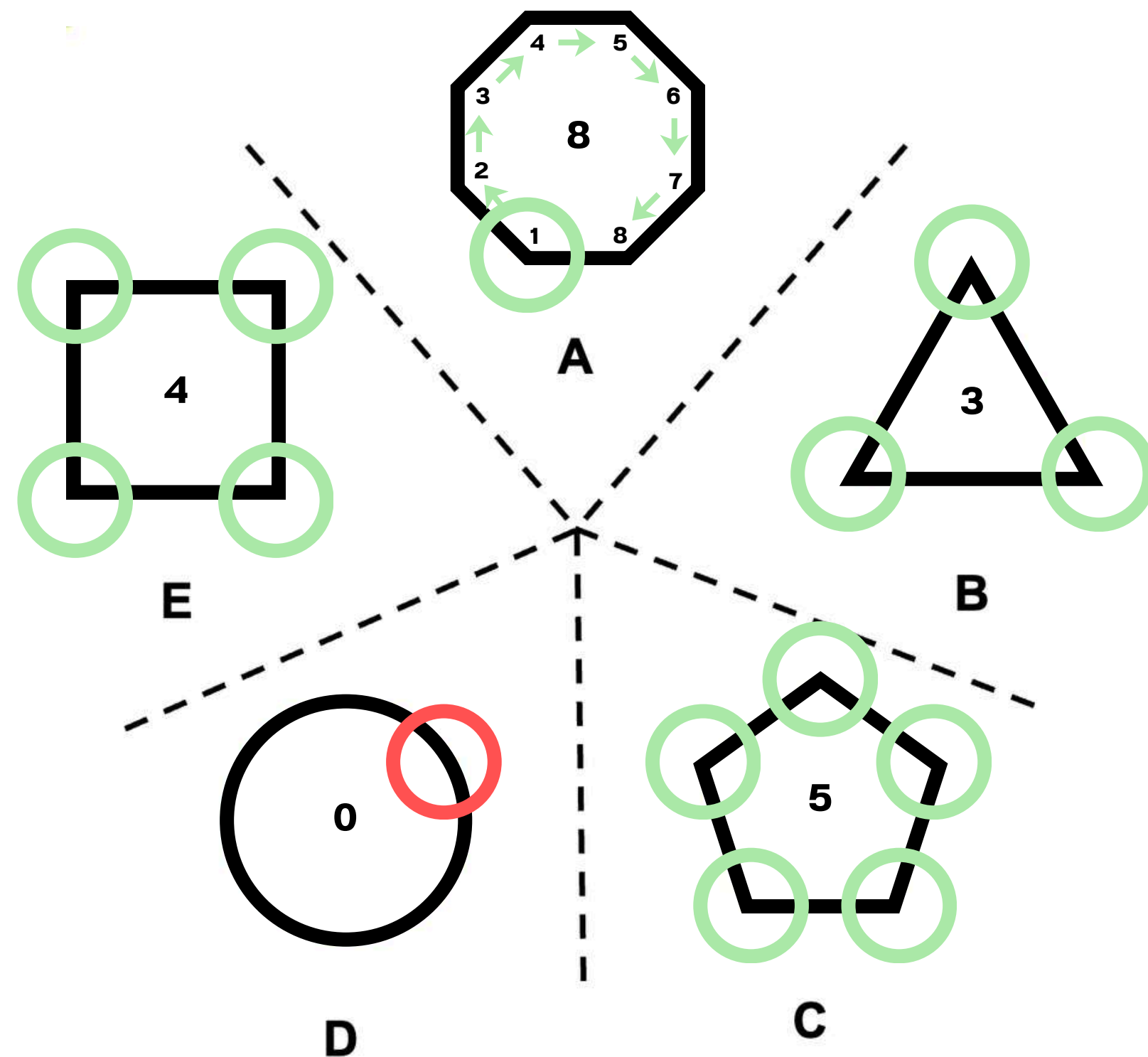
ปัญหา: เรียงลำดับรูปภาพต่อไปนี้ ยังไงดี?



3. การรู้จำรูปแบบ (Pattern Recognition)

- คือ การวิเคราะห์และแสดงความสัมพันธ์ของสิ่งต่างๆ ที่มีลักษณะสำคัญร่วมกันอย่างมีเงื่อนไข
- สังเกตสิ่งที่ปรากฏชัดเจน และ ช้อนเร้น
- สังเกตสิ่งที่ปรากฏ มีอะไรคล้ายกัน
- สังเกตสิ่งที่ปรากฏ มีอะไรต่างกัน
- เลือกรูปแบบที่สังเกตได้และมีความเป็นไปได้ที่จะใช้ในการแก้ปัญหา

ปัญหา: เรียงลำดับรูปภาพต่อไปนี้ ยังไงดี?



4. การออกแบบอัลกอริทึม (Algorithm Design)

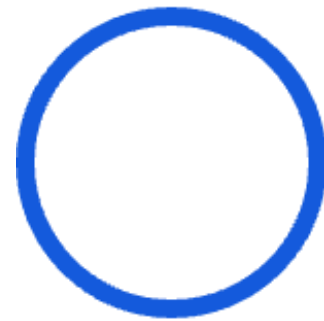
- เริ่มต้นขั้นตอน ขั้นตอนถัดไป และสิ้นสุดที่ขั้นตอนใด
- ลองคิดออกแบบหลายๆ ขั้นตอน และเปรียบเทียบเพื่อหาขั้นตอนที่เหมาะสมที่สุด

ปัญหา: เรียงลำดับรูปภาพต่อไปนี้ ยังไงดี?

การออกแบบอัลกอริทึม (Algorithm Design)

- เริ่มต้นขั้นตอน ขั้นตอนถัดไป และสิ้นสุดที่ขั้นตอนใด
- ลองคิดออกแบบหลายๆ กระบวนการ และเปรียบเทียบเพื่อหาขั้นตอนที่เหมาะสมที่สุด

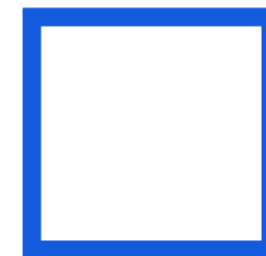
เรียงลำดับรูปภาพด้วย
อัลกอริทึมจากน้อยไปหามาก
(Ascending Order)



D



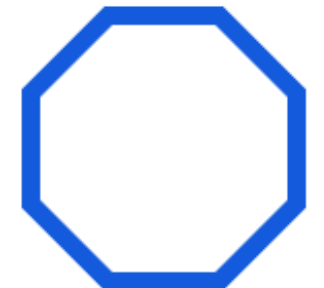
B



E



C



A

เรียงลำดับรูปภาพด้วย
อัลกอริทึมจากมากไปหาน้อย
(Descending Order)



A



C



E



B



D

โค้ดถึง Coding คืออะไร?

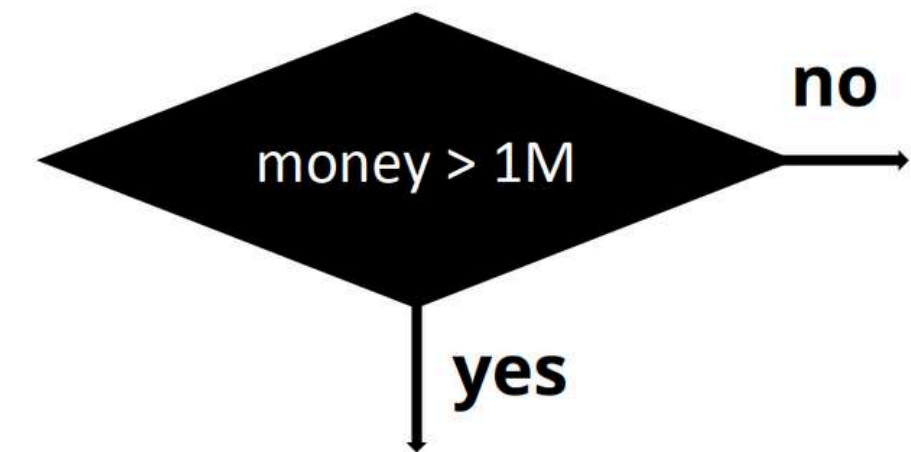
A

10011100110101

B

print("Hello World!")

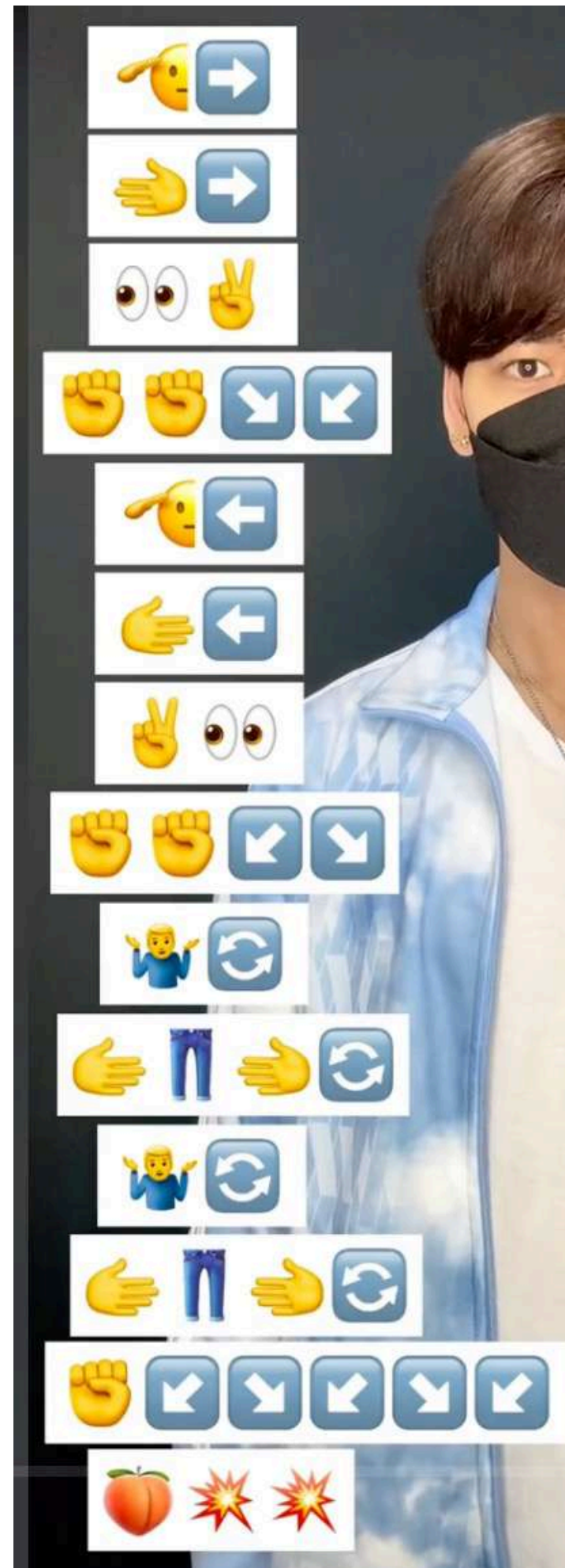
C



D



ทำเต้นเพลงอะไร?



ทำเต้นเพลงอะไร?



on TikTok

Watch n

Watch now

@xxtakaharuxx

#มองนานๆ #มองนานๆchallenge

โค้ดดิ้ง (Coding)

- Code คือ ระบบของคำ อักษร หรือ สัญลักษณ์ ในการแทนข้อความในรูปแบบหนึ่งๆ หรืออาจเป็นระบบจำนวน/ตัวเลข อักษร หรือสัญลักษณ์ที่ใช้แทนอะไรบางอย่างในลักษณะที่สั้นกระชับ หรือ อยู่ในรูปแบบที่เข้าใจได้สะดวกขึ้น
- ดังนั้น พวก รหัส, เลวฐาน, สูตร, Pseudocode, Flowchart, Diagram, Computer Programming ต่าง ๆ นับว่าเป็น Code ในลักษณะหนึ่ง เพราะอยู่ในระบบของคำ สัญลักษณ์ที่สั้นกระชับ ชัดเจน สะดวกต่อการทำความเข้าใจ และการแปลงไปสู่หน่วยอื่น ๆ ได้

dictionary.cambridge.org

Cambridge Dictionary

code

noun • **UK**  /kəʊd/ **US**  /koʊd/

code *noun* (LANGUAGE)

★ **B2** [C or U] **a system of words, letters, or signs used to represent a message in secret form, or a system of numbers, letters, or signals used to represent something in a shorter or more convenient form:**

*The message was written **in** code.*

*She managed to **decipher/break/crack** (= succeed in understanding) the code.*

Each entry in this dictionary has a grammar code.

Coding - Debug

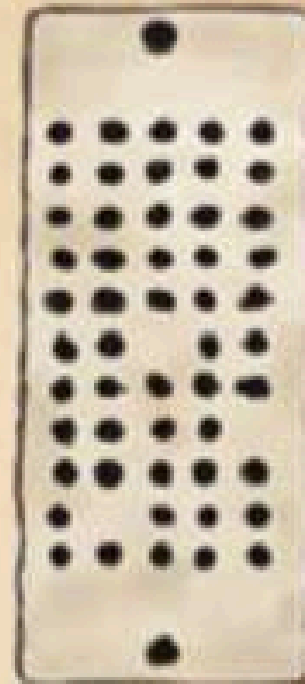
ADA LOVELACE FIRST COMPUTER PROGRAMMER



The Analytical Engine

Lovelace's program turned a complex formula into simple calculations that could be encoded on punched cards and fed into Charles Babbage's Analytical Engine, a mechanical computer that he designed but never built. She published it in 1843, a century before the modern computer age.

"I went to put in something about Bernoulli's Number, in one of my Notes, as an example of how an explicit function may be worked out by the engine, without having been worked out by human head and hands first."



$$\frac{x}{e^x - 1} = \frac{1}{1 + \frac{x}{2} + \frac{x^2}{2 \cdot 3} + \frac{x^3}{2 \cdot 3 \cdot 4} + \text{etc.}}$$



A Universal Computer

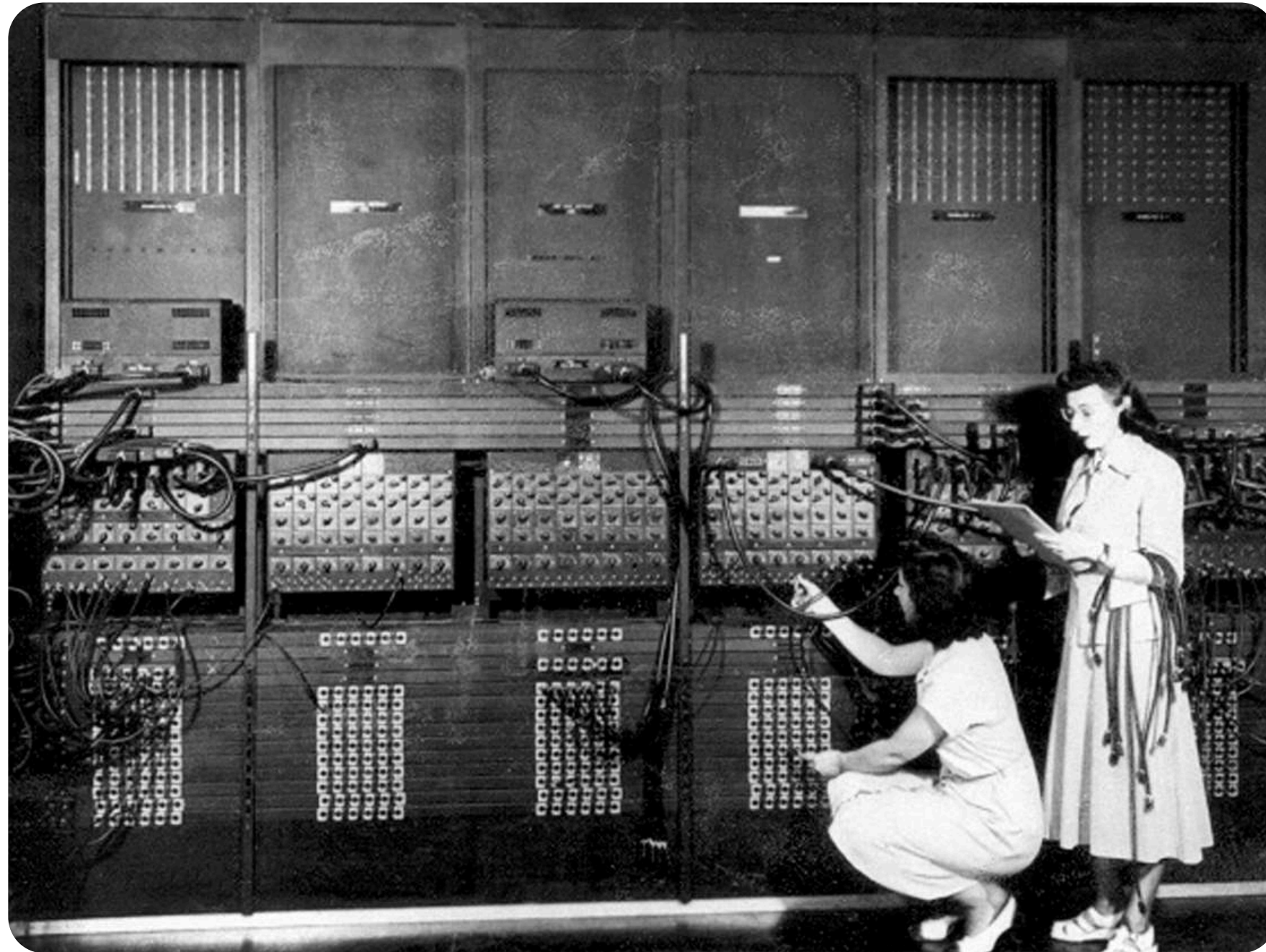
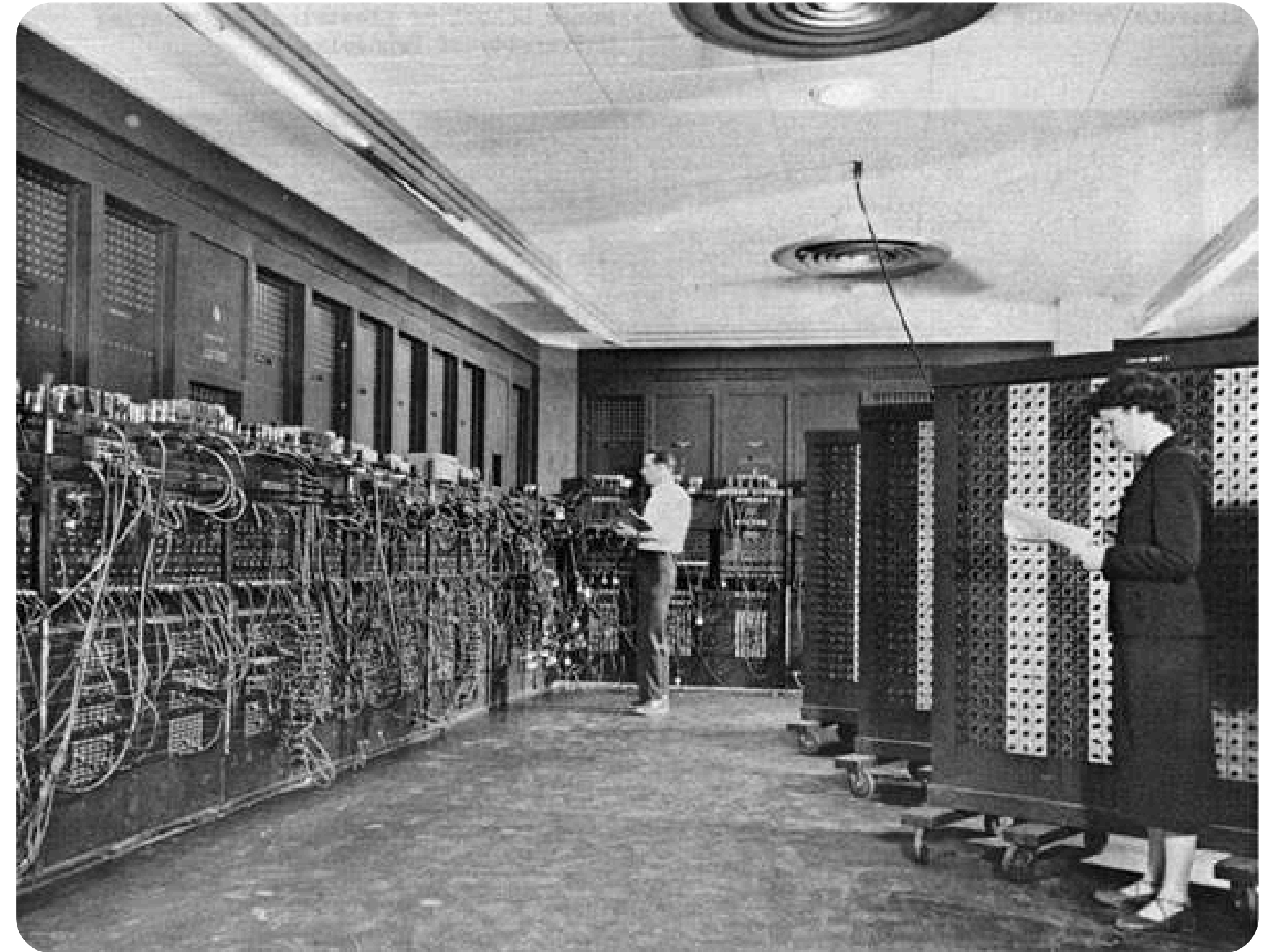
Lovelace did more than write the first computer program. She was also the first person to realize that a general purpose computer could do anything, given the right data and instructions.

"The Analytical Engine weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves."

"Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent."

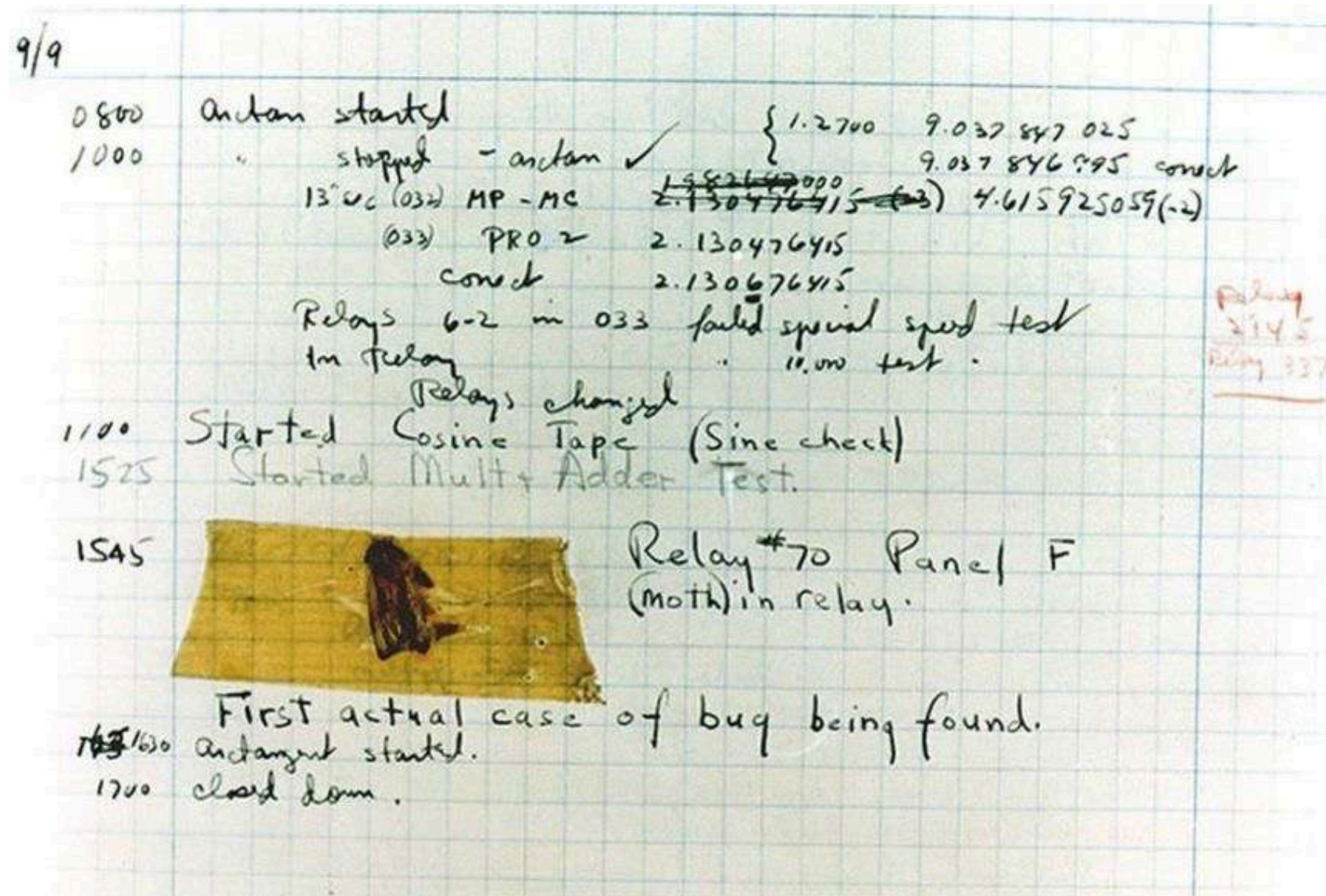


Coding - Debug





**On the 9th of September of 1946,
a Harvard technical team was asked
to take a look at a malfunction that was
happening with the Mark II calculator.**



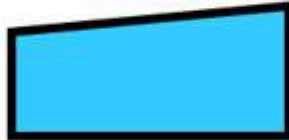
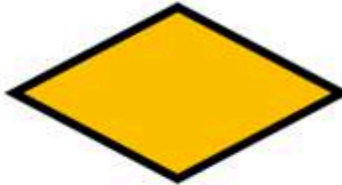
Grace Hopper, an American computer scientist and United States Navy rear admiral working on the Mark II calculator project at the time, added the caption “First actual case of bug being found,” and that’s the first time anyone used the word bug to describe a computer glitch. Naturally, the term debugging followed.

Grace Brewster Hopper



โค้ดดิ้ง (Coding)

รหัสจำลองสำหรับการเดินทาง	
รหัสจำลอง	ความหมาย
➡	เดินไปข้างหน้า
⬆	เดินขึ้นข้างบน
⬇	เดินลงไปข้างล่าง
⬅	เดินไปทางซ้าย
↗	เดินเฉียงขวา
↖	เดินเฉียงซ้าย
↗	เดินเฉียงขึ้นข้างบน
↘	เดินเฉียงลงข้างล่าง

สัญลักษณ์	ความหมาย	ภาษาอังกฤษ
	หมายถึง จุดเริ่มต้น หรือ จุดสิ้นสุดการทำงาน	Begin และ End
	หมายถึง ทิศทางการไหลของงาน	Flow line
	หมายถึง การดำเนินการ หรือ การประมวลผล	Process
	หมายถึง การนำเข้าข้อมูลด้วย การป้อนค่าทางแป้นพิมพ์	Manual input
	หมายถึง การตัดสินใจตาม เงื่อนไขที่กำหนดไว้	Decision
	หมายถึง การแสดงผลบนหน้าจอ	Display

โค้ดดิ้ง (Coding)

รหัสจำลองของชุดคำสั่งที่ 1 : ลากเส้นจากจุด

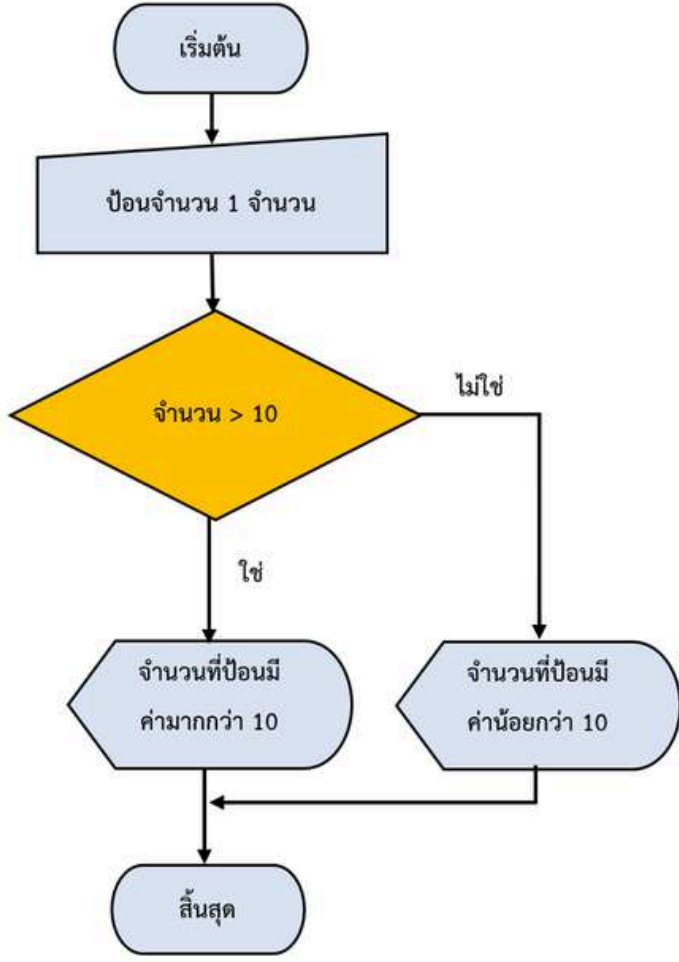
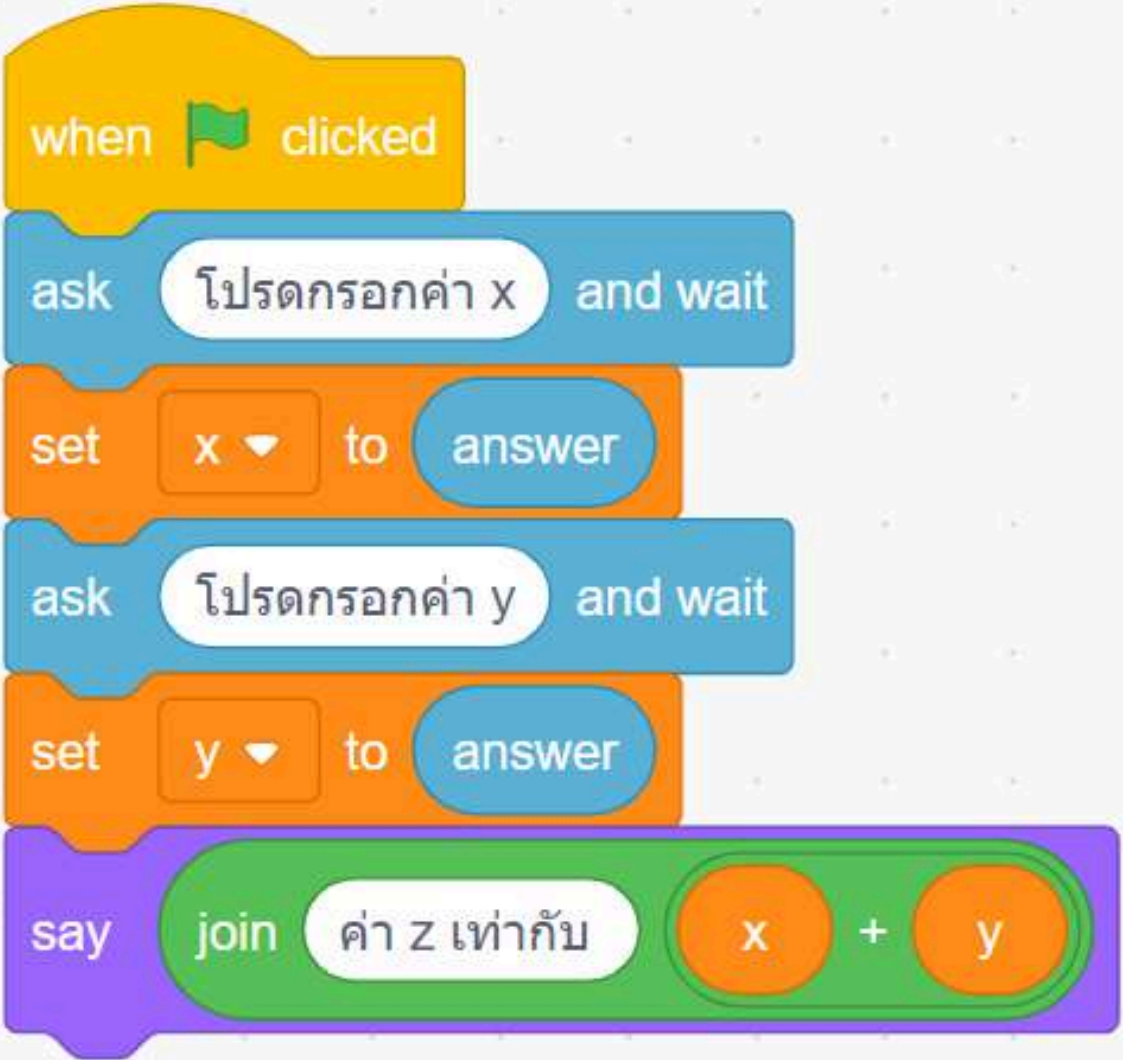
(C,2) → (A,4) → (A,7) → (B,7) → (B,5) → (C,5) → (C,9) → (D,9) → (D,7) → (G,7) → (G,9) → (H,9) → (H,6) → (I,7) → (I,6) → (H,5) → (H,2) → (C,2)

รหัสจำลองของชุดคำสั่งที่ 2 : ลากเส้นจากจุด

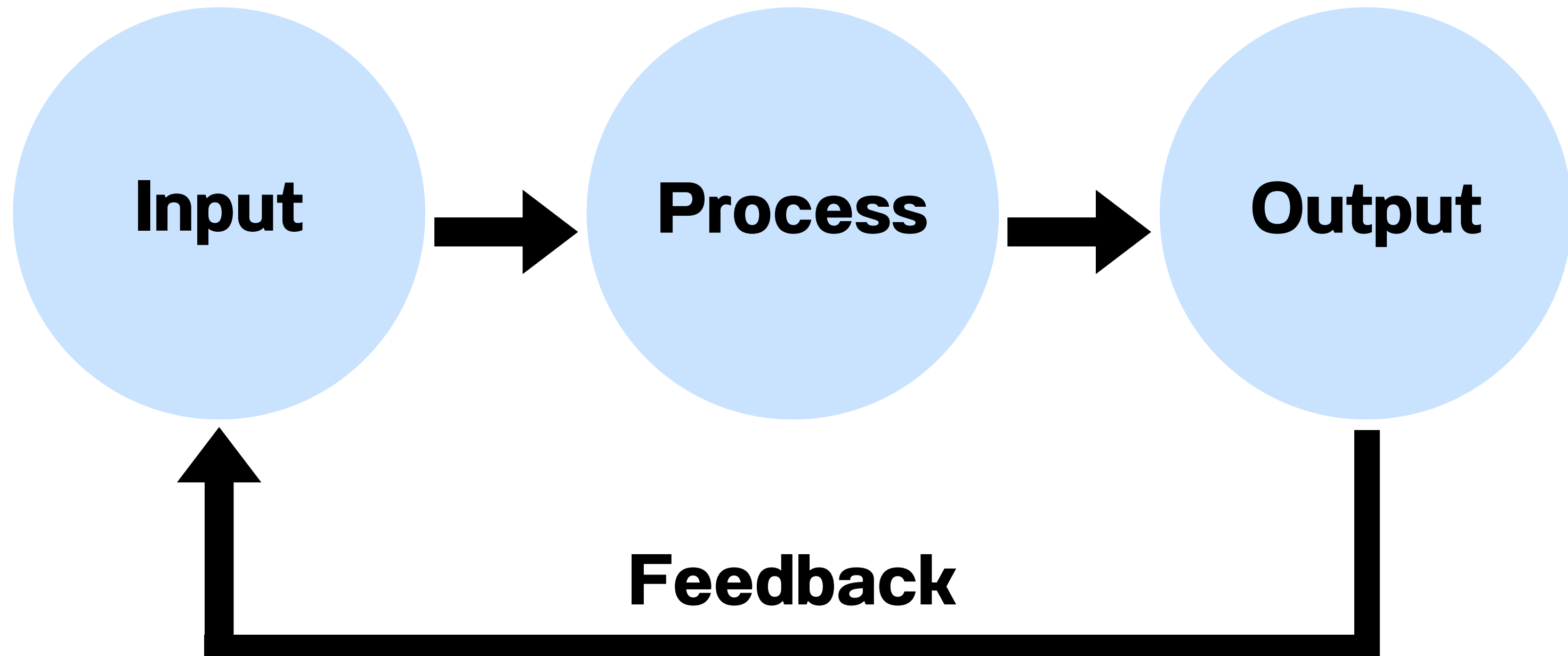
(D,5) → (E,6) → (F,6) → (F,3) → (E,3) → (D,4)

แสดงผล: ?

วัตถุประสงค์: ป้อนจำนวนด้วยแป้นพิมพ์และอยากรู้ว่าจำนวนที่ป้อนเข้ามามากกว่า 10 หรือไม่



System Theory



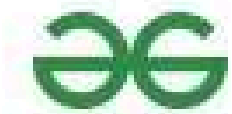
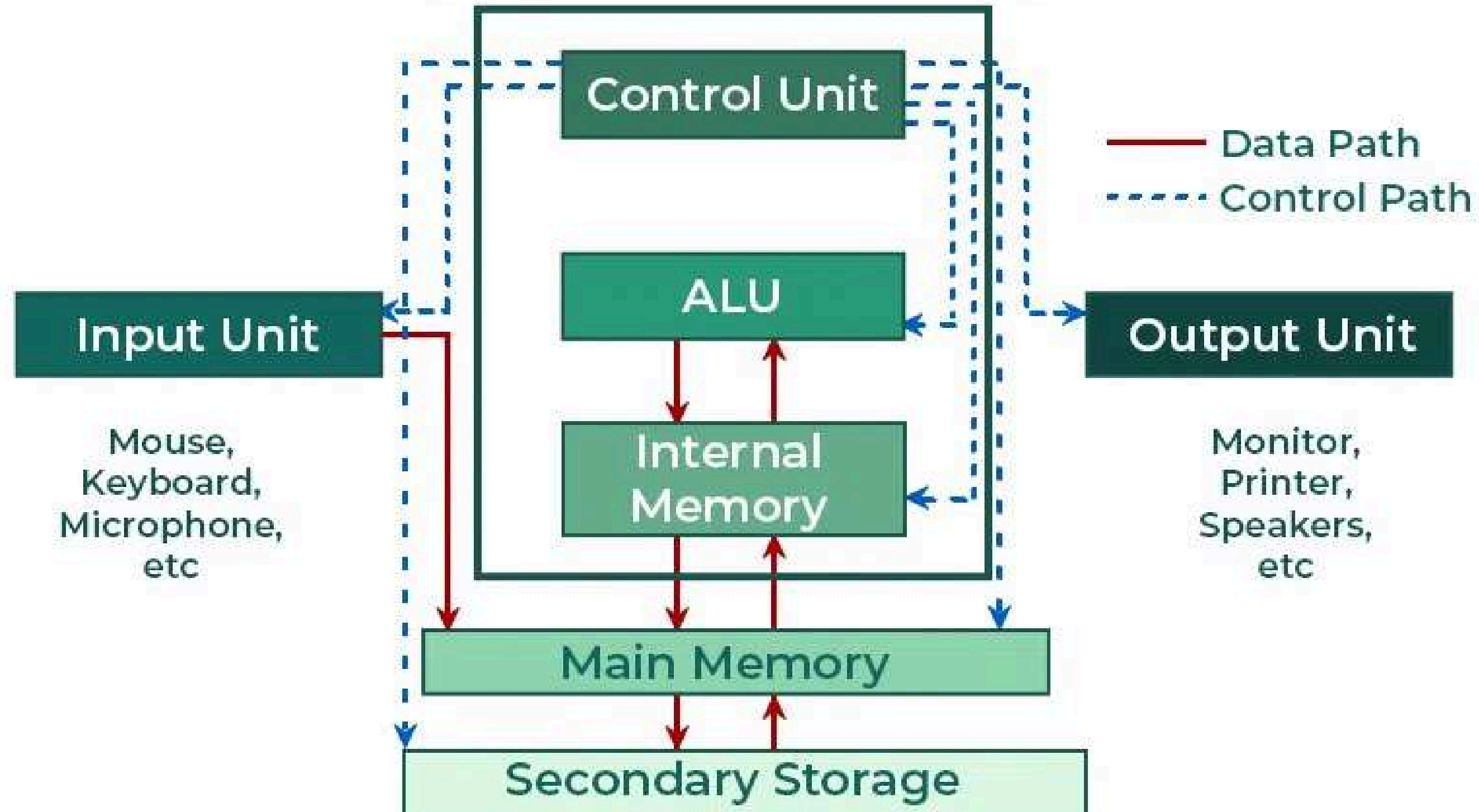
Computer System

Computer

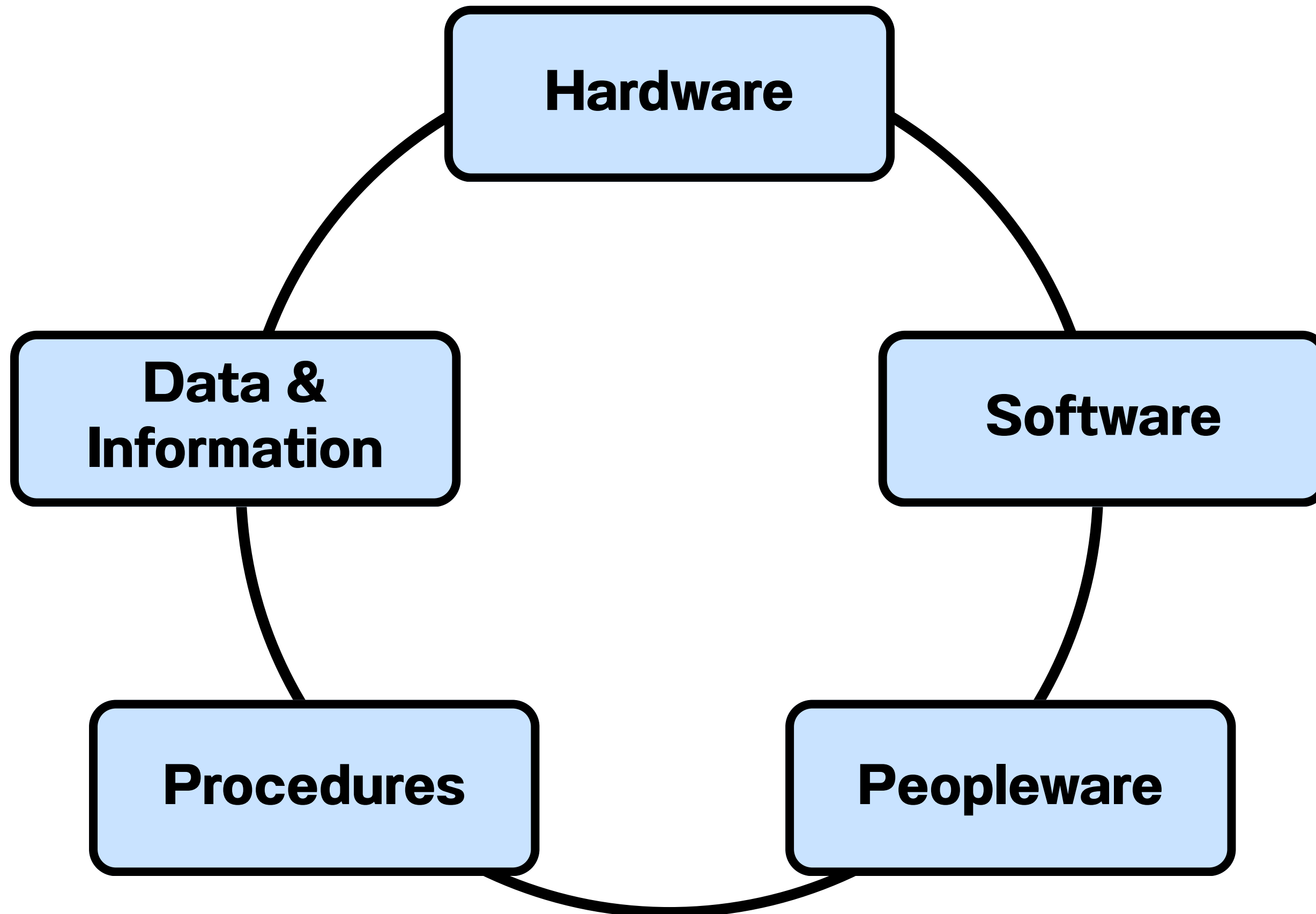


Computer Components

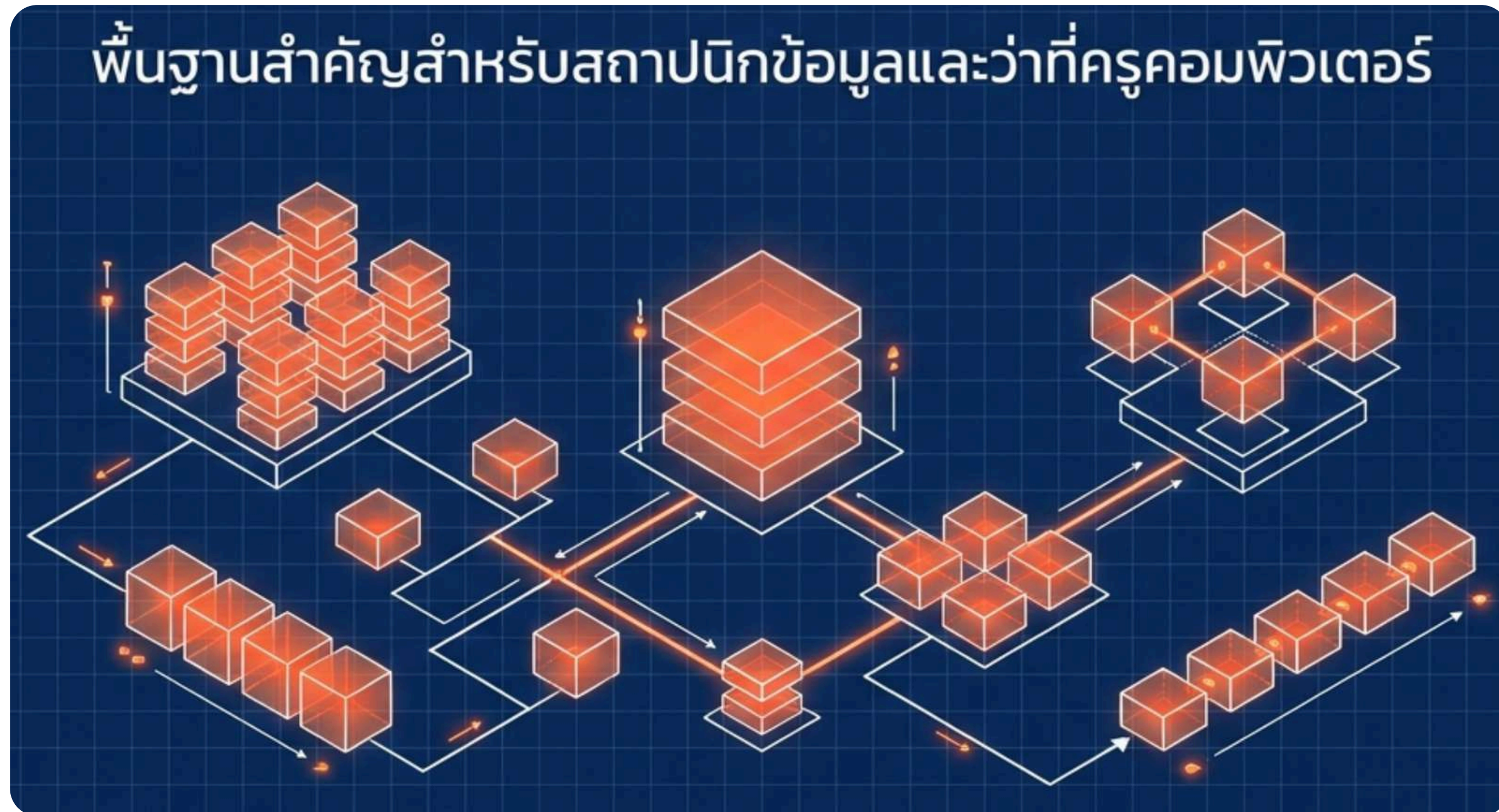
Components of computer



Computer Components

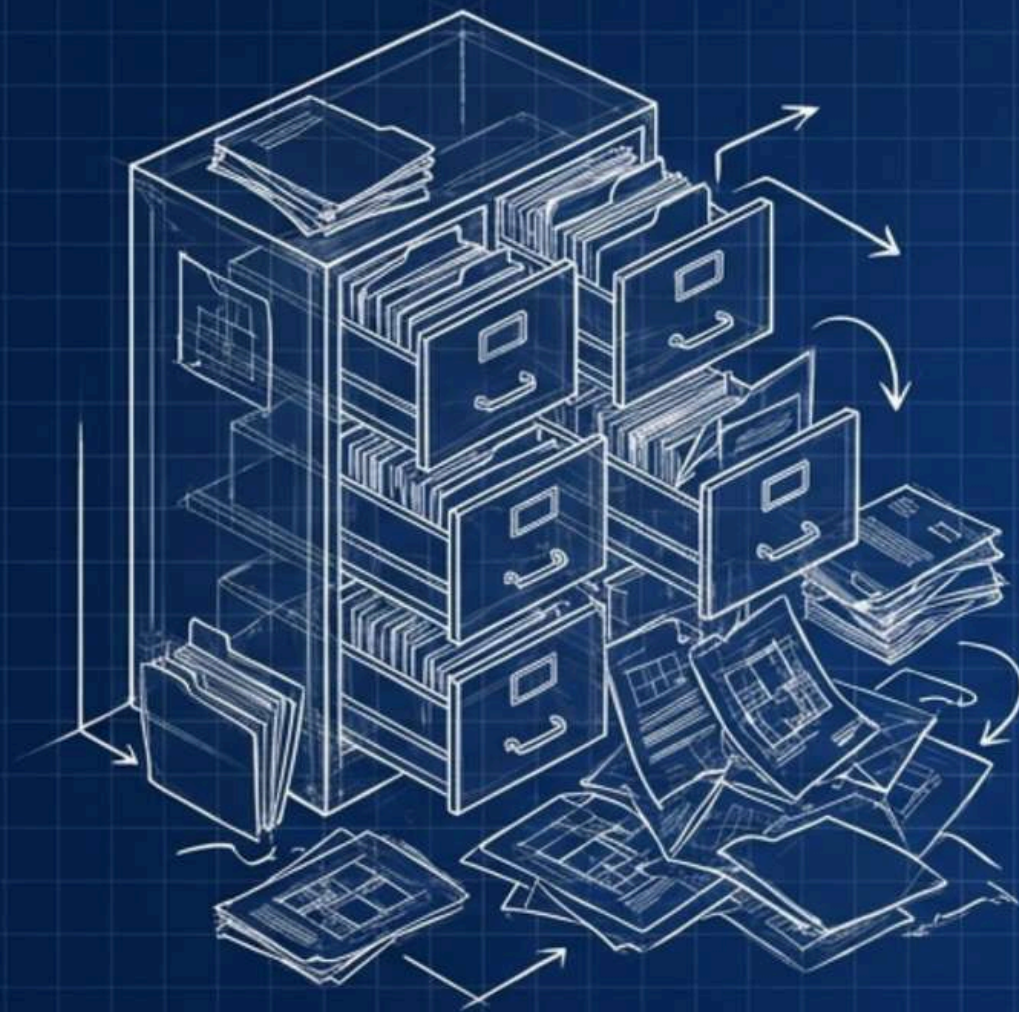


ความหมายและความสำคัญของโครงสร้างข้อมูล (Data Structures)



ความหมายและความสำคัญของโครงสร้างข้อมูล (Data Structures)

จากความยุ่งเหยิงในโลกจริง สู่ความเป็นระเบียบในโลกดิจิทัล



Past - Physical World

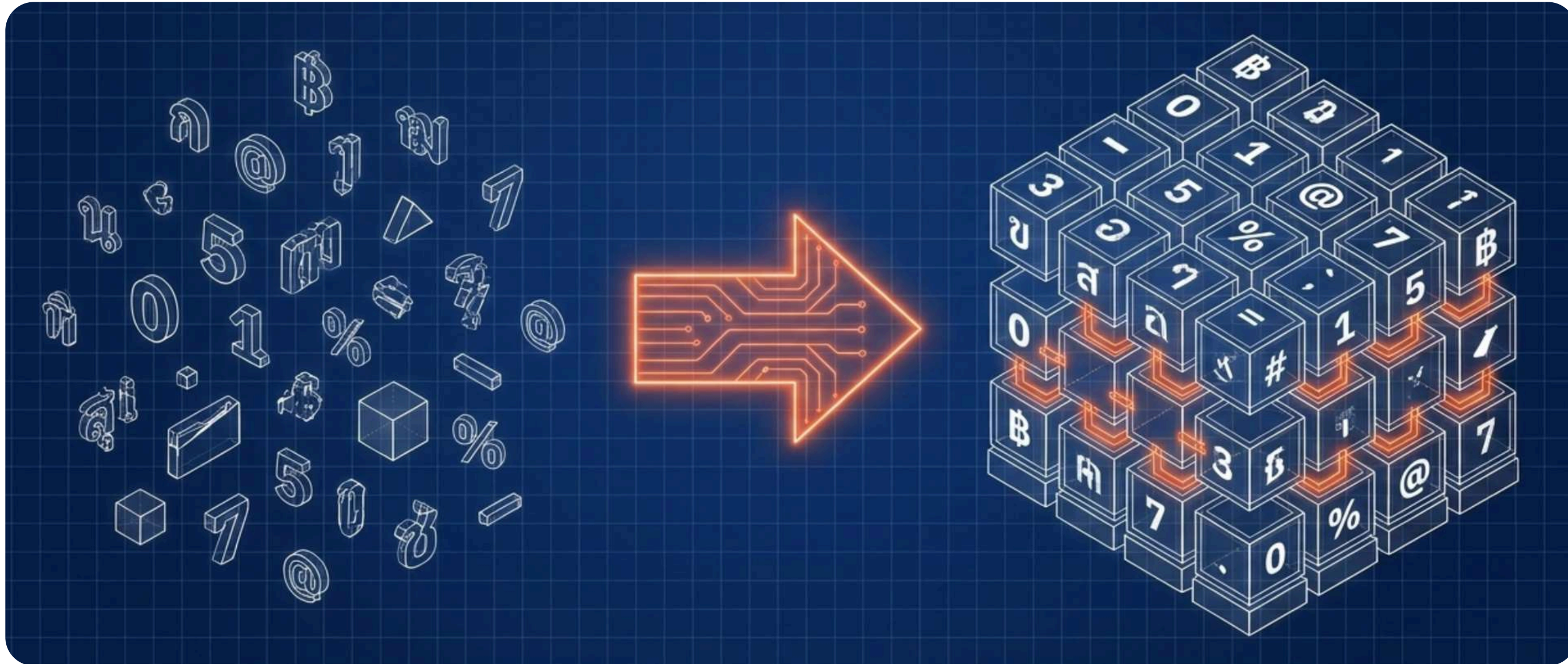
การจัดเก็บแบบแฟ้มเอกสารหรือกระดาษ ประสบปัญหา
เมื่อข้อมูลมีจำนวนมาก การค้นหาทำได้ล่าช้าและไม่สะดวก



Present - Digital World

คอมพิวเตอร์รับข้อมูล (Input) นำไปประมวลผล (Process)
เพื่อสร้างผลลัพธ์ (Output) อย่างรวดเร็ว แต่คอมพิวเตอร์
จะทำได้ดีต้องอาศัย “การจัดรูปแบบข้อมูล” ที่เหมาะสม

ความหมายและความสำคัญของโครงสร้างข้อมูล (Data Structures)



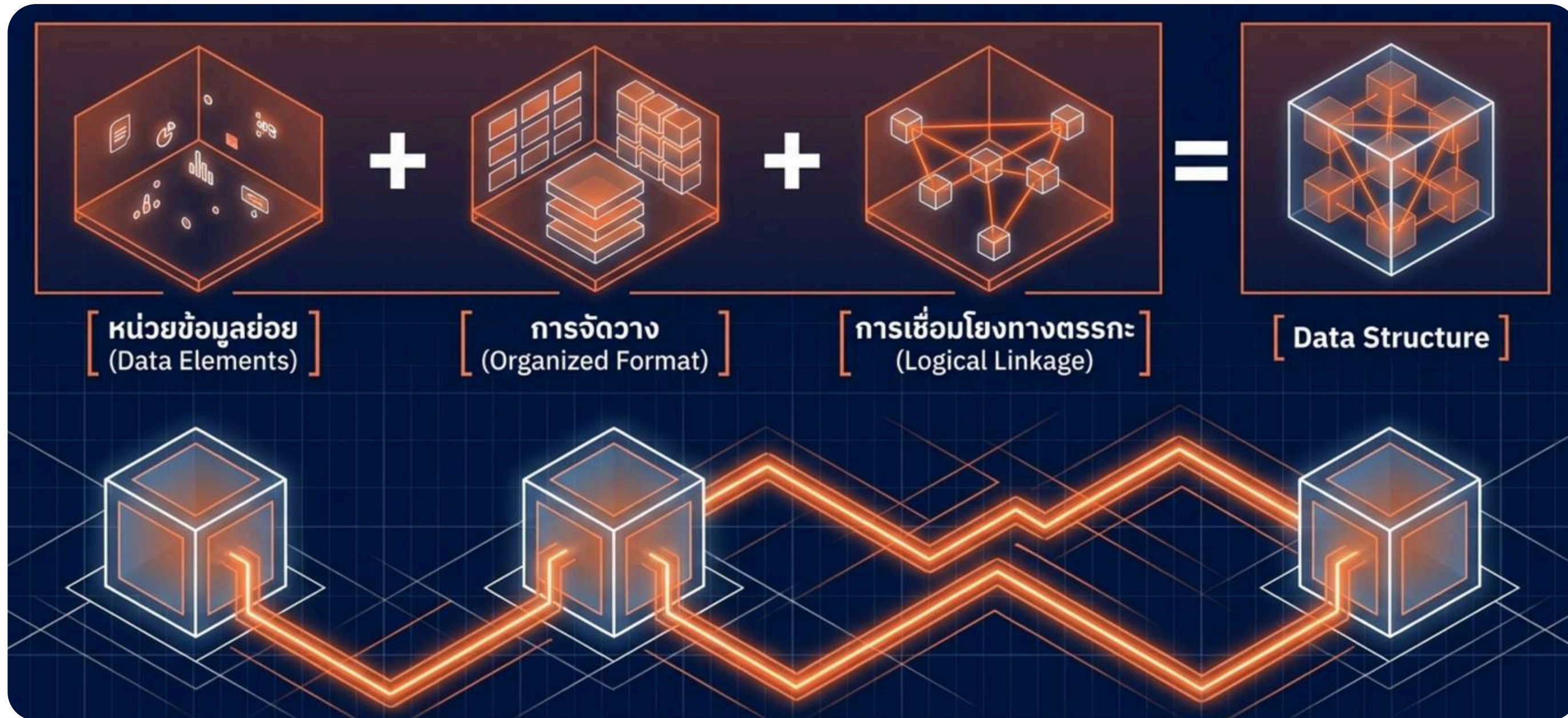
โครงสร้างข้อมูล (Data Structures) คืออะไร?

การจัดรูปแบบหรือการจัดเก็บข้อมูลในรูปแบบต่าง ๆ เพื่อให้ระบบคอมพิวเตอร์สามารถ “เข้าใจ” และ “จัดการ” กับกลุ่มข้อมูลที่จัดเก็บได้อย่างมีประสิทธิภาพ

ความหมายและความสำคัญของโครงสร้างข้อมูล (Data Structures)

- **โครงสร้าง** หมายถึง ส่วนประกอบที่สำคัญๆ ซึ่งนำมาเชื่อมเข้าด้วยกันให้เป็นรูปร่างเดียวกัน
- **ข้อมูล** หมายถึง ข้อเท็จจริง หรือสิ่งที่ถือหรือยอมรับว่าเป็นข้อเท็จจริง สำหรับใช้เป็นหลักฐานหาความจริงหรือการคำนวณ
- **โครงสร้างข้อมูล** หมายถึง ส่วนประกอบสำคัญของรูปแบบการจัดระบบข้อมูลแต่ละชนิด ในหน่วยความจำของระบบคอมพิวเตอร์ โดยมีการกำหนดนิยามและลักษณะ ความสัมพันธ์ที่เหมาะสมสำหรับการทำงานในโปรแกรม กล่าวเป็นหลักการได้ว่าจะต้องมี การบันทึกข้อมูลตามรูปแบบโครงสร้างข้อมูลในหน่วยความจำหลัก โดยมีการกำหนด ความสัมพันธ์ (Relationship) ระหว่างข้อมูล พร้อมทั้งกำหนดวิธีปฏิบัติที่สามารถใช้กับ ข้อมูลภายในโครงสร้างนั้น

นิยามเชิงวิชาการ: สมการของโครงสร้างข้อมูล



การเชื่อมโยงทางตรรกะ คือ ข้อมูลจะเชื่อมโยงกันด้วยความสัมพันธ์ ไม่ใช่ความใกล้ชิดทางกายภาพ

ทำไมต้องศึกษาโครงสร้างข้อมูล

ผู้ที่ต้องการพัฒนาโปรแกรมคอมพิวเตอร์เพื่อจัดการกับข้อมูล จำเป็นต้องเข้าใจ ลักษณะวิธีการจัดการข้อมูลตามโครงสร้างต่าง ๆ เพื่อประโยชน์ 3 ประการ:



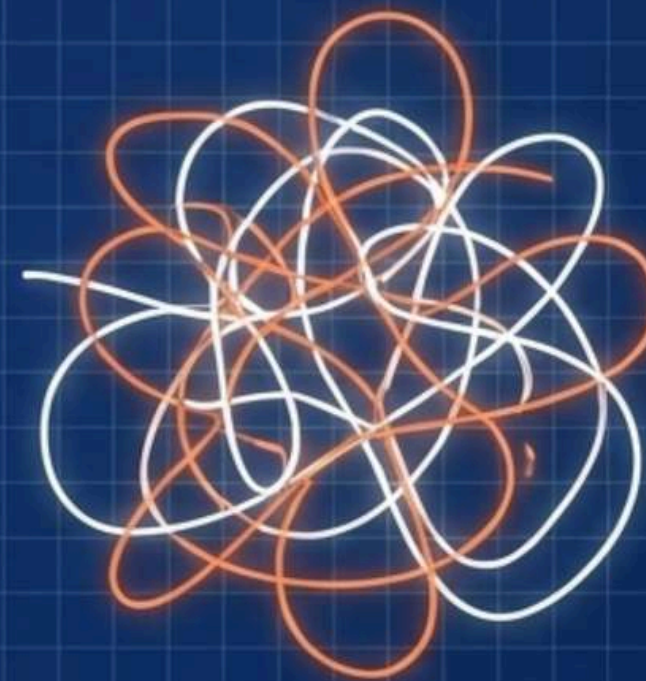
1. ทำงานถูกต้อง (Correctness):
ข้อมูลถูกจัดเก็บและเรียกใช้
อย่างแม่นยำ



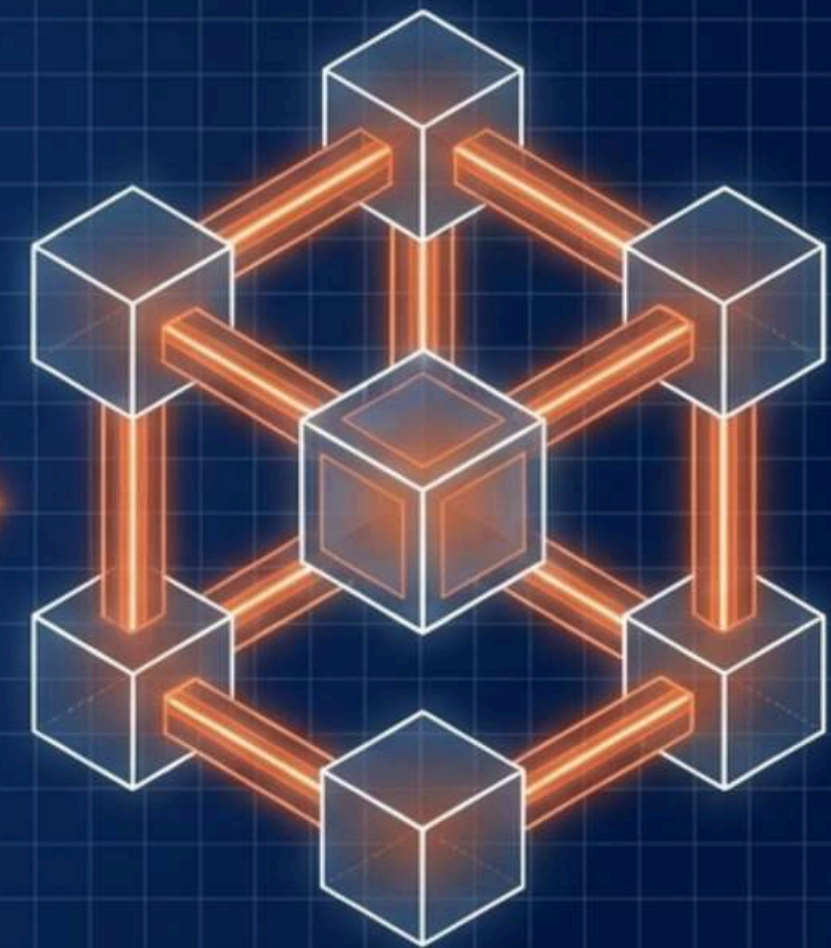
2. ตรวจสอบง่าย (Maintainability):
มีระเบียบชัดเจน ทำให้แก้ไข
ข้อผิดพลาด (Debug) ได้รวดเร็ว



3. มีประสิทธิภาพ (Efficiency):
จัดการทรัพยากรระบบได้ดีเยี่ยม



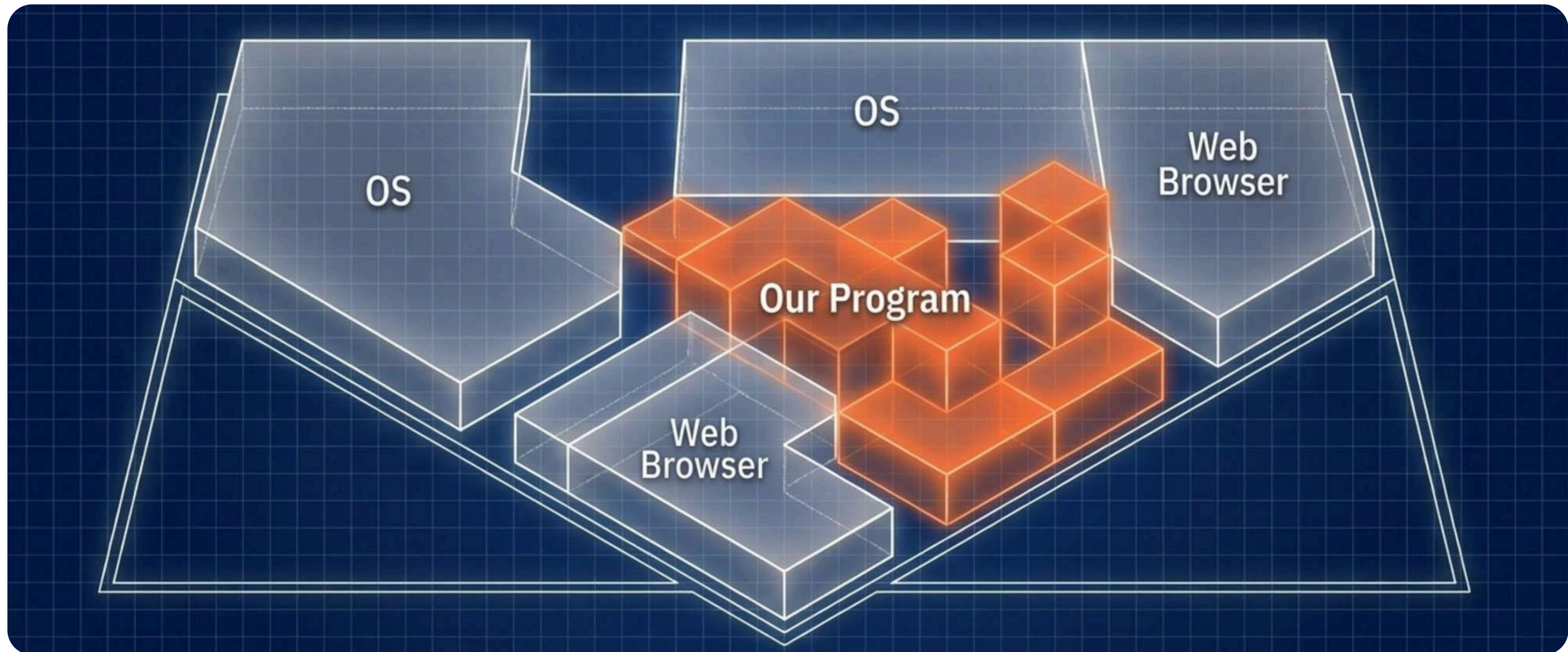
[Spaghetti Code]



[Structured Programming]

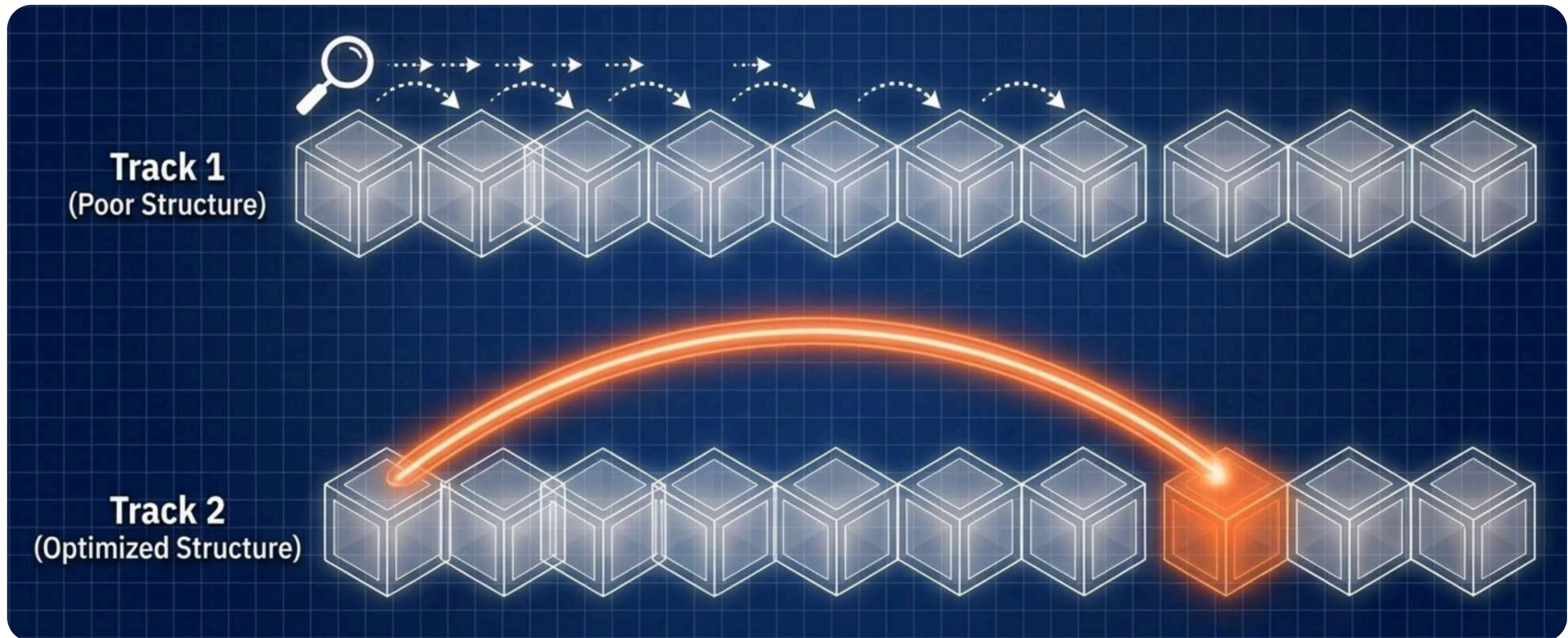
การจัดการหน่วยความจำ (Memory Management): สมรภูมิใน RAM

คอมพิวเตอร์อ่านข้อมูลในหน่วยความจำหลัก (RAM) ซึ่งมีพื้นที่จำกัดและต้องใช้งานร่วมกัน โครงสร้างข้อมูลที่ดีคือการ “แพ็คเกจ” ลงพื้นที่ที่เหลือให้คุ้มค่าที่สุด

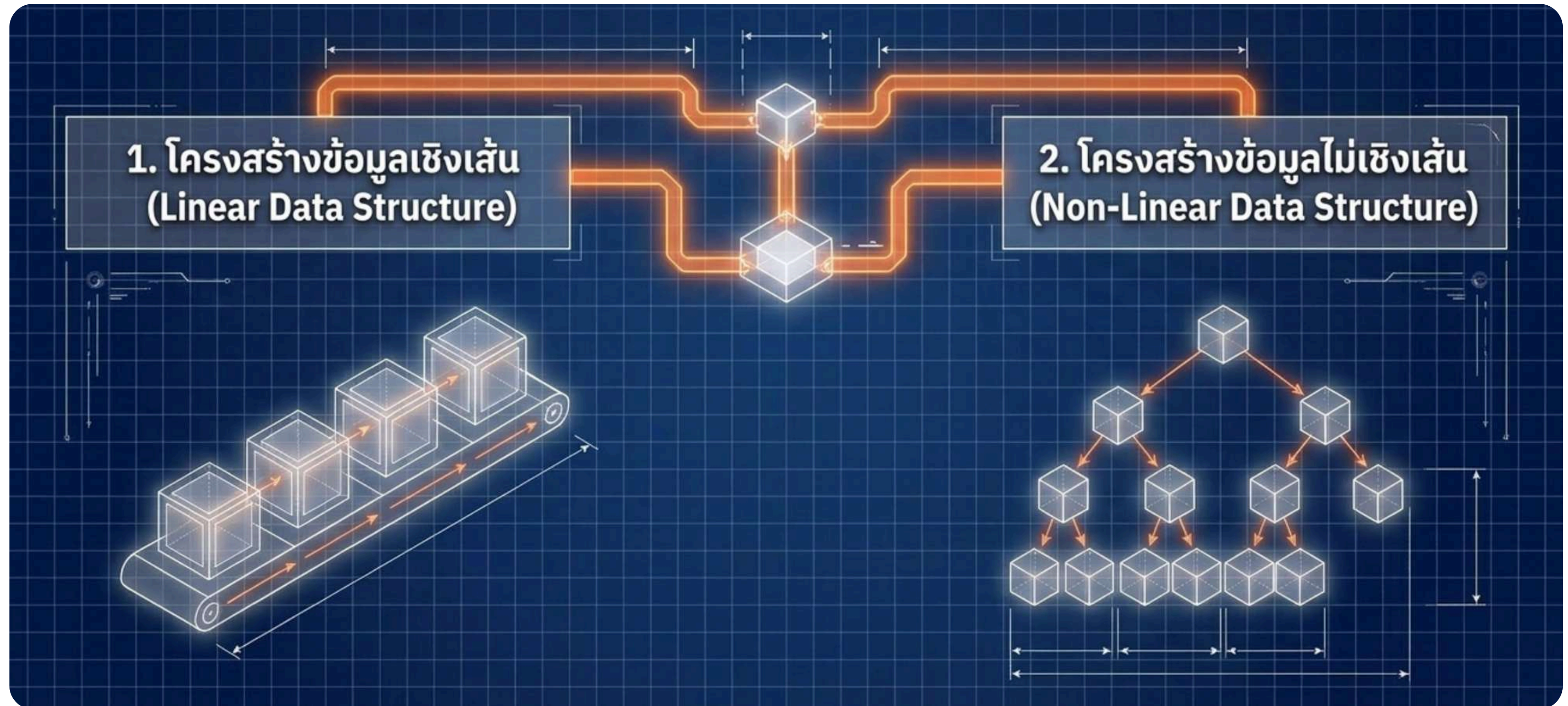


ความเร็วในการค้นหา (Search Efficiency)

นอกจากประหยัดพื้นที่แล้ว โครงสร้างข้อมูลที่ถูกต้องยังช่วยลด **“เวลา”** เมื่อเราต้องการดึงข้อมูลมาใช้งาน คอมพิวเตอร์จะไม่เสียเวลาสแกนพื้นที่ว่างเปล่าหรือข้อมูลที่ไม่เกี่ยวข้อง

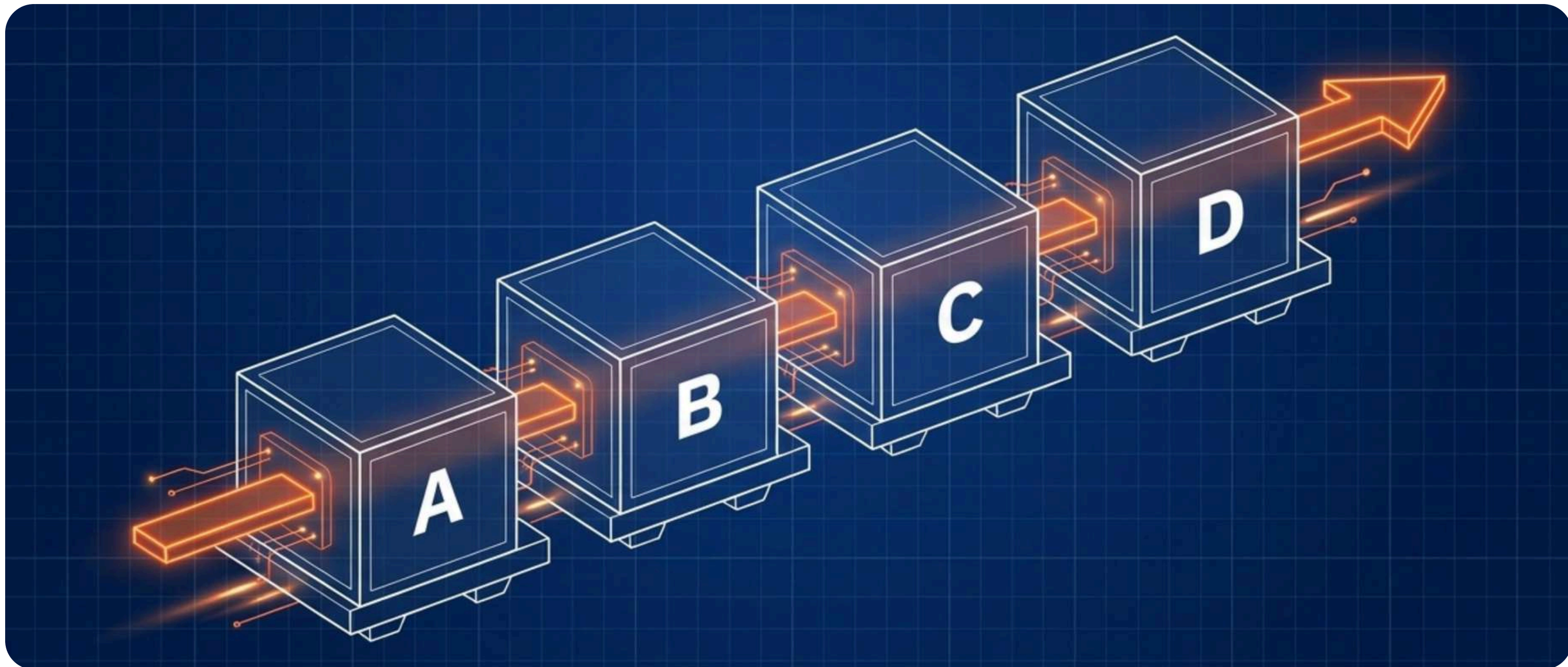


การแบ่งประเภทโครงสร้างข้อมูล (Data Structure Types)



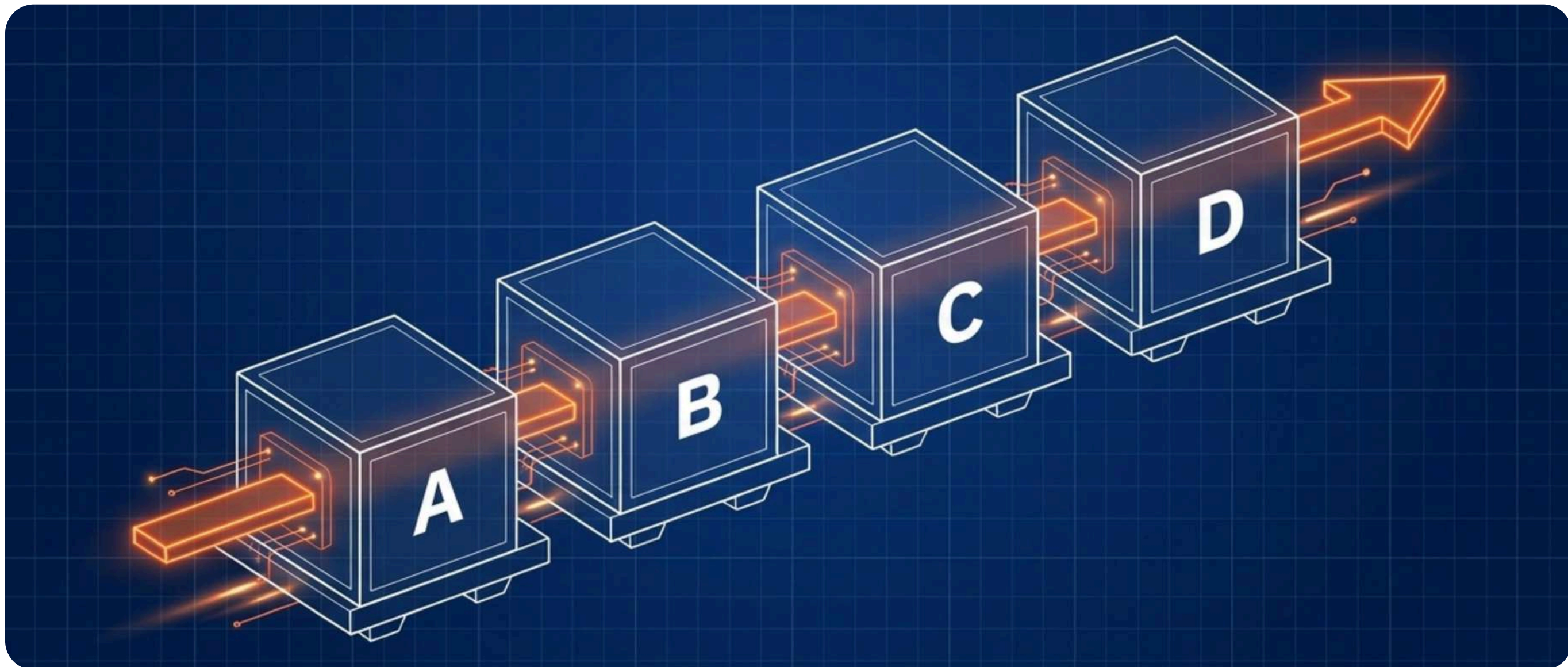
โครงสร้างข้อมูลเชิงเส้น (Linear Data Structure)

รูปแบบการจัดเก็บข้อมูลที่มีการจัดเรียงข้อมูลให้ต่อเนื่องกัน และเข้าถึงข้อมูลตามลำดับ



โครงสร้างข้อมูลเชิงเส้น (Linear Data Structure)

โครงสร้างข้อมูลเชิงเส้น (Linear Data Structures) คือ วิธีการจัดระเบียบข้อมูลในคอมพิวเตอร์ให้เรียงต่อกันเป็นเส้นตรงเหมือน "ลูกปัดที่ร้อยอยู่ในเส้นเชือก" ข้อมูลแต่ละตัวจะมีตำแหน่งที่แน่นอน มีตัวก่อนหน้า และมีตัวถัดไป เรียงลำดับกันไปเรื่อย ๆ อย่างเป็นระบบ ประกอบด้วย 4 แบบ ดังนี้



โครงสร้างข้อมูลเชิงเส้น (Linear Data Structure)

1. Array (อาร์เรย์) เปรียบเหมือน "ตู้ล็อกเกอร์นักเรียนที่มีเบอร์แปะไว้"

- หลักการ: เป็นการจองพื้นที่ขนาดใหญ่ทีเดียว ข้อมูลทุกตัวจะอยู่ติดกัน
- จุดเด่น: มี "เลขกล่อง (Index)" กำกับชัดเจน เช่น กล่องที่ 0, 1, 2, 3 ทำให้เราเดินไปหยิบของในกล่องที่ต้องการได้ทันทีโดยไม่ต้องไล่หาทีละกล่อง
- ข้อจำกัด: ขยายขนาดทีหลังไม่ได้ ถ้าจองตู้ไว้ 30 ช่อง แต่มีนักเรียนย้ายมาเพิ่มเป็น 31 คน ก็จะไม่ใส่ไม่ได้ ต้องซื้อตู้ใหม่ทั้งหมด

2. Linked-List (ลิงก์ลิสต์) เปรียบเหมือน "เกมเดินตามคำใบ้ / ลำสมบัติ"

- หลักการ: ข้อมูลไม่จำเป็นต้องอยู่ติดกันในหน่วยความจำ แต่ข้อมูลตัวที่ 1 จะมีกระดานไม้บอกว่า "ข้อมูลตัวที่ 2 อยู่ที่ไหน" และตัวที่ 2 ก็จะบอกทางไปตัวที่ 3 ต่อไปเรื่อย ๆ
- จุดเด่น: ยืดหยุ่นมาก ถ้ามีข้อมูลเพิ่มเข้ามาใหม่ ก็แค่เขียนคำใบ้ให้ชี้ไปหาตัวใหม่ได้ทันที ไม่ต้องกลัวตู้เต็มเหมือนอาร์เรย์
- ข้อจำกัด: ถ้าอยากได้ข้อมูลตัวที่ 10 เราหยิบทีเดียวไม่ได้ ต้องเสียเวลาเริ่มอ่านคำใบ้ตั้งแต่ตัวที่ 1 ไล่ไปเรื่อย ๆ จนถึงตัวที่ 10

โครงสร้างข้อมูลเชิงเส้น (Linear Data Structure)

3. Stack (สแต็ค) เปรียบเหมือน "ตั่งจานข้าวในโรงอาหาร"

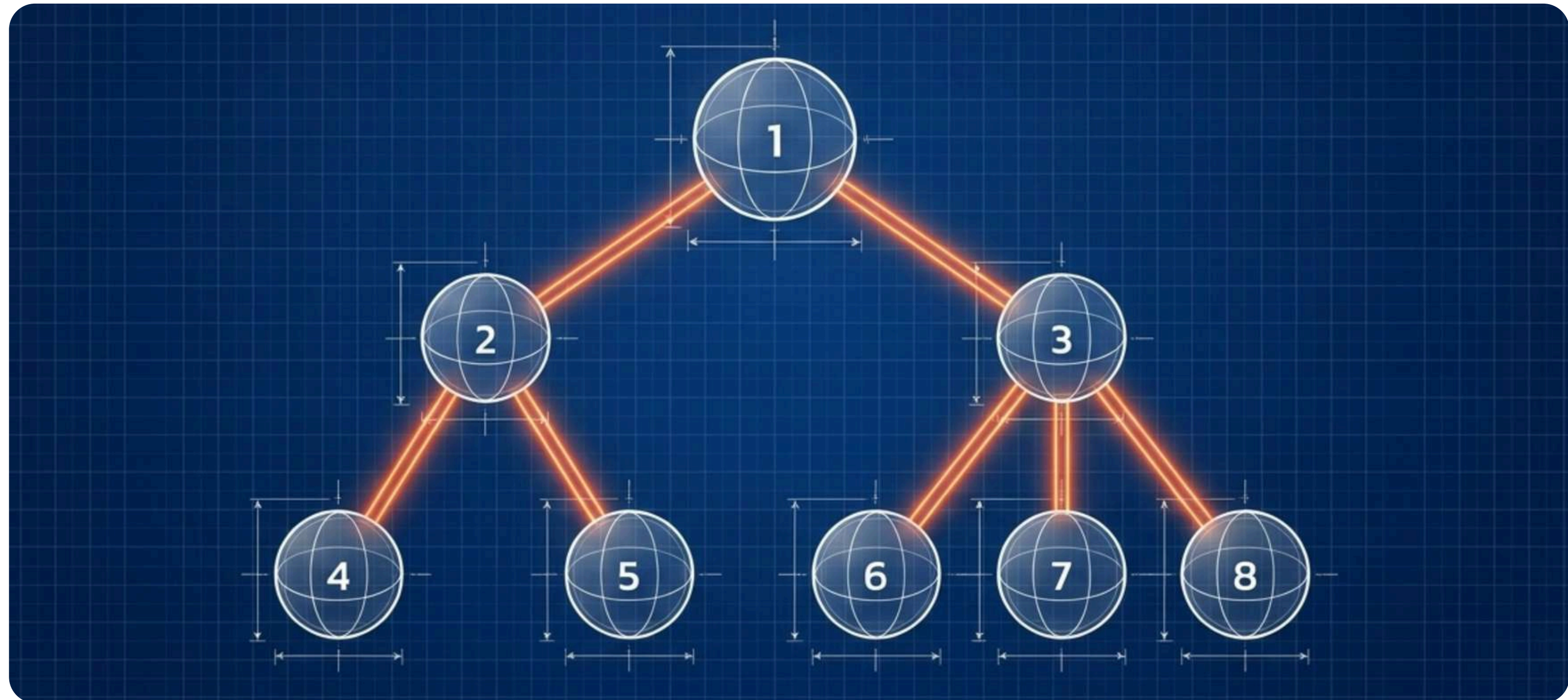
- หลักการ: ทำงานแบบ "เข้าทีหลัง แต่ได้ออกก่อน" (LIFO - Last In, First Out)
- จุดเด่น: เวลาคนล้างจานเสร็จ เขาจะวางจานซ้อนขึ้นไปข้างบนเรื่อย ๆ (Push) และเวลาแม่ครัวจะหยิบจานไปใช้ ก็ต้องหยิบใบที่อยู่บนสุดออกไปก่อนเสมอ (Pop)
- การใช้งาน: เหมือนปุ่ม "Undo (Ctrl+Z)" ในคอมพิวเตอร์ สิ่งที่เราเพิ่งพิมพ์ลงไปล่าสุด จะเป็นสิ่งแรกที่ถูกลบออกเมื่อเรากดย้อนกลับ

4. Queue (คิว) เปรียบเหมือน "การต่อแถวซื้อไอศกรีม"

- หลักการ: ทำงานแบบตรงไปตรงมาคือ "มาก่อน ได้ก่อน" (FIFO - First In, First Out)
- จุดเด่น: ใครมาเข้าแถวก่อน (Enqueue) จะได้ซื้อก่อนและออกจากแถวไปก่อน (Dequeue) คนที่มาทีหลังก็ต้องไปต่อท้ายแถวตามระเบียบ
- การใช้งาน: เหมือนระบบส่งพิมพ์เอกสาร (Printer) ที่ส่งงานไป 5 งาน งานที่กดสั่งคนแรกจะได้พิมพ์ออกมาก่อนเพื่อน

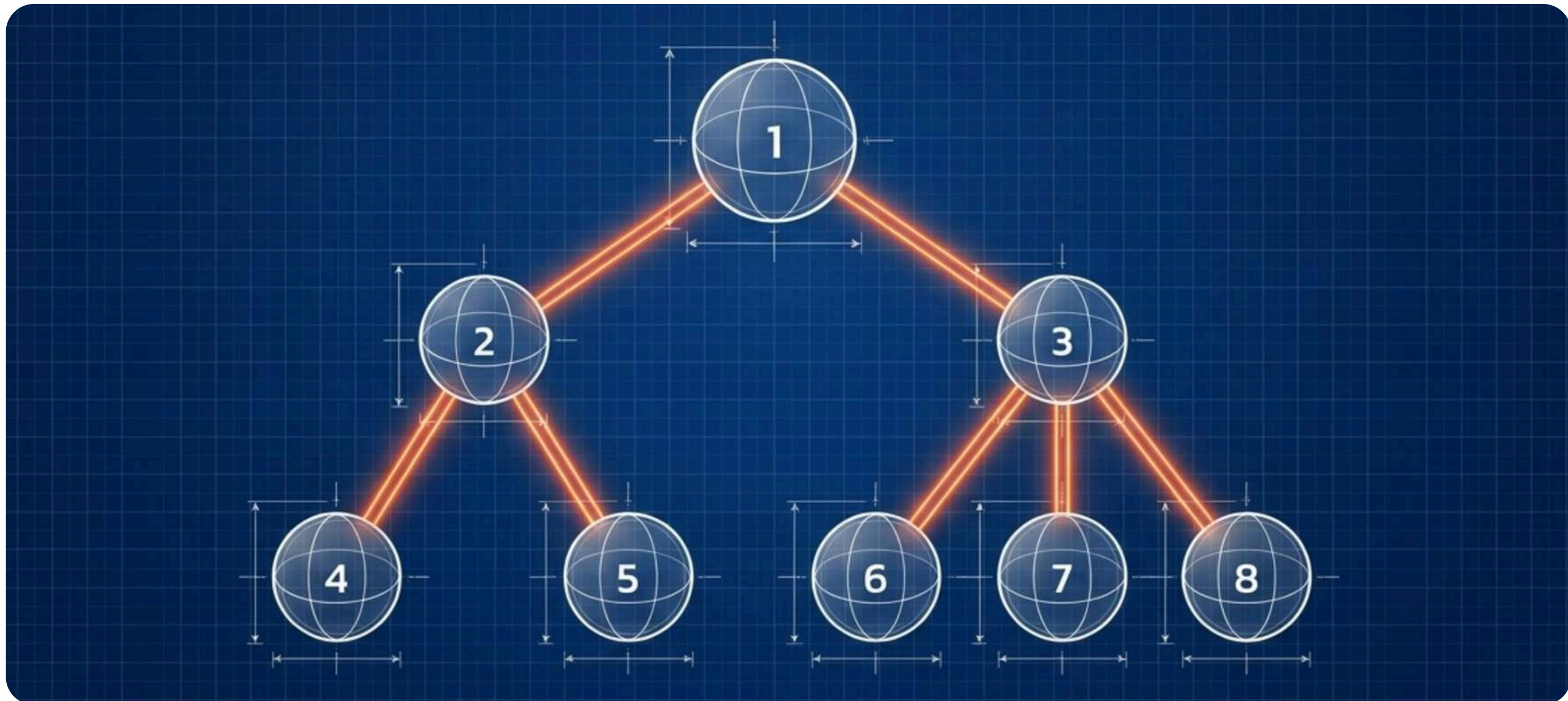
โครงสร้างข้อมูลไม่เชิงเส้น (Non-Linear Data Structure)

การจัดเก็บข้อมูลแบบไม่ต่อเนื่อง ใช้วิธีการเข้าถึงข้อมูลโดยการระบุตำแหน่งหรือความสัมพันธ์



โครงสร้างข้อมูลไม่เชิงเส้น (Non-Linear Data Structure)

โครงสร้างข้อมูลแบบไม่เป็นเชิงเส้น (Non-Linear Data Structures) คือ วิธีการจัดระเบียบข้อมูลในคอมพิวเตอร์ที่ไม่ได้เรียงต่อกันเป็นเส้นตรงแบบ 1, 2, 3, 4 อีกต่อไปแล้วครับ แต่ข้อมูลหนึ่งตัวสามารถเชื่อมโยงแตกกิ่งก้านสาขาไปหาข้อมูลอื่น ๆ ได้พร้อมกันหลายตัว เปรียบเหมือน "เครือข่ายแมงมุม" หรือ "แผนผังต้นไม้"



โครงสร้างข้อมูลไม่เชิงเส้น (Non-Linear Data Structure)

1. Tree (โครงสร้างแบบต้นไม้) เปรียบเหมือน "แผนผังครอบครัว (Family Tree)" หรือ "โครงสร้างการบริหารโรงเรียน"

- หลักการ: เป็นการจัดเก็บข้อมูลที่มีความสัมพันธ์ลดหลั่นกันเป็นชั้น ๆ (Hierarchy) มีจุดเริ่มต้นบนสุดเรียกว่า "ราก (Root)" แล้วแตกกิ่งก้านลงมาข้างล่างเป็น "กิ่ง" และ "ใบ"
- ลองจินตนาการ:
 - ผังโรงเรียน: บนสุดคือ “ผู้อำนวยการ” แตกกลงมาเป็น “รองผู้อำนวยการฝ่ายต่าง ๆ” แตกกลงมาเป็น “หัวหน้ากลุ่มสาระฯ” ล่างสุดคือ “คุณครูประจำวิชา”
 - โฟลเดอร์ในคอมพิวเตอร์: โดรฟ์ C: โฟลเดอร์ "แผนการสอน" โฟลเดอร์ย่อย "วิชาคอมพิวเตอร์" ไฟล์ "สไลด์บทที่ 1"
- จุดเด่น: ค้นหาข้อมูลได้เร็วมาก เพราะคอมพิวเตอร์สามารถเลือกเดินตามกิ่งก้านที่ถูกต้องได้เลย โดยไม่ต้องไล่ดูข้อมูลทุกตัวตั้งแต่ต้นจนจบ

โครงสร้างข้อมูลไม่เชิงเส้น (Non-Linear Data Structure)

2. Graph (โครงสร้างแบบกราฟ) เปรียบเหมือน "เครือข่ายเพื่อนใน Facebook" หรือ "แผนที่เส้นทางรถไฟฟ้า"

- หลักการ: เป็นโครงสร้างที่อิสระที่สุด ข้อมูลแต่ละจุด (เรียกว่า Vertex) สามารถลากเส้นเชื่อมโยง (เรียกว่า Edge) ไปหาจุดอื่น ๆ ได้อย่างทั่วถึง ไม่มีหัวหน้า ไม่มีลูกน้อง ทุกตัวมีสิทธิ์เชื่อมโยงถึงกันได้หมดเป็นเครือข่าย
- ลองจินตนาการ:
 - โซเชียลมีเดีย: ตัวเรา (จุด A) เป็นเพื่อนกับ นายเจี๋ยว (จุด B) และ นายแดง (จุด C) ขณะเดียวกัน นายเจี๋ยวก็เป็นเพื่อนกับนายแดงด้วย ทุกคนเชื่อมโยงกันเป็นวงกลมหรือใยแมงมุม
 - แผนที่การเดินทาง: สถานีรถไฟฟ้าแต่ละสถานีคือจุดข้อมูล ส่วนรางรถไฟคือเส้นเชื่อม ถ้าเราจะเดินทางจากสถานีสยามไปจตุจักร คอมพิวเตอร์จะใช้โครงสร้างแบบกราฟนี้ในการคำนวณหาเส้นทางที่สั้นที่สุดให้เรา
- จุดเด่น: ใช้แสดงความสัมพันธ์ที่ซับซ้อนในชีวิตจริงได้ดีที่สุด

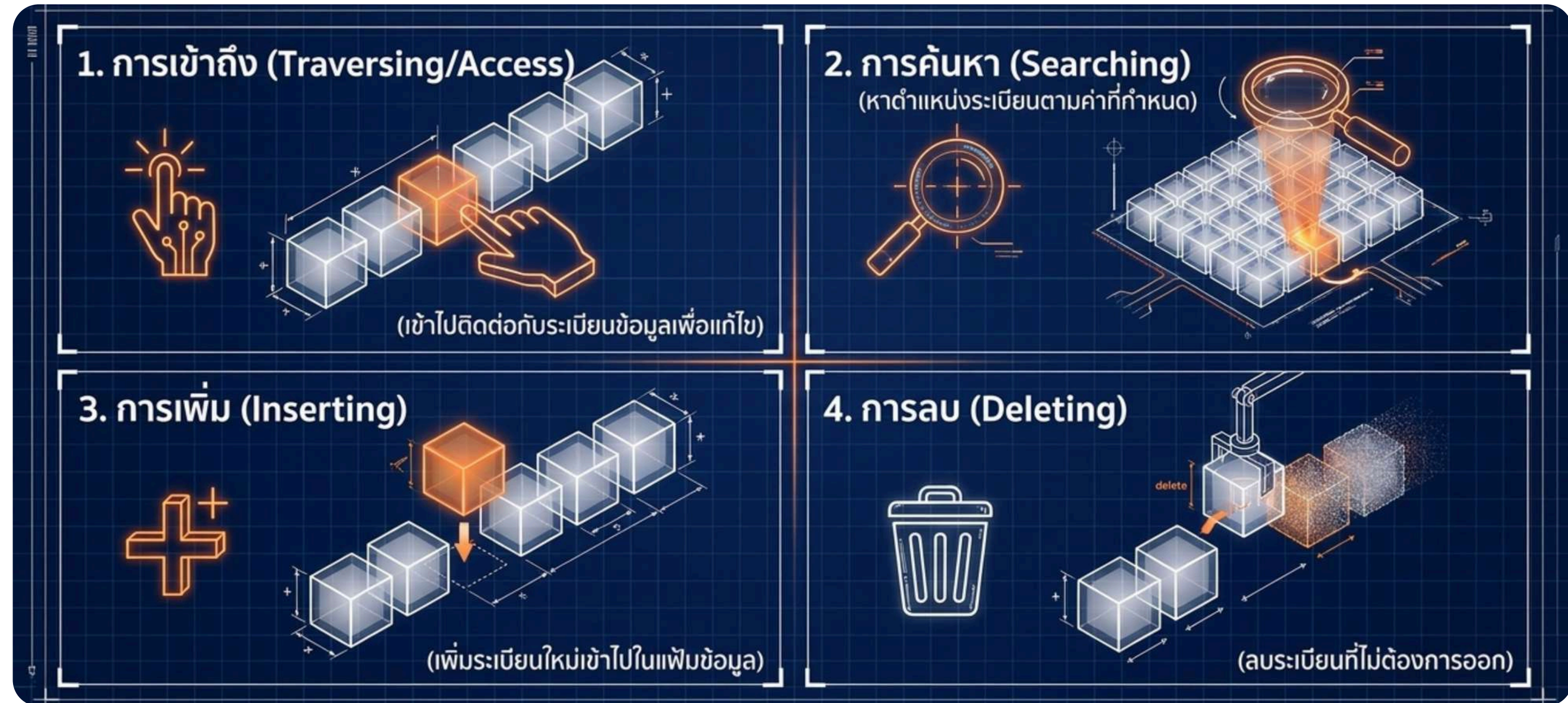
ตารางเปรียบเทียบ:

โครงสร้างข้อมูลเชิงเส้น vs โครงสร้างข้อมูลไม่เชิงเส้น

มิติการเปรียบเทียบ	เชิงเส้น (Linear)	ไม่เชิงเส้น (Non-Linear)
ลำดับการจัดเก็บ	ต่อเนื่องเป็นแถว	ไม่ต่อเนื่อง แยกกิ่งก้าน
วิธีการเข้าถึงข้อมูล	ตามลำดับ (Sequential)	ตำแหน่งอ้างอิง (Relational)
รูปแบบภาพจำ	ขบวนรถไฟ / การต่อคิว	แผนผังต้นไม้ / เครือข่ายใยแมงมุม

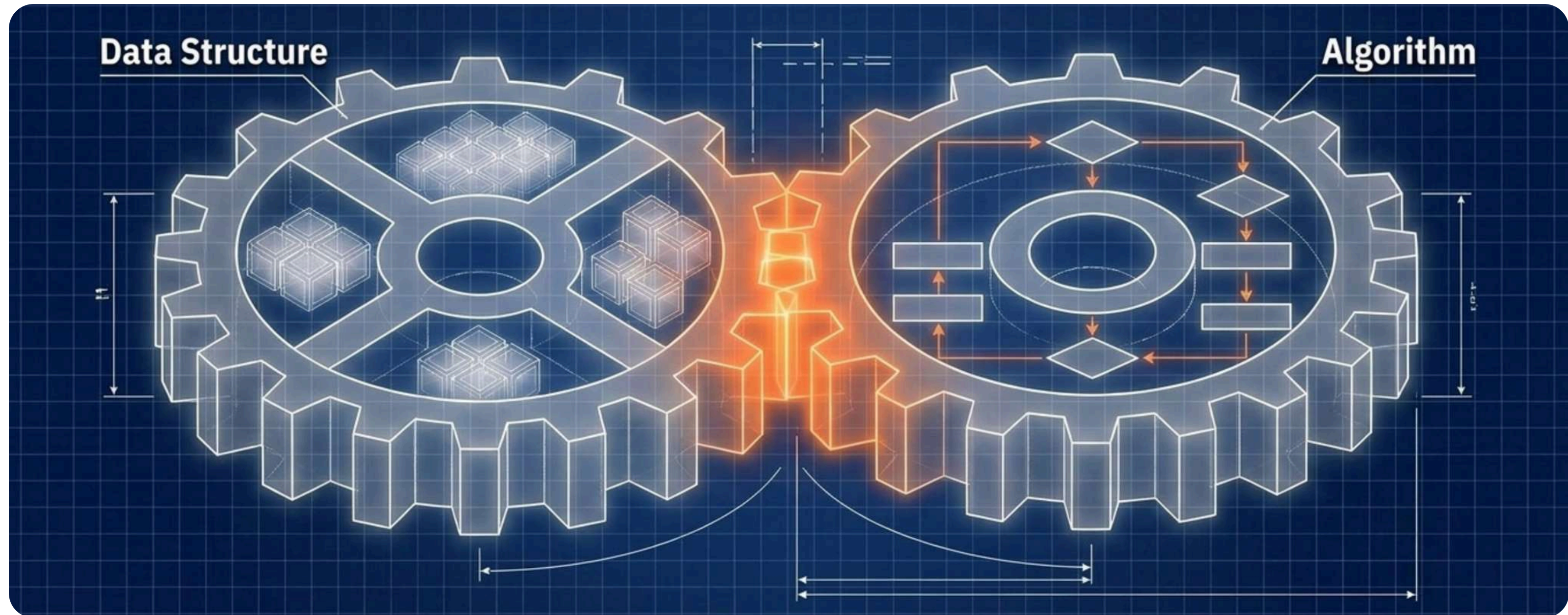
การดำเนินการพื้นฐาน (Core Operations Dashboard)

ไม่ว่าโครงสร้างข้อมูลจะมีรูปแบบใด จะต้องรองรับวิธีการดำเนินการพื้นฐาน 4 ประการเสมอ



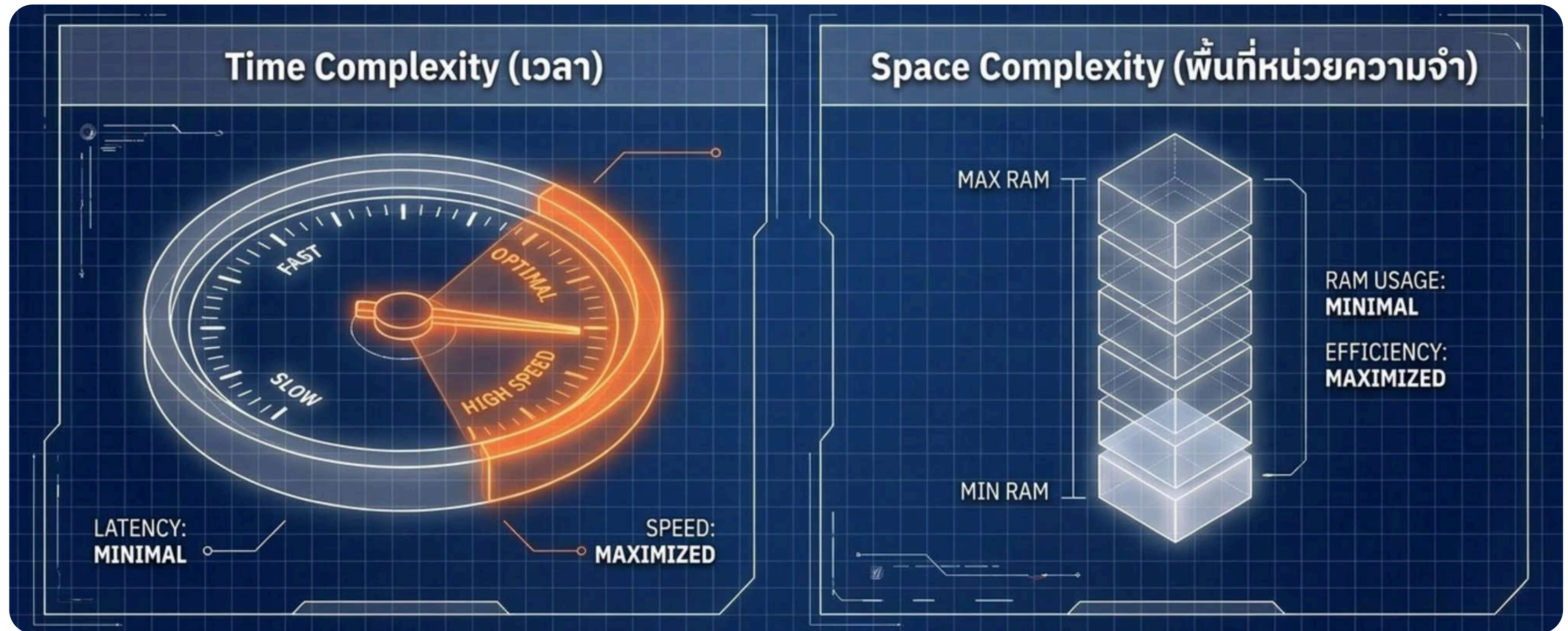
ฟันเฟืองแห่งประสิทธิภาพ: โครงสร้างข้อมูล + อัลกอริทึม

โครงสร้างข้อมูล (Data Structure) คือ สถานที่และรูปแบบการเก็บข้อมูล
ส่วน **อัลกอริทึม (Algorithm)** คือ ขั้นตอนและวิธีการนำข้อมูลไปใช้งาน



สิ่งขัดขวางความฉลาดของอัลกอริทึม

อัลกอริทึมใดที่สามารถแก้ปัญหาเดียวกันได้ แต่ใช้เวลาในการประมวลผลเร็วกว่า และใช้หน่วยความจำน้อยกว่า ถือว่าเป็น “**อัลกอริทึมที่ฉลาดกว่า**”



อัลกอริทึม (Algorithm)

อัลกอริทึม มีที่มาจากชื่อของนักคณิตศาสตร์ชาวเปอร์เซียในยุคศตวรรษที่ 9 ชื่อ อะบู अब್ದิลลาห์ มุฮัมมัด บิน มูซา อัลคอวาริซมี (Abu Abdillah Muhammad bin Musa al-Khawarizmi) คำว่าอัลคอวาริซมี (al-Khawarizmi) ได้เพี้ยนเป็นอัลกอริทมิ (Algoritmi) เมื่องานเขียนของเขา ได้รับการแปลเป็นภาษาละติน แล้วกลายเป็นอัลกอริทึม (Algorithm) ซึ่งหมายถึงกฎที่ใช้ในการคิดคำนวณเลขคณิตในช่วงศตวรรษที่ 18 และในปัจจุบันคำนี้ได้มีความหมายที่กว้างขึ้น โดยหมายรวมถึงขั้นตอน วิธีการในการแก้ปัญหิต่าง ๆ

อ้างอิง

<https://www.scimath.org/lesson-technology/item/8809-2018-09-21-02-51-34>



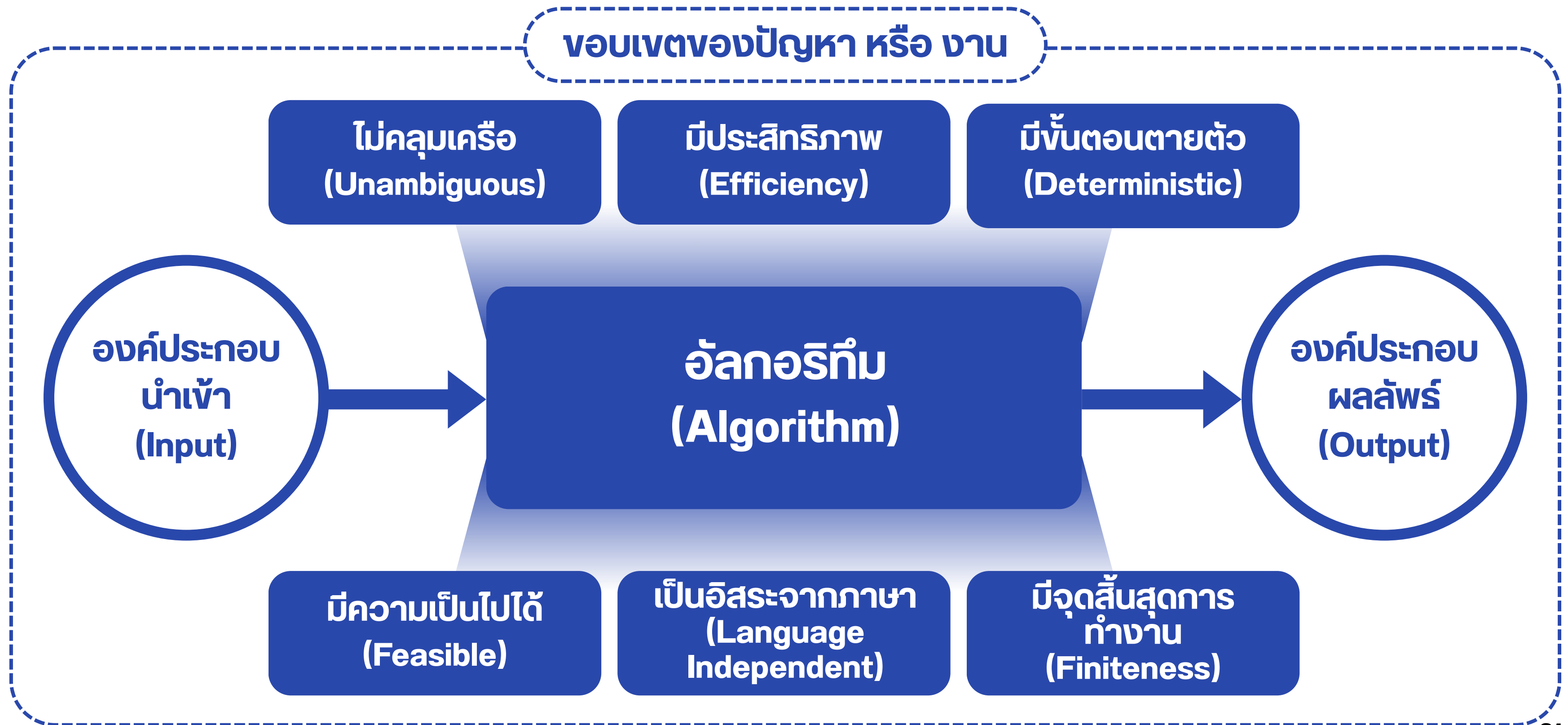
อัลกอริทึม (Algorithm)

อัลกอริทึม (Algorithm) คือ ชุดของขั้นตอนและกฎกติกาที่ถูกกำหนดมา เพื่อแก้ปัญหาคือดำเนินการกระบวนการที่สามารถทำงานได้ด้วยมนุษย์ คอมพิวเตอร์หรือเครื่องจักรอัตโนมัติ ซึ่งขั้นตอน (Algorithm) เป็นการปฏิบัติตามลำดับและหลักการที่แน่นอน เพื่อให้ได้ผลลัพธ์ที่ถูกต้องและเชื่อถือได้ทุกครั้งที่ทำซ้ำ หรือได้ผลลัพธ์ภายใต้เงื่อนไขที่เหมือนกันเสมอ อัลกอริทึมเป็นสิ่งสำคัญที่ส่วนใหญ่ในการแก้ปัญหาคือการพัฒนาซอฟต์แวร์ การวิเคราะห์ข้อมูล การเรียนรู้ของเครื่อง การค้นหาข้อมูล และงานอื่น ๆ ที่เกี่ยวข้องกับการประมวลผลทางด้านคอมพิวเตอร์

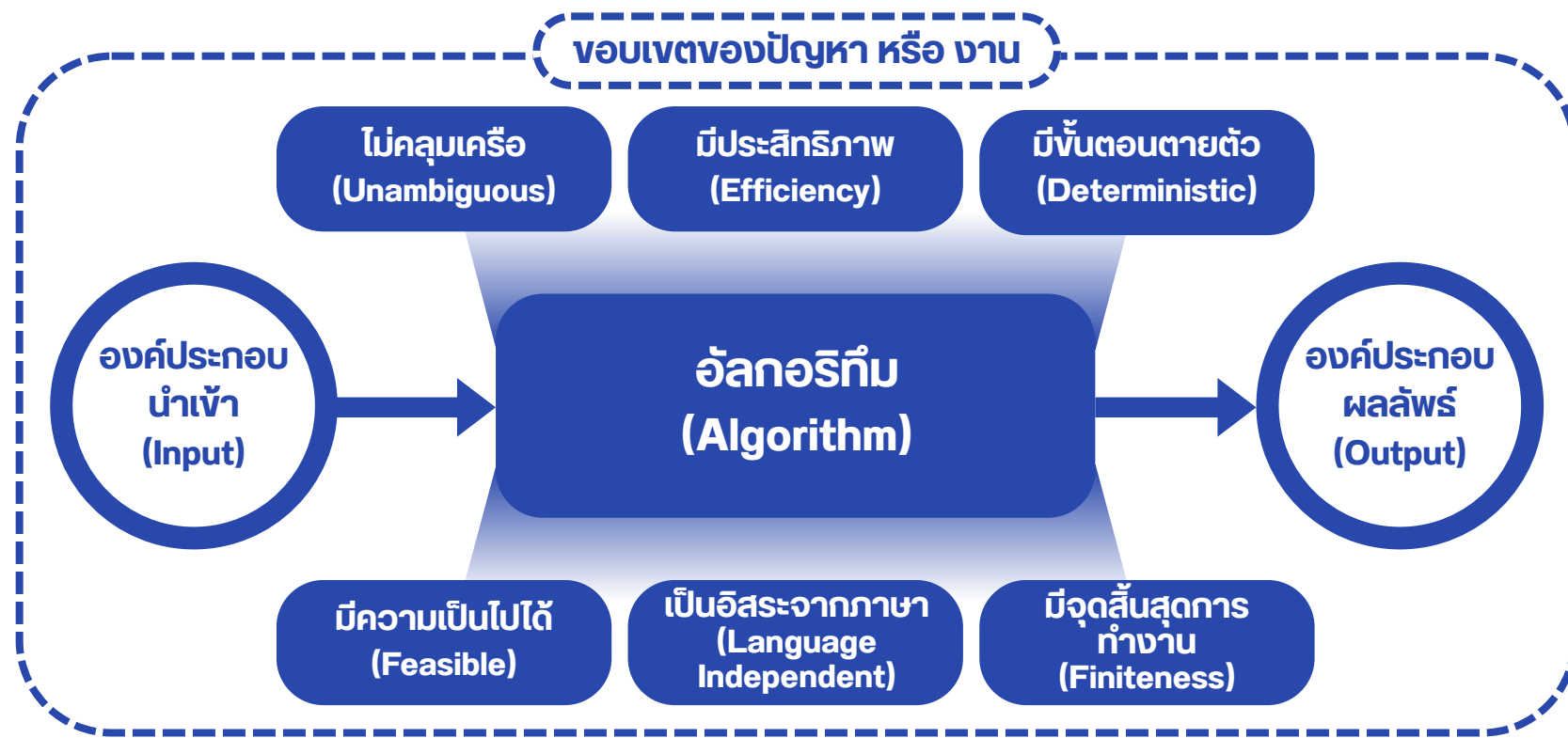
อ้างอิง

- 1.Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press. ISBN: 978-0262033848.
- 2.Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley Professional. ISBN: 978-0321573513.
- 3.Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. (2008). Algorithms. McGraw-Hill Science/Engineering/Math. ISBN: 978-0073523408.

คุณลักษณะของอัลกอริทึม (Characteristics of an Algorithm)



คุณลักษณะของอัลกอริทึม (Characteristics of an Algorithm)



ลักษณะของอัลกอริทึมที่ดีที่ประกอบด้วยคุณสมบัติ ดังนี้

- 1. ไม่คลุมเครือ (Unambiguous)** อัลกอริทึมต้องไม่มีความกำกวมที่ไม่ชัดเจน และต้องมีชุดคำสั่งที่เข้าใจง่ายและไม่ทำให้เกิดความสับสนในการประมวลผล
- 2. มีประสิทธิภาพ (Efficiency)** อัลกอริทึมควรมีประสิทธิภาพในการทำงาน คือต้องให้ผลลัพธ์ที่ถูกต้อง และใช้เวลาน้อยที่สุดในการแก้ปัญหา

3. มีขั้นตอนตายตัว (Deterministic) อัลกอริทึมต้องเป็น Deterministic หมายความว่า จะต้องให้ผลลัพธ์ที่เหมือนกันสำหรับข้อมูลนำเข้าที่เหมือนกันทุกครั้งที่รับ

4. มีความเป็นไปได้ (Feasible) อัลกอริทึมต้องเป็นไปได้ในการนำไปใช้งานจริง สามารถทำงานบนเครื่องคอมพิวเตอร์หรือสภาพแวดล้อมที่เหมาะสม

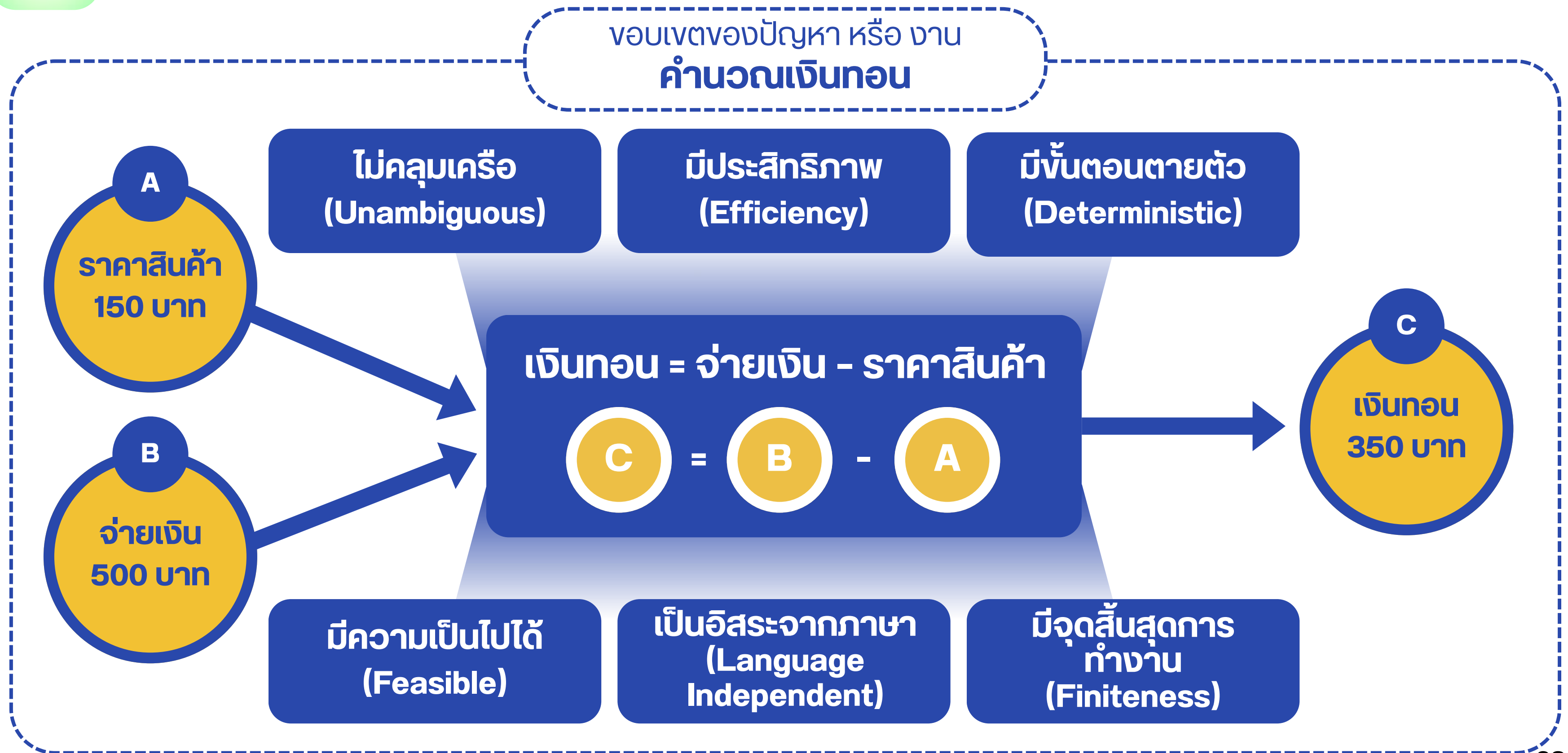
5. เป็นอิสระจากภาษา (Language Independent) อัลกอริทึมควรไม่เกี่ยวข้องกับภาษาเฉพาะเสียเพียงอย่างเดียว และสามารถนำไปใช้งานในหลายภาษาโปรแกรมได้

6. มีจุดสิ้นสุดการทำงาน (Finiteness) อัลกอริทึมต้องสามารถสิ้นสุดการทำงานหลังจากทำขั้นตอนทุกขั้นตอนเสร็จสมบูรณ์ และไม่ต้องเข้าสู่วงจรไม่สิ้นสุด (Infinity Loop)

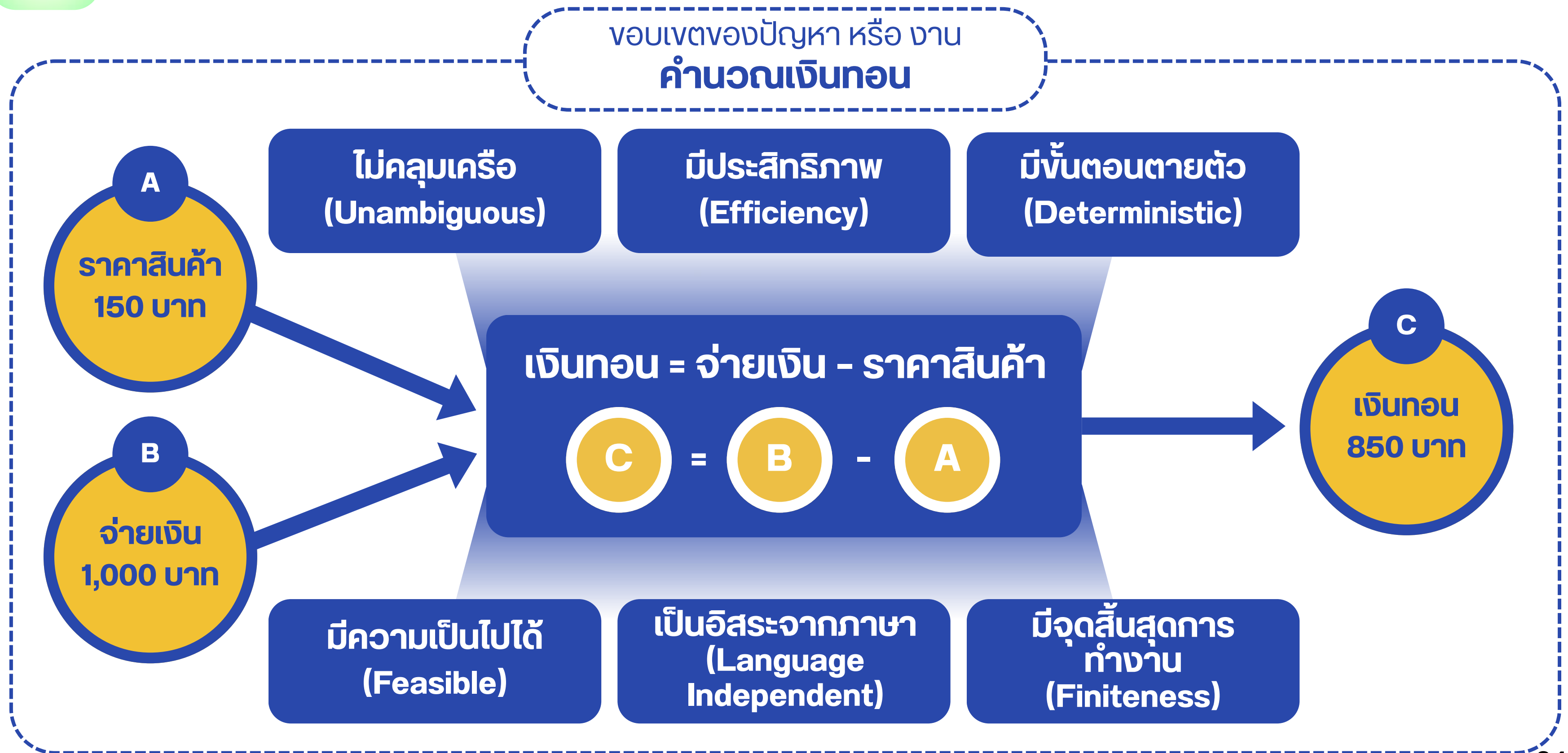
อ้างอิง

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press. ISBN: 978-0262033848.
- Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley Professional. ISBN: 978-0321573513.

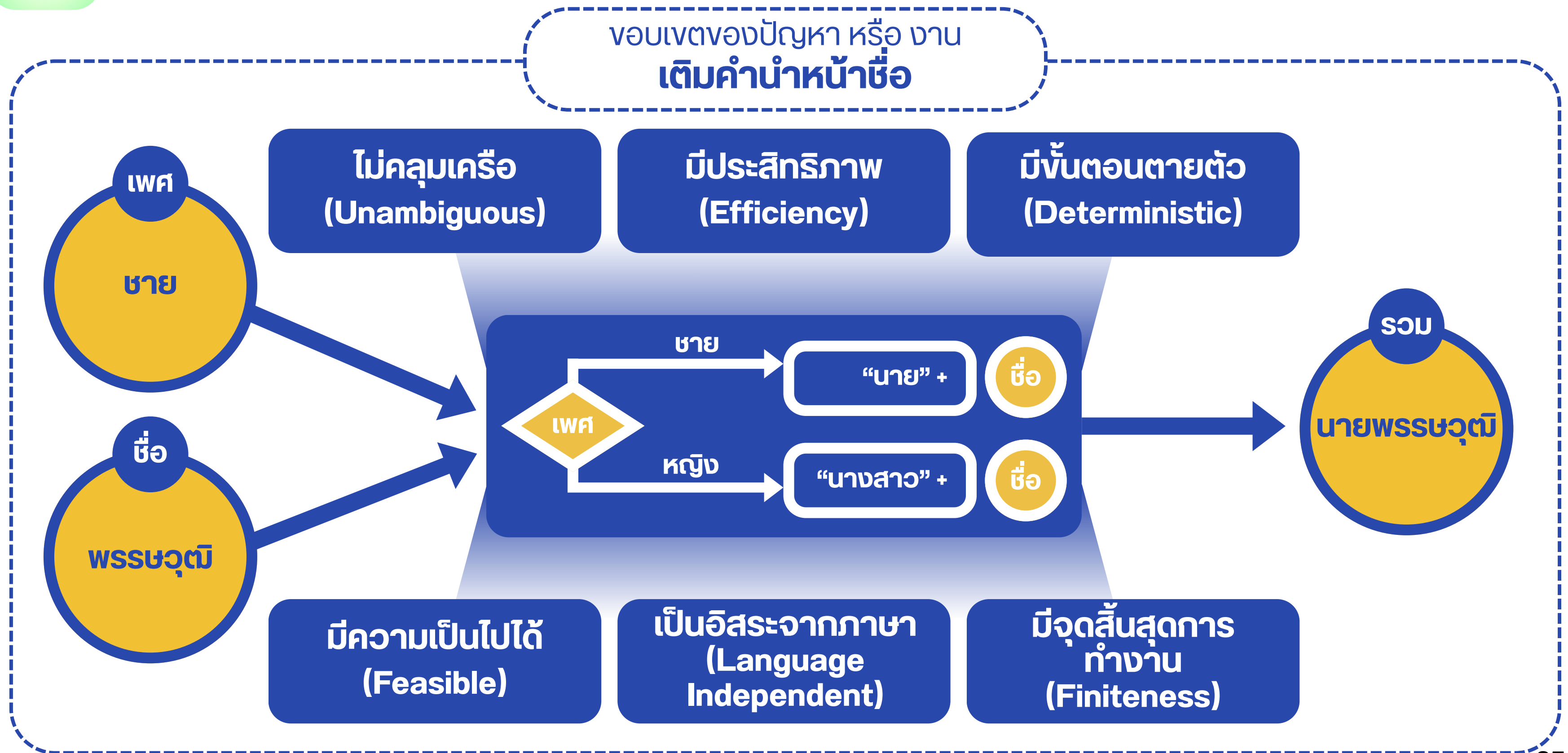
คุณลักษณะของอัลกอริทึม (Characteristics of an Algorithm)



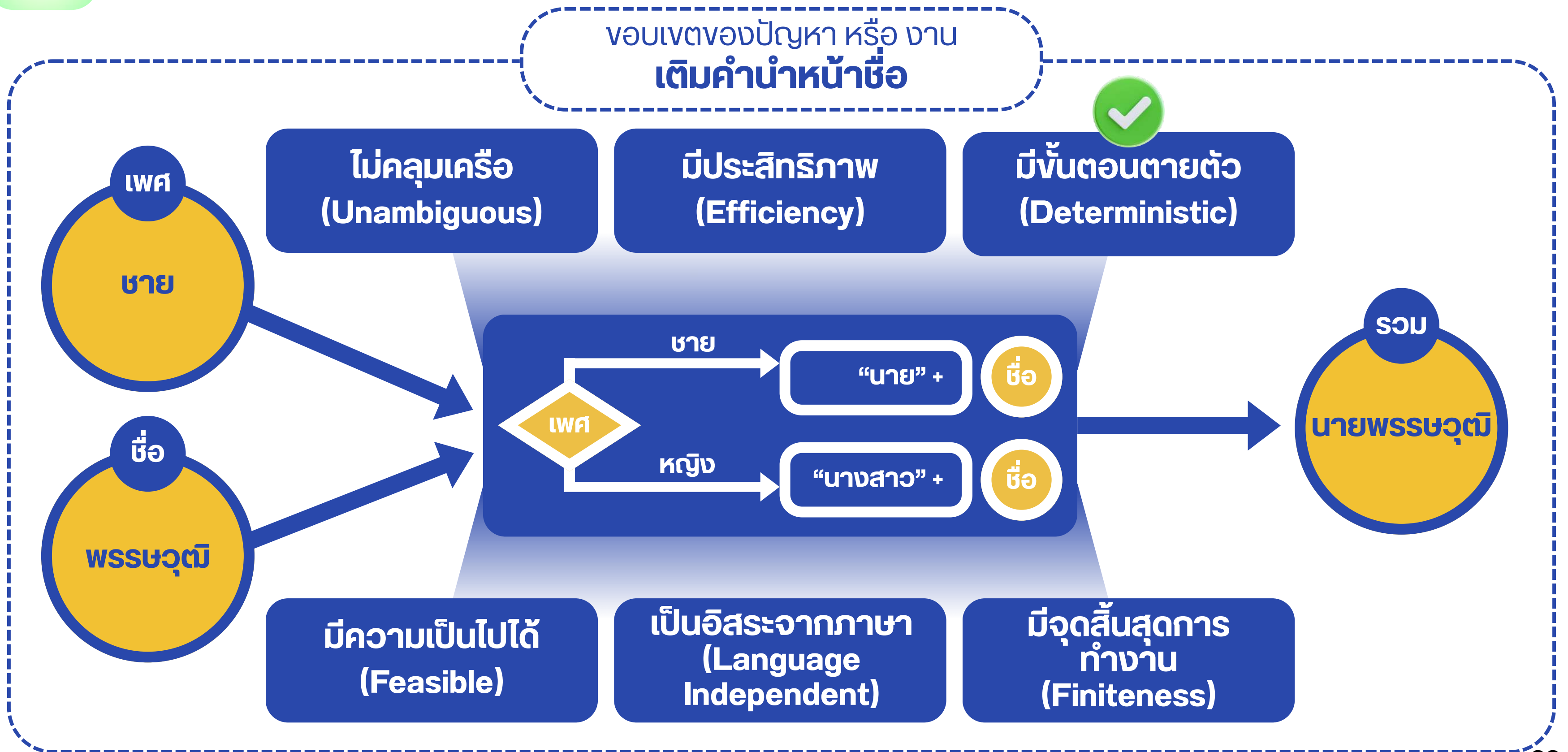
คุณลักษณะของอัลกอริทึม (Characteristics of an Algorithm)



คุณลักษณะของอัลกอริทึม (Characteristics of an Algorithm)



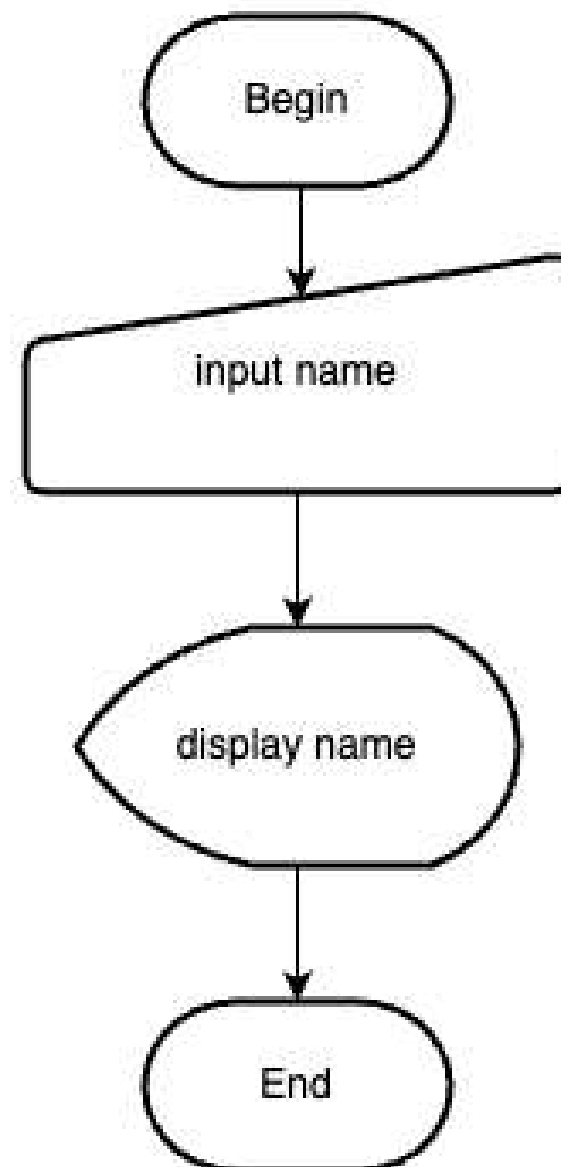
ให้ผลลัพธ์ถูกต้องตามเงื่อนไขที่กำหนดไว้เสมอ คือ คุณลักษณะข้อใด



อัลกอริทึม (Algorithm)

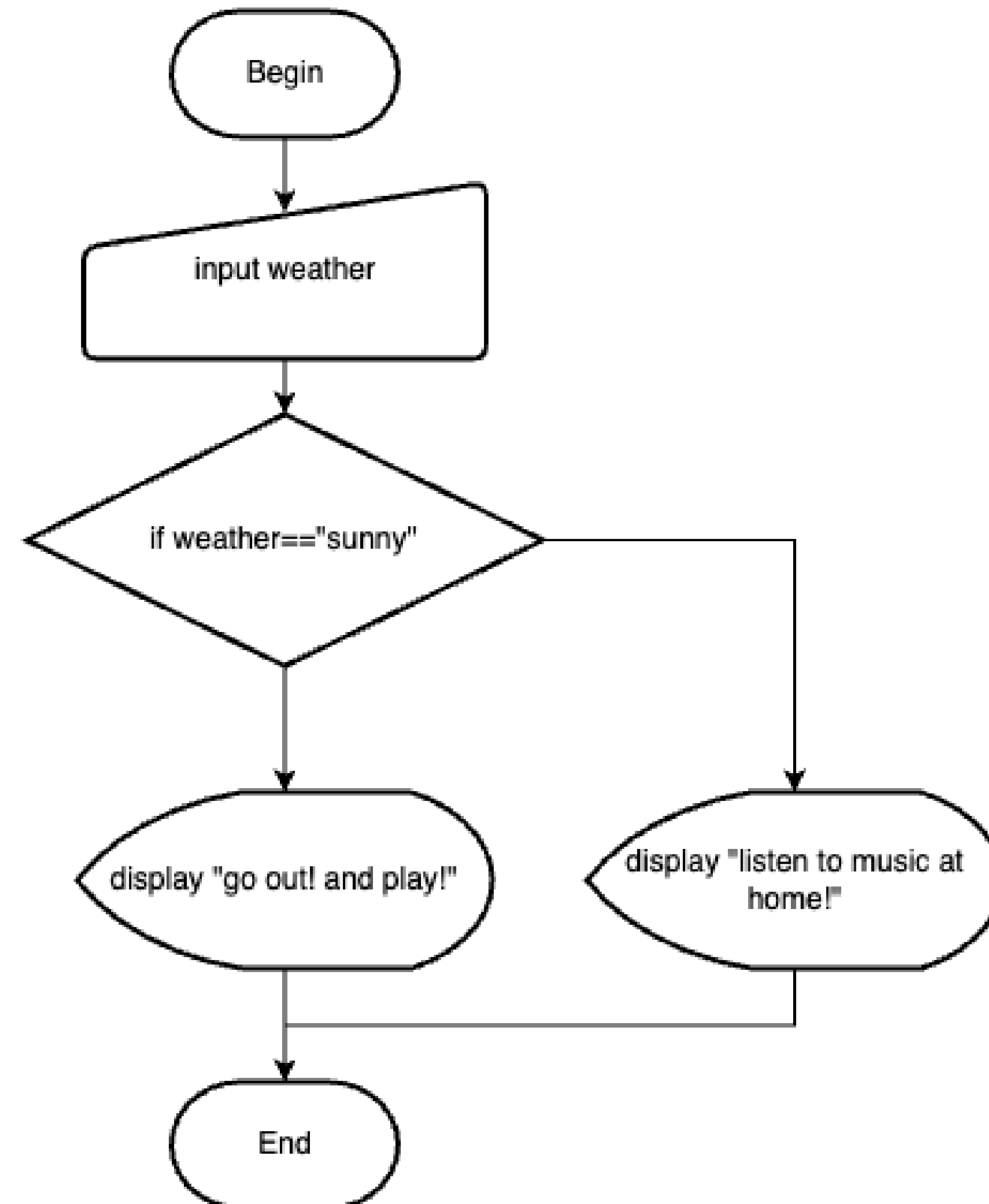
Flowchart: Show name

Objective: Display name from keyboard input



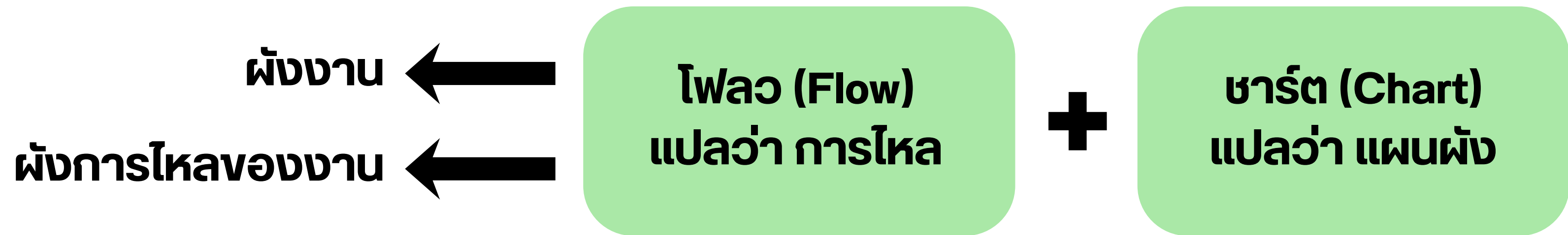
Flowchart: Play or Not?

Objective: Evaluate the weather conditions in deciding to go out and play.



อัลกอริทึม (Algorithm)

การแสดงอัลกอริทึมด้วยผังงาน หรือ โฟลวชาร์ต (Flowchart)



เมื่อรวมกันแล้ว จึงหมายถึง แผนผังที่แสดงการไหลของงาน หรือ แผนผังที่แสดงขั้นตอนการทำงานอย่างชัดเจน นั่นเอง ซึ่งสามารถนำผังงานมาใช้แสดงขั้นตอนการทำงานของโปรแกรมได้

Assignment # 1

1. จงอธิบายความหมายของโครงสร้างข้อมูลและอัลกอริทึม
2. จงอธิบายความสำคัญของการศึกษาโครงสร้างข้อมูลและอัลกอริทึม
3. จงอธิบายประเภทของโครงสร้างข้อมูล และยกตัวอย่างพร้อมอธิบายข้อมูลแต่ละประเภท



Data Structures and Algorithms

Thank You

COZ1302 Data Structures and Algorithms

Department of Innovation and Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University



Pasawut Cheerapakorn
Suan Sunandha Rajabhat University