แนวคิดเชิงคำนวณ

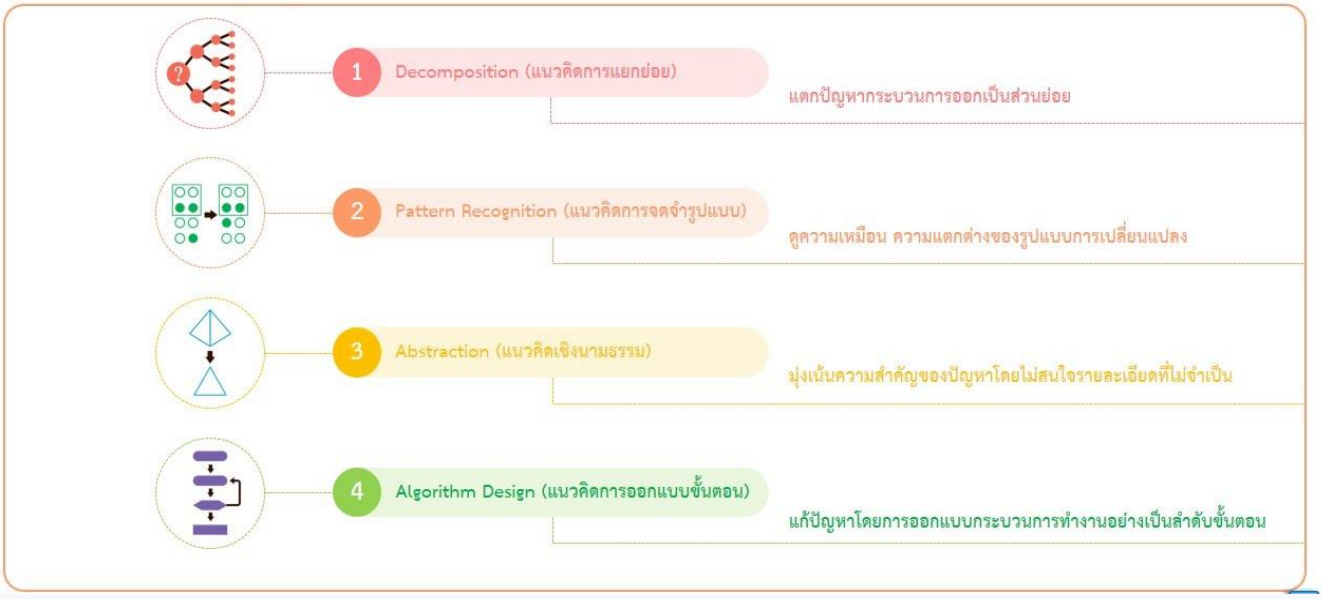
แนวคิดเชิงคำนวณ (Computational thinking) ซึ่งเป็นพื้นฐานของการคิดแก้ปัญหาต่าง ๆ ที่สามารถนำไปประยุกต์ในการแก้ปัญหาในชีวิตประจำวัน แนวคิดนี้ไม่ใช่เรื่องใหม่ เพราะมนุษย์ต้องแก้ปัญหาต่าง ๆ อยู่ตลอดเวลาความท้าทายหลักของแนวคิดเชิงคำนวณอยู่ที่การออกแบบกระบวนการแก้ปัญหาที่คลุมเครือให้เป็นขั้นตอนที่ชัดเจนมากพอที่จะนำไปแก้ปัญหาได้

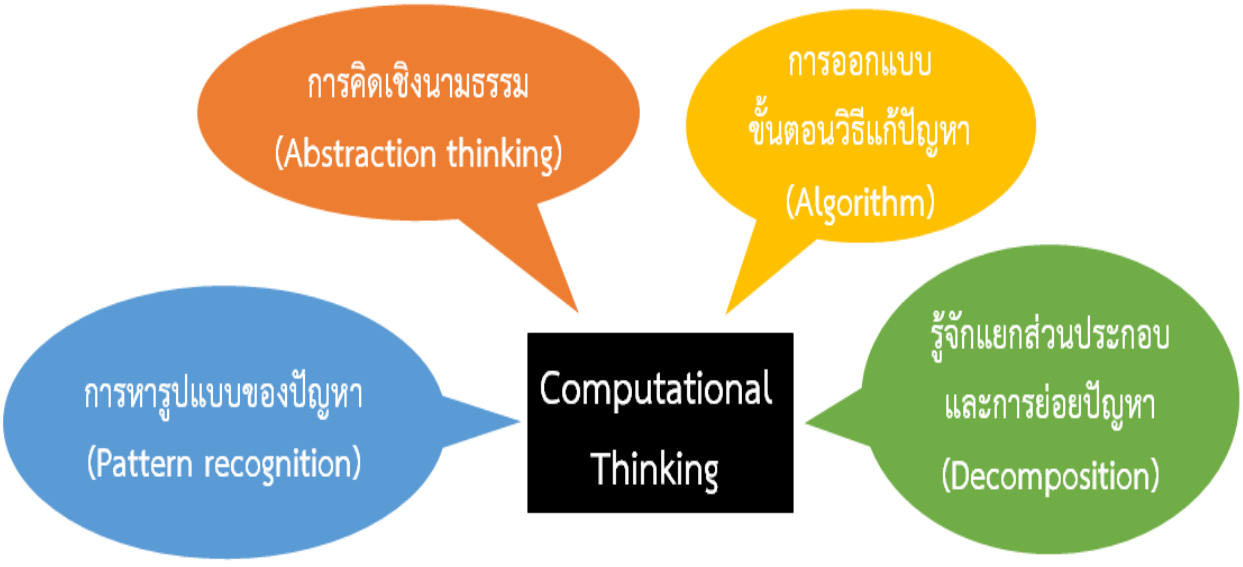
การคิดเชิงคำนวณ (computational thinking) คือกระบวนการแก้ปัญหาในหลากหลายลักษณะ เช่น การจัดลำดับเชิงตรรกศาสตร์ การวิเคราะห์ข้อมูลและการสร้างสรรค์วิธีแก้ปัญหาไปทีละขั้นตอน (หรือที่เรียกว่าอัลกอริทึม) รวมทั้งการย่อยปัญหาที่ช่วยให้รับมือกับปัญหาที่ซับซ้อนหรือมีลักษณะเป็นคำถามปลายเปิดได้ วิธีคิดเชิงคำนวณมีความจำเป็นในการพัฒนาแอพพลิเคชันต่างๆ สำหรับคอมพิวเตอร์ แต่ในขณะเดียวกัน วิธีคิดนี้ยังช่วยแก้ปัญหาในวิชาต่างๆ ได้ด้วย ดังนั้นเอง เมื่อมีการบูรณาการวิธีคิดเชิงคำนวณผ่านหลักสูตรในหลากหลายแขนงวิชา นักเรียนจะเห็นความสัมพันธ์ระหว่างแต่ละวิชา รวมทั้งสามารถนำวิธีคิดที่เป็นประโยชน์ ไปใช้แก้ปัญหาในชีวิตจริงได้ในระยะยาว

แนวคิดเชิงคำนวณ

1 แนวคิดเชิงคำนวณ

แนวคิดเชิงคำนวณ (Computational Thinking) คือ กระบวนการคิดในการแก้ไขปัญหาที่ทั้งมนุษย์และคอมพิวเตอร์สามารถเข้าใจร่วมกันได้





แนวคิดเชิงคำนวณ

1) Decomposition ชื่อไทยคือ “การแยกส่วนประกอบ และการย่อยปัญหา”
Decomposition เป็นการพิจารณาเพื่อแบ่งปัญหา หรืองานออกเป็นส่วนย่อย ทำให้สามารถจัดการกับปัญหาหรืองานได้ง่ายขึ้น
พูดง่ายๆ เอาปัญหามาแยกย่อยออกเป็นส่วนๆ

☞ ตัวอย่างการนำแนวคิดนี้ไปใช้ตอนเขียนโปรแกรม
เช่น การเขียนโปรแกรมแยกเป็นส่วนๆ แยกเป็นแพ็กเกจ แยกเป็นโมดูล หรือทำระบบเป็น services ย่อยๆ หรือมองเป็น layer เป็นต้น

☞ ตัวอย่างการนำไปใช้นอกจากเขียนโปรแกรม
-รัฐบาลจะปฏิรูปประเทศไทย ก็จะนำปัญหาประเทศมาแยกย่อยออกเป็นปฏิรูป 11 ด้าน จากนั้นจึงไปปฏิรูปปัญหาย่อยทีละด้าน
-เราจะเรียนรู้ว่าจักรยานทำงานอย่างไร? ก็ให้พิจารณาแยกชิ้นส่วนจักรยานว่ามีอะไรบ้าง แล้วก็ไปศึกษาทีละชิ้น
-เราจะเดินทางไปเที่ยวหาดใหญ่ จะวางแผนเดินทางอย่างไร? ซึ่งเราอาจแยกย่อยวิธีเดินทางเป็น 4 รูปแบบ เช่น ขับรถไปเอง หรือนั่งรถทัวร์ หรือนั่งเครื่องบิน หรือนั่งรถไฟ จากนั้นก็มาวิเคราะห์ถึงข้อดีข้อเสียแต่ละวิธีการ

แนวคิดเชิงคำนวณ

2) Pattern recognition ชื่อไทยคือ “การหารูปแบบ”

Pattern recognition เป็นทักษะการหาความสัมพันธ์ที่เกี่ยวข้อง แนวโน้ม และลักษณะทั่วไปของสิ่งต่าง ๆ

☞ ตัวอย่างการนำแนวคิดนี้ไปใช้ตอนเขียนโปรแกรม

เมื่อมีการทำงานของโปรแกรมที่หลากหลายแบบ แต่ทว่ามีรูปแบบที่แน่นอนซ้ำๆ กัน เราสามารถยุบโค้ดมาอยู่ในฟังก์ชั่นเดียวกันได้หรือไม่ หรือเขียนเป็นโปรแกรมวนลูป ให้อยู่ในลูปเดียวกัน เป็นต้น

☞ ตัวอย่างการนำไปใช้นอกจากเขียนโปรแกรม

-จัดหมวดหมู่สัตว์ที่คล้ายคลึงกัน ให้อยู่ในสปีชีส์เดียวกัน เพื่อให้ง่ายต่อการศึกษา

-หาพฤติกรรมการบริโภคของคน ว่านิยมซื้ออะไร ช่วงเวลาไหน มีรูปแบบพฤติกรรมซ้ำๆ อะไรบ้าง

แนวคิดเชิงคำนวณ

3) Algorithm ชื่อไทย “ขั้นตอนวิธี”

Algorithm คือลำดับขั้นตอนในการแก้ปัญหาหรือการทำงานที่ชัดเจน การคิดค้น อธิบายขั้นตอนวิธีในการแก้ปัญหาต่าง ๆ

☞ ตัวอย่างการนำแนวคิดนี้ไปใช้ตอนเขียนโปรแกรม

สำหรับคนเขียนโปรแกรม คงรู้จักกันดีไม่ต้องอธิบายมาก เช่น
-จะคำนวณหาพื้นที่เส้นรอบวง ต้องมีสเตปคำนวณอย่างไรบ้าง
-จะค้นหาข้อมูลแบบ binary search ต้องมีขั้นตอน 1,2,3 อย่างไรบ้าง
-จะหาเส้นทางที่ใกล้สุดในกราฟ ด้วยวิธี Dijkstra จะมีขั้นตอน 1,2,3 อย่างไรบ้าง

☞ ตัวอย่างการนำไปใช้นอกจากเขียนโปรแกรม

-จะเต้นเพลงคุกกี้เสี่ยงทาย ต้องมีเสตป 1, 2, 3 อย่างไร?
-จะวางแผนจีบสาว มีขั้นตอนอย่างไร?
-จะไปเที่ยวเขาใหญ่ ต้องวางแผนว่าในแต่ละวันทำอะไรบ้าง เที่ยวไหน กินข้าวที่ไหน มีลำดับตามช่วงเวลา?

แนวคิดเชิงคำนวณ

4) Abstract thinking ชื่อไทย “การคิดเชิงนามธรรม”

Abstract thinking เป็นกระบวนการคัดแยกคุณลักษณะที่สำคัญออกจากรายละเอียดปลีกย่อย ในปัญหา หรืองานที่กำลังพิจารณา เพื่อให้ได้ข้อมูลที่จำเป็นและเพียงพอในการแก้ปัญหา

☞ ตัวอย่างการนำแนวคิดนี้ไปใช้ตอนเขียนโปรแกรม

-จากโจทย์ปัญหาเขียนโปรแกรมที่ดูยุ่งยาก สามารถทำให้ง่ายขึ้นด้วยการสกัดเอาลักษณะสำคัญออกมาวาดเป็น Object ใช้ Class diagram ลากเส้นแสดงความสัมพันธ์กัน จากนั้นก็เริ่มเขียนโปรแกรมเป็นแบบเชิงวัตถุ เป็นต้น

-ถ้าเราจะส่งข้อมูลข้าม network แล้วเขียนโปรแกรมหาระยะทางสั้นที่สุดต้องทำอย่างไร? วิธีคิดก็จะสกัดรายละเอียดสำคัญออกมา เช่น server ก็วาดเป็นโหนด แล้วมีเส้นเชื่อมระหว่างโหนด พร้อมระบุระยะทางบนเส้น พอคิดแบบเชิงนามธรรมได้แล้ว ก็จะได้ง่ายมากที่จะเอาทฤษฎีกราฟมาคำนวณหาระยะทางสั้นที่สุด เป็นต้น

☞ ตัวอย่างการนำไปใช้นอกจากเขียนโปรแกรม

- เราจะดูแผนที่ประเทศไทย เพื่อเที่ยวภาคเหนือ ถ้าดูเต็มรูปแบบ จะยุ่งยาก งงตาลาย มีหลายเส้นทางเยอะไปหมด แต่เราสามารถแก้ปัญหา โดยตัดรายละเอียดส่วนเกินทิ้ง เอาสถานที่และเส้นทางที่สำคัญที่จะใช้เดินทาง มาวาดใส่กระดาษก็พอ

ขั้นตอนวิธี (algorithm)

อัลกอริทึม มีที่มาจากชื่อของนักคณิตศาสตร์ชาวเปอร์เซียในยุคศตวรรษที่ ๙ ชื่ออะบู อับดิลลาห์ มูฮัมหมัด บิน มูซา อัลคอวาริซมีย์ (Abu Abdillah Muhammad binMusa al-Khawarizmi) คำว่าอัลคอวาริซมีย์ (al-Khawarizmi)ได้เพี้ยนเป็นอัลกอริทมี (Algoritmi)เมื่องานเขียนของเขาได้รับการแปลเป็นภาษาละติน แล้วกลายเป็นอัลกอริทึม (Algorithm) ซึ่งหมายถึงกฎที่ใช้ในการคิดคำนวณเลขคณิตในช่วงศตวรรษที่ 18 ในปัจจุบันคำนี้ได้มีความหมายที่กว้างขึ้นโดยหมายรวมถึงขั้นตอนวิธีการในการแก้ปัญหาต่าง ๆ มีนักการศึกษา นักวิชาการ ให้ความหมายของคำว่า อัลกอริทึม (Algorithm) ไว้หลายท่านดังนี้

ราชบัณฑิตยสถานได้บัญญัติศัพท์คำว่า “Algorithm” ใช้คำว่า “ขั้นตอนวิธี” หมายถึง ขั้นตอนวิธีการแก้ปัญหาเชิงคำนวณด้วยคอมพิวเตอร์

อัลกอริทึม (Algorithm) หมายถึง ขั้นตอนหรือลำดับการประมวลผลในการแก้ปัญหาใดปัญหาหนึ่งซึ่งจะช่วยให้ผู้พัฒนาโปรแกรมเห็นขั้นตอนการเขียนโปรแกรมอย่างง่ายขึ้น

อัลกอริทึม (Algorithm) หมายถึง แนวคิดอย่างมีเหตุผลที่ผู้เขียนโปรแกรมใช้ในการอธิบายวิธีการอย่างเป็นขั้นตอนตามลำดับในการที่จะพัฒนาโปรแกรมนั้น ๆ เพื่อตรวจสอบขั้นตอนต่าง ๆ ในการทำงานและความถูกต้องในแต่ละขั้นตอน

อัลกอริทึม (Algorithm) คือ กระบวนการแก้ปัญหาที่สามารถอธิบายออกมาเป็นขั้นตอนที่ชัดเจน เมื่อนำเข้าอะไร แล้วจะต้องได้ผลลัพธ์เช่นไร กระบวนการนี้ประกอบด้วยจะประกอบด้วย วิธีการเป็นขั้นๆ และมีส่วนที่ต้องทำแบบวนซ้ำอีก จนกระทั่งเสร็จสิ้นการทำงาน (http://mindphp.com)

ขั้นตอนวิธี (algorithm)

ซึ่งในการเริ่มต้นในการเขียนโปรแกรมอัลกอริทึมมีความสำคัญเป็นอย่างมากเพราะ เป็นการจัดลำดับขั้นตอนวิธีการแก้ปัญหาหรือจัดการความคิดให้เป็นขั้นตอนต่าง ๆ เพื่อแก้ไขปัญหาในขั้นตอนการเขียนโปรแกรมที่สอดคล้องกรรมวิธีแก้ปัญหาที่กำหนดไว้ การเขียนอัลกอริทึมจึงเป็นการแสดงลำดับการทำงานตามคุณสมบัติด้านการประมวลผลของคอมพิวเตอร์ ที่พร้อมจะนำไปแปลงเป็นลำดับคำสั่งให้คอมพิวเตอร์ทำงาน การเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาที่เหมาะสมเพื่อสั่งให้คอมพิวเตอร์ทำงานตามอัลกอริทึมที่กำหนดไว้ และการเขียนอัลกอริทึมออกมาให้ตรวจสอบความถูกต้องได้ครบถ้วนขึ้น

หลักการเขียนอัลกอริทึม

- กระบวนการสำคัญเริ่มต้นที่จุดจุดเดียวในการมีจุดเริ่มต้นหลายที่จะทำให้กระบวนการวิธีสับสน จนในที่สุดอาจทำให้ผลลัพธ์ที่ได้ไม่ตรงกับความต้องการ หรืออาจทำให้อัลกอริทึมนั้นไม่สามารถทำงานได้เลย
- กำหนดการทำงานเป็นขั้นเป็นตอนอย่างชัดเจน การกำหนดอัลกอริทึมที่ดีควรมีขั้นตอนที่ชัดเจนไม่คลุมเครือ เสร็จจากขั้นตอนหนึ่ง ไปยังขั้นตอนที่สองมีเงื่อนไขการทำงานอย่างไรควรกำหนดให้ชัดเจน
- การทำงานแต่ละขั้นตอนควรสั้นกระชับ เพราะการกำหนดขั้นตอนการทำงานให้สั้นกระชับนอกจากจะทำให้โปรแกรมทำงานได้รวดเร็วแล้ว ยังเป็นประโยชน์ต่อผู้อื่นที่มาพัฒนาโปรแกรมต่อด้วยเพราะสามารถศึกษาอัลกอริทึมจากโปรแกรมที่เขียนไว้ได้ง่าย
- ผลลัพธ์ในแต่ละขั้นตอนควรต่อเนื่องกัน การออกแบบขั้นตอนที่ดีนั้นผลลัพธ์จากขั้นตอนแรกควรเป็นข้อมูลสำหรับนำเข้า ให้กับข้อมูลในขั้นต่อไป ต่อเนื่องกันไปจนกระทั่งได้ผลลัพธ์ตามที่ต้องการ
- การออกแบบอัลกอริทึมที่ดี ควรออกแบบให้ครอบคลุมการทำงานในหลายรูปแบบ เช่น การออกแบบโดยคิดไว้ล่วงหน้าว่าหากผู้ใช้โปรแกรมป้อนข้อมูลเข้าผิดประเภท โปรแกรมจะมีการเตือนว่าผู้ใช้งานมีการใส่ข้อมูลที่ผิดประเภทโดยโปรแกรมจะไม่รับข้อมูลนั้น เพื่อให้ใส่ข้อมูลใหม่อีกครั้ง เพื่อป้องกันการเกิดจุดบกพร่องของโปรแกรมได้

ขั้นตอนวิธี (algorithm)

ประโยชน์ของอัลกอริทึม

ประโยชน์ของอัลกอริทึม (Algorithm) คือ ทำให้ไม่สับสนกับวิธีดำเนินงาน เพราะทุกอย่างจะถูกจัดเรียงเป็นขั้นตอนมีวิธีการและทางเลือกไว้ให้ เมื่อนำมาใช้จะทำให้การทำงานสำเร็จอย่างรวดเร็ว ทำให้ปัญหาลดลงหรือสามารถค้นหาต้นเหตุของปัญหาได้อย่างรวดเร็ว เนื่องจากกระบวนการถูกแยกแยะกิจกรรม ขั้นตอน และความสัมพันธ์ ออกมาให้เห็นอย่างชัดเจน

รูปแบบการเขียนอัลกอริทึม

การเขียนอัลกอริทึมมีหลายรูปแบบ โดยผู้เขียนสามารถใช้อัลกอริทึมหลายรูปแบบประกอบกันในการออกแบบอัลกอริทึมนั้นเพื่อใช้ในการแก้ปัญหาการเขียนโปรแกรมได้

รูปแบบการเขียนอัลกอริทึม

1.แบบลำดับ (Sequential) มีลักษณะการทำงานจะเป็นไปตามขั้นตอน ก่อน-หลัง ต่อเนื่องกันไปเป็นลำดับ โดยการทำงานแต่ละขั้นตอนต้องทำให้เสร็จก่อน แล้วจึงไปทำขั้นตอนต่อไป

ตัวอย่างเช่น อัลกอริทึมแบบลำดับ การทอดไข่เจียว

1. เริ่มต้น
2. หยิบไข่ไก่
3. ตอกไข่ไก่ใส่ภาชนะ
4. ปรุงรส ด้วยเครื่องปรุง
5. ตีไข่ด้วยช้อนส้อม
6. ตั้งกระทะบนเตา
7. เปิดแก๊ส และติดไฟ
8. ใส่น้ำมันพืช
9. นำไข่ที่ปรุงรสแล้วใส่ลงในกระทะที่ร้อน
10. ทอดจนสุก
11. ตักขึ้นใส่จานที่เตรียมไว้
12. จบการทำงาน

รูปแบบการเขียนอัลกอริทึม

2.แบบทางเลือก (Decision) อัลกอริทึมรูปแบบนี้ มีเงื่อนไขเป็นตัวกำหนดเส้นทางการทำงานของกระบวนการแก้ปัญหา โดยตัวเลือกนั้นอาจจะมีตั้งแต่ 2 ตัวขึ้นไป เช่น สอบข้อเขียน คะแนนเต็ม 50 ได้คะแนน 30 สอบผ่าน ถ้าต่ำกว่า 30 สอบไม่ผ่าน

ตัวอย่างการเขียนอัลกอริทึมแบบทางเลือก อัลกอริทึมตัดเกรดวิชาคอมพิวเตอร์

1. เริ่มต้น
2. คะแนนสอบของนักเรียน
3. ตรวจสอบคะแนน (คะแนนที่สอบผ่าน 50 คะแนน)
4. ถ้ามากกว่า 50 คะแนน สอบผ่าน
5. ถ้าน้อยกว่า 50 คะแนน สอบตก
6. ประกาศผล
7. จบการทำงาน

รูปแบบการเขียนอัลกอริทึม

3.แบบทำซ้ำ (Repetition) อัลกอริทึมแบบนี้คล้ายกับแบบทางเลือก คือ มีการตรวจสอบเงื่อนไข แต่แตกต่างกันตรงที่เมื่อการทำงานตรงตามเงื่อนไขที่กำหนด โปรแกรมจะกลับไปทำงานอีกครั้งวนการทำงานแบบนี้เรื่อย ๆ จนกระทั่งไม่ตรงกับเงื่อนไขที่กำหนดไว้จึงหยุดการทำงานหรือทำงานในขั้นต่อไป

ตัวอย่างการเขียนอัลกอริทึมแบบทำซ้ำ อัลกอริทึมการซื้อมังคุด 1 กิโลกรัม

1. เริ่มต้น
2. หยิบถุงพลาสติก
3. หยิบมังคุดมาเลือก โดยกดที่เปลือกที่นิ่มๆ
4. ตรวจสอบเงื่อนไข (น้อยกว่า 1 กิโลกรัม)
5. ถ้าจริง เลือกมังคุดต่อ
6. ถ้าเท็จ หยุดเลือก
7. จ่ายเงินให้กับผู้ที่ขาย
8. จบการทำงาน

คุณลักษณะการเขียนอัลกอริทึม

ในการแก้ปัญหาแต่ละปัญหามหลายวิธี ดังนั้นการเขียนอัลกอริทึมเพื่อแก้ปัญหาแต่ละปัญหาก็มีหลายวิธีด้วย แต่ละวิธีมีทั้งข้อเด่นข้อด้อย ดังนั้นต้องเลือกให้เหมาะสมกับงานและสภาพแวดล้อมในขณะนั้น โดยทั่วไปอัลกอริทึมที่ดี ต้องคุณลักษณะดังต่อไปนี้

- มีความถูกต้อง ความถูกต้องเป็นคุณสมบัติข้อแรกที่จะต้องพิจารณา นั่นคือเมื่อทำงานตามอัลกอริทึม แล้วจะต้องได้ผลลัพธ์ที่ถูกต้อง ซึ่งถ้าผลลัพธ์ที่ได้จากอัลกอริทึมไม่ถูกต้อง จะถือว่าไม่ใช่อัลกอริทึมที่ดี โดยที่ไม่จำเป็นต้องพิจารณาคุณสมบัติข้ออื่น ๆ

- ใช้เวลาในการปฏิบัติงานน้อยที่สุด

- สั้น กระชับ มีเฉพาะขั้นตอนที่จำเป็นเท่านั้น

- ใช้เนื้อที่ในหน่วยความจำน้อยที่สุด เนื้อที่ในหน่วยความจำจะถูกใช้สำหรับเก็บค่าของตัวแปร และเก็บคำสั่งที่ใช้ในการทำงาน ดังนั้น ถ้าอัลกอริทึมยาวเกินความจำเป็น จะทำให้ใช้เนื้อที่มาก และ ถ้ามีตัวแปรมากเกินความจำเป็น ก็จะทำให้เสียเนื้อที่ในหน่วยความจำไปด้วย

- มีความยืดหยุ่นในการใช้งาน

- ใช้เวลาในการพัฒนาน้อยที่สุด เมื่อนำอัลกอริทึมไปแปลงเป็นโปรแกรมภาษาคอมพิวเตอร์แล้วจะต้องใช้เวลาน้อยที่สุด

- ง่ายต่อการทำความเข้าใจ

การวิเคราะห์อัลกอริทึม

การวิเคราะห์อัลกอริทึม

การวิเคราะห์ขั้นตอนวิธีการ ต้องทำการแยกแยะระบบว่าเป็นข้อมูลเข้า หรือออกดังนี้

อะไรเป็นข้อมูลเข้า (Input)

วิธีการประมวลผลที่จะนามาซึ่งคาตอบ (Process)

อะไรเป็นข้อมูลออก (Output)

ตัวอย่างที่ 1 ต้องการหาค่าเฉลี่ยของคะแนนวิชา Fundamental of Computer ของนักศึกษาจานวน 50 คน

วิเคราะห์อัลกอริทึม

ข้อมูลเข้า : คะแนนของนักศึกษาแต่ละคน (Score)

ประมวลผล : ผลรวมของคะแนนนักศึกษาทุกคน ธ 50 (Average)

ข้อมูลออก : พิมพ์ผลลัพธ์

ตัวอย่างที่ 2 จงคำนวณหาพื้นที่สามเหลี่ยม โดยให้ผู้ใช้สามารถ Input ข้อมูลความสูงและความยาวฐานได้

วิเคราะห์อัลกอริทึม

ข้อมูลเข้า : ความสูง (H) , ความยาวฐาน (B)

ประมวลผล : คำนวณ Area = ฝ * B*H

ข้อมูลออก : แสดงค่าพื้นที่สามเหลี่ยมที่หน้าจอ

เครื่องมือช่วยในการเขียนอัลกอริทึม

การเขียนอัลกอริทึม เป็นการวางแผนเกี่ยวกับการแก้ปัญหา โดยจะอธิบายการทำงานที่ชัดเจน เพื่อเป็นแนวทางในการเขียนโปรแกรม ช่วยให้การเขียนโปรแกรมทำได้ง่ายขึ้น ช่วยให้โปรแกรมมีข้อผิดพลาดน้อยลง นอกจากนี้ยังช่วยตรวจสอบการทำงานของโปรแกรม ทำให้ทราบขั้นตอนการทำงานของโปรแกรมได้อย่างรวดเร็ว โดยไม่ต้องดูจากโปรแกรมจริง

ในการเขียนอัลกอริทึม มีเครื่องมือช่วยในการเขียนที่นิยมใช้ 3 แบบ คือ

1. บรรยาย (narrative description)

2. ผังงาน (flowchart)

3. รหัสลำลอง (pseudo code)

การเขียนอัลกอริทึม

1. การเขียนอัลกอริทึมแบบบรรยาย เป็นการแสดงขั้นตอนการทำงานในลักษณะการบรรยายเป็นข้อความด้วยภาษาพูดใด ๆ เช่น ภาษาไทย ภาษาอังกฤษ ภาษาเกาหลี ภาษาญี่ปุ่น หรือภาษาจีน เป็นต้น ขึ้นอยู่กับความถนัดของผู้เขียนอัลกอริทึม มักเขียนบรรยายขั้นตอนการทำงานเป็นข้อๆ เช่น

ตัวอย่าง การปลูกต้นไม้ แสดงขั้นตอนการทำงานด้วยอัลกอริทึมแบบบรรยายได้ดังนี้

1. เริ่มต้น
2. ขุดหลุม
3. ใส่ปุ๋ย
4. นำต้นไม้ลงหลุม
5. กลบดิน
6. ปักหลักยึดต้นไม้
7. รดน้ำ
8. จบการทำงาน

ข้อดีของการเขียนอัลกอริทึมแบบบรรยาย

การเขียนอัลกอริทึมแบบบรรยาย มีข้อดี คือ ง่านในการเขียนบรรยาย เนื่องจากใช้ภาษาพูดที่ผู้เขียนอัลกอริทึมคุ้นเคยอยู่แล้ว ดังนั้นจึงง่ายในการเขียนบรรยาย

ข้อเสียของการเขียนอัลกอริทึมแบบบรรยาย

เนื่องจากการเขียนมีลักษณะบรรยาย มีรายละเอียดดังนี้

ขอบเขตของการบรรยายกว้างเกินไปยืดเยื้อเกินไป

ยากต่อความเข้าใจ

ยากในตรวจสอบความถูกต้อง

ยากในการแปลงเป็นโปรแกรม

การเขียนอัลกอริทึม

2. การเขียนอัลกอริทึมแบบผังงาน การเขียนอัลกอริทึมแบบผังงานจะแสดงขั้นตอนการทำงานในลักษณะของรูปภาพหรือสัญลักษณ์ ซึ่งเป็นสัญลักษณ์ที่เป็นมาตรฐาน ไม่อ้างอิงภาษาใดภาษาหนึ่ง ทำให้เห็นลำดับการทำงานก่อนหลังได้ชัดเจน เช่น

ตัวอย่าง การปลูกต้นไม้ แสดงขั้นตอนการปลูกต้นไม้ด้วยผังงาน ดังภาพต่อไปนี้

ภาพผังงานแสดงขั้นตอนการปลูกต้นไม้

ที่มา : http://algorithm-elearning.esy.es/

ข้อดีของการเขียนอัลกอริทึมแบบผังงาน

1. ทำความเข้าใจได้ง่าย
2. ตรวจสอบความถูกต้องได้ง่าย
3. พัฒนาโปรแกรมได้ง่าย
4. ง่ายต่อการบำรุงรักษาโปรแกรม

การเขียนอัลกอริทึม

3. การเขียนอัลกอริทึมโดยใช้รหัสลำลอง เป็นการเขียนขั้นตอนการทำงานในลักษณะของคำอธิบายที่มีรูปแบบโครงสร้างชัดเจน ไม่ขึ้นกับภาษาใดภาษาหนึ่ง แต่สามารถเปลี่ยนเป็นภาษาคอมพิวเตอร์ได้ง่าย ดังนั้นโครงสร้างส่วนใหญ่จึงนิยมใช้คำสั่งเฉพาะที่มีอยู่ในคอมพิวเตอร์เพื่อแทนการทำงานต่าง ๆ เช่น Read if Case หรือ While/Do เป็นต้น การเขียนอัลกอริทึมมีลักษณะดังต่อไปนี้

1. รูปแบบเป็นภาษาพูดง่าย ๆ ภาษาอังกฤษ หรือภาษาไทยก็ได้
2. ไม่มีกฎที่แน่นอนตายตัว แต่ลักษณะคล้ายกับภาษาคอมพิวเตอร์
3. ไม่เจาะจงภาษาคอมพิวเตอร์ใดภาษาหนึ่ง
4. ใช้คำเฉพาะ (Keyword)
5. เริ่มต้น คือคำสั่งที่อยู่บรรทัดแรก และ สิ้นสุด คือ คำสั่งที่อยู่บรรทัดสุดท้าย
6. ใช่ย่อหน้าในการเขียนการทำงานย่อยที่อยู่ภายใน
7. ไม่ระบุอุปกรณ์ในการรับข้อมูลหรือแสดงผลข้อมูล
8. ข้อมูลต่าง ๆที่ใช้ควรจะอยู่ในรูปของตัวแปร

ลักษณะที่ดีของรหัสเทียม การเขียนรหัสเทียมที่ดี จะต้องมีลักษณะดังนี้

1. ชัดเจน
2. สั้น กระชับและได้ใจความ
3. รูปแบบแน่นอนกะทัดรัด และมองคล้ายภาษาคอมพิวเตอร์ระดับสูง
4. แปลเป็นภาษาคอมพิวเตอร์ได้เร็ว

รูปแบบ Algorithm แบบรหัสเทียม

<ชื่อของอัลกอริทึม>
START
1……
2……
3……
END

การเขียนอัลกอริทึม

ตัวอย่างโจทย์

จงเขียนโปรแกรมคำนวณหาพื้นที่ สามเหลี่ยมทั่วไป โดยให้โปรแกรมมีการรับค่า ฐาน และ สูงจากทาง Keyboard จากนั้นแสดงผลลัพธ์ของพื้นที่ออกทางจอภาพ

วิเคราะห์โจทย์

แบ่งออกเป็นสองส่วนคือ ส่วนในการรับค่า คือรับค่า ฐาน และ สูงจากผู้ใช้มาเก็บไว้ในตัวแปร และส่วนในการคำนวณ สูตรในการหาพื้นที่สามเหลี่ยมคือ (1/2) * ฐาน * สูง

ตัวอย่าง การเขียนอัลกอริทึม คำนวณหาพื้นที่สามเหลี่ยมอัลกอริทึม (Algorithm) การหาพื้นที่สามเหลี่ยม

1. เริ่มต้น
2. รับค่าความยาวของฐานมาเก็บในตัวแปร BASE
3. รับค่าความยาวของสูงมาเก็บในตัวแปร HEIGHT
4. คำนวณหาพื้นที่ AREA = 0.5 * BASE*HEIGHT
5. แสดงผลพื้นที่
6. จบ

ซูโดโค้ด (Pseudo codes) Algorithm Triangle

1. START
2. READ BASE
3. READ HEIGHT
4. AREA = 0.5 * BASE * HEIGHT
5. PRINT AREA
6. END

ที่มา

- หนังสือเรียน รายวิชาพื้นฐานวิทยาศาสตร์ เทคโนโลยี (วิทยาการคำนวณ) ชั้นมัธยมศึกษาปีที่ 4 ของสถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี