

แผนการสอนสัปดาห์ที่ 1

บทที่ 1 ทฤษฎีของขั้นตอนวิธี

จำนวนชั่วโมง 3

- 1.1 ทำความรู้จักกับอัลกอริธึม
- 1.2 การวิเคราะห์ปัญหาทางคอมพิวเตอร์
- 1.3 การออกแบบอัลกอริธึม

จุดประสงค์การสอน (จุดประสงค์ทั่วไป)

- 1.1 รู้จักกับอัลกอริธึม
- 1.2 รู้ถึงการวิเคราะห์ปัญหาทางคอมพิวเตอร์
- 1.3 เข้าใจการออกแบบอัลกอริธึม

ผลการเรียนรู้ (จุดประสงค์เฉพาะ)

- 1.1 บอกถึงหลักการของอัลกอริธึม
- 1.2 บอกถึงวิเคราะห์ปัญหาทางคอมพิวเตอร์
- 1.3 อธิบายถึงออกแบบอัลกอริธึม

วิธีสอนและกิจกรรมการเรียนการสอน

สอนด้วยการบรรยายพร้อมยกตัวอย่างประกอบอธิบายเนื้อหาโดย

1. ผู้สอนแจ้งวัตถุประสงค์การเรียนรู้ให้นักศึกษาทราบ
2. ผู้สอนบรรยายนำเข้าสู่บทเรียน เนื้อหาในบทเรียน ใช้ power point ประกอบการบรรยายและเขียนอธิบายเพิ่มเติมบนกระดานไวท์บอร์ด
3. ให้ผู้เรียนตอบคำถามท้ายบทเรียน

สื่อการสอน/อุปกรณ์การสอน

1. กระดานไวท์บอร์ด
2. เอกสารประกอบการสอน
3. สไลด์ power point นำเสนอเนื้อหาประกอบการสอน

การวัดผล

1. สังเกตพฤติกรรมการเรียน
2. สอบกลางภาค

บทที่ 1 ทฤษฎีของขั้นตอนวิธี

1.1 ทำความรู้จักกับอัลกอริทึม

อัลกอริทึม หมายถึง ลำดับขั้นตอนวิธีในการทำงานของโปรแกรมเพื่อแก้ปัญหาใดปัญหาหนึ่ง ซึ่งถ้าปฏิบัติตามขั้นตอนอย่างถูกต้องแล้ว จะต้องสามารถช่วยแก้ปัญหาหรือประมวลผลตามต้องการได้สำเร็จ เช่น แนวคิดในการแก้ปัญหาโดยใช้ผังงาน หรือซูโดโค้ด (รหัสเทียม)

1. การกำหนดค่าเริ่มต้น (Initialization) เป็นการกำหนดค่าเริ่มต้นให้กับตัวแปรบางตัว เช่น ตัวแปรที่ทำหน้าที่เป็นหน่วยนับ (Counter) ตัวแปรที่ค่าเป็นผลการคำนวณสะสม เช่น เรากำหนดให้ High มีค่าเริ่มต้นเท่ากับ 0 ($high = 0$) หรือ long มีค่าเริ่มต้นเท่ากับ 0 ($long = 0$)
2. การรับข้อมูล (Input) เป็นการรับค่าตัวแปรที่ระบุไว้ในขั้นตอนการนำ เข้าข้อมูลของการวิเคราะห์งานการรับข้อมูลจะ ต้องกระทำก่อนจะมีการนำข้อมูลนั้นไปใช้ เช่น นำไปคำนวณ เช่น เรารับข้อมูลของ $High = 10, long = 5$
3. การประมวลผล (Process) เป็นการแสดงวิธีประมวลผล เช่น พื้นที่ของรูปสี่เหลี่ยม = $high * long$ ดังนั้น พื้นที่ของรูปสี่เหลี่ยม = $10 * 5$
4. การแสดงผลลัพธ์ (Output) เป็นการแสดงผลลัพธ์ หรือค่าของตัวแปรที่ระบุไว้ ในหัวข้อผลลัพธ์ที่ต้องแสดงในการวิเคราะห์งาน เช่น เรานำความสูงของรูปสี่เหลี่ยมที่คำนวณได้ที่เท่ากับ 50 มาแสดงทางหน้าจอ
5. การเปรียบเทียบ (Decision) เป็นการเปรียบเทียบเพื่อเรียงข้อมูลหรือเพื่อย้อน กลับไปทำซ้ำตัวอย่างโจทย์ที่ใช้ในการแก้ปัญหา “จงหาว่ารูปสามเหลี่ยมมีพื้นที่เท่าไร?”

ตัวอย่างวิธีทำ

1. ตรวจสอบว่าควรใช้ตัวแปรใดในการเก็บข้อมูล จากสูตรการหาพื้นที่สามเหลี่ยม คือ $0.5 * สูง * ฐาน$ ทำให้เรารู้ว่าความสูงและความยาวของฐานมาจากการที่ใส่ข้อมูลเราจึงจะต้องกำหนดตัวแปรให้ ความสูงกับความยาวของฐานโดยทั่วไปจะไม่นิยมใช้ชื่อตัวแปรเป็นภาษาไทย

จากสูตรจะให้ $high =$ ความสูง

$long =$ ความยาวของฐาน

$total =$ พื้นที่ของรูปสามเหลี่ยม

2. จากสูตรการรับข้อมูลเรารับข้อมูลจากคีย์บอร์ด ซึ่งจะมีตัวแปรที่รับเข้ามาคือ $high = 5, long = 5$

3. การประมวลผลเราจะทำการประมวลผลจากสูตร $0.5 * high * long$

จากการรับข้อมูลเข้าเราทำการคำนวณได้ดังนี้ $total = 0.5 * 5 * 5 = 12.5$

4. การแสดงผลลัพธ์เราจะนำค่าของพื้นที่สามเหลี่ยมที่คำนวณได้มาแสดงผลทางหน้าจอในที่นี้คือ 12.5

1.1 ทำความรู้จักกับขั้นตอนวิธี (An Introduction to Algorithm)

■ ขั้นตอนวิธี (Algorithm) = ?

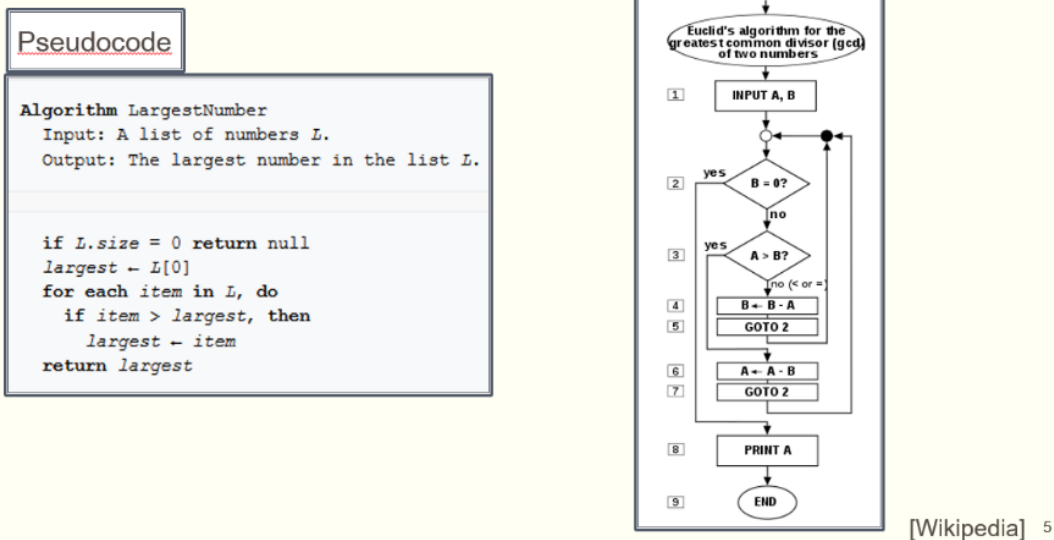
→ วิธีการเครื่องมือที่ช่วยแก้ปัญหาอย่างเป็นขั้นตอนตามลำดับ เพื่อให้ได้ผลลัพธ์อย่างมีประสิทธิภาพ บางปัญหาต้องการวิธีการทางคอมพิวเตอร์ (computer algorithm) ช่วยในการแก้ปัญหา ซึ่งแตกต่างจากการแก้ปัญหาแบบสามัญสำนึก (heuristic)

→ ขั้นตอนวิธีในการแก้ปัญหาอาจนำเสนอ: natural languages, pseudo code, flowcharts

เมื่อได้เขียนขั้นตอนวิธีแล้วก็จะทำการพัฒนาโปรแกรมด้วยภาษาคอมพิวเตอร์ (High-level programming languages: C++, Java) ให้ได้ผลลัพธ์ตามขั้นตอนวิธีที่ได้ออกแบบไว้

Algorithms

Flowchart

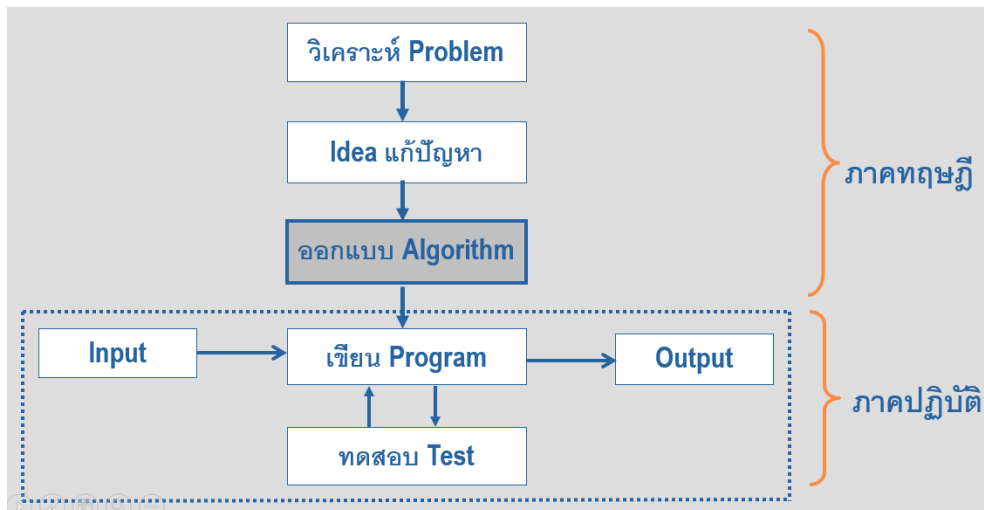


ภาพที่ 1.1 เขียนขั้นตอนวิธีแบบซูโดโค้ดและผังงานเพื่อแก้ปัญหาทางคอมพิวเตอร์

ในภาพที่ 1.1 แสดง algorithm ประกอบด้วย วิธีการเป็นขั้นๆ ดังนี้

- แบบวนซ้ำ (iterate)
- เรียกตัวเอง (recursive) โดยใช้ตรรกะ (logic) และ/หรือในการเปรียบเทียบ (comparison) ในขั้นตอนต่างๆ จนกระทั่งเสร็จสิ้นการทำงาน
- Algorithms ที่ต่างกันซึ่งอาจมีจำนวนชุดคำสั่งที่ใช้ต่างกัน จะใช้เวลาในการประมวลผล และขนาดหน่วยความจำ (space) ที่ต้องการต่างกัน กล่าวคือ มีความซับซ้อน (complexity) ต่างกัน

1.2 การวิเคราะห์ปัญหาทางคอมพิวเตอร์ (Problems Analysis/Solving)



ภาพที่ 1.2 การวิเคราะห์ปัญหาทางคอมพิวเตอร์

กระบวนการแก้ปัญหาด้วยคอมพิวเตอร์ ประกอบไปด้วย 4 ขั้นตอนดังแสดงในภาพที่ 1.2 รายละเอียดดังนี้

1. การวิเคราะห์ปัญหา (Problem Analysis) เพื่อให้เข้าใจปัญหา/ชัดเจน
2. หาแนวคิดในการแก้ปัญหา (Solution Idea) ว่ามีวิธีใดบ้าง
3. การออกแบบและวิเคราะห์ขั้นตอนวิธีทางคอมพิวเตอร์ (Design and Analysis Computer Algorithm) ซึ่งอาจเป็น แบบวนซ้ำ (iterate)/เรียกตัวเอง (recursive)/ตรรกะ (logic)/การเปรียบเทียบ (comparison)
4. การพัฒนาและทดสอบโปรแกรม (Program Development and Testing)

1.3 การออกแบบอัลกอริธึม

กระบวนการทำงาน (procedure) ใน Algorithm เพื่อใช้แก้ปัญหาให้บรรลุเป้าหมายนั้นจะต้อง

- มีขั้นตอนการดำเนินงานที่ชัดเจน
- กระบวนการทำงานควรไม่ซับซ้อน เพื่อที่จะได้สามารถนำไปปฏิบัติการโดยเครื่องคอมพิวเตอร์
- กระบวนการทำงานมีจุดจบ (finiteness)
- นิยมเขียนในรูปแบบของคำสั่งจำลอง (pseudo code)

การออกแบบแนวคิดในการแก้ปัญหา/อัลกอริธึม

- การกำหนดค่าเริ่มต้น (Initialization) เป็นการกำหนดค่าเริ่มต้นให้กับตัวแปรบางตัว
- การรับข้อมูล (Input) เป็นการรับค่าตัวแปรที่ระบุไว้ในขั้นตอนการนำ เข้าข้อมูลของการวิเคราะห์งานการรับข้อมูลจะ ต้องกระทำก่อนจะมีการนำข้อมูลนั้นไปใช้คำนวณ
- การประมวลผล (Process) เป็นการแสดงวิธีประมวลผล เช่น สูตรต่างๆ procedure
- การเปรียบเทียบ (Decision) เป็นการเปรียบเทียบเพื่อเรียงข้อมูลหรือเพื่อย้อนกลับไปทำซ้ำตัวอย่างโจทย์ที่ใช้ในการแก้ปัญหา
- การแสดงผลลัพธ์ (Output) เป็นการแสดงผลลัพธ์ หรือค่าของตัวแปรที่ระบุไว้ ในหัวข้อผลลัพธ์ที่ต้องแสดงในการวิเคราะห์งาน

ในการออกแบบอัลกอริธึม เราจะต้องคำนึงถึง

- ความถูกต้อง
- มีประสิทธิภาพ
- ใช้หน่วยความจำให้น้อยที่สุด
- มีขั้นตอนที่ชัดเจน เข้าใจง่าย