

## Algorithms

### วัตถุประสงค์

1. เพื่อพัฒนาทักษะในการวิเคราะห์ปัญหาของนักศึกษา
2. เพื่อให้นักศึกษาสามารถแก้ปัญหาทางคอมพิวเตอร์ได้

### วิธีการเรียนการสอน

บรรยาย การบ้าน ศึกษาคนคว้าด้วยตนเอง

### การวัดผล

การบ้าน สอบกลางภาค



ภาพการแก้ปัญหา (ที่มา <http://www.hc-sc.gc.ca/>)

### Problem Solving

การแก้ปัญหา คือ การค้นหาคำตอบของปัญหาเพื่อไปสู่จุดมุ่งหมายใดจุดมุ่งหมายหนึ่ง การแก้ปัญหาที่มีประสิทธิภาพมีหลายวิธี ขอเสนอวิธีหนึ่งเรียกว่า IDEAL (Anita E. Woolfolk 1993) ดังต่อไปนี้

**I** : Identify the Problem คือ ทำปัญหาให้กระจ่างชัด โดยการวิเคราะห์ว่าอะไรกันแน่ที่เป็นปัญหาที่ต้องแก้

**D** : Defining and Representing the Problem คือ การค้นหาข้อมูลรายละเอียดที่สำคัญของปัญหา แล้วคัดเลือกข้อมูลสำคัญที่จะนำมาใช้ในการแก้ปัญหา

**E** : Exploring Possible Solution Strategies คือ ค้นหาวิธีการแก้ปัญหาที่อาจเป็นไปได้

**A** : Acting on the Strategies คือการลงมือแก้ปัญหาดตามวิธีที่เลือกไว้

**L** : Look Back and Evaluated the Effect of Your Activities คือ พิจารณาผลการแก้ปัญหาว่าเป็นอย่างไร สำเร็จหรือไม่ ควรปรับปรุงแก้ไขอย่างไร

**ปัญหาที่ 1**

ชายคนหนึ่งต้องการพา หมาป่า และ ผัก ข้ามแม่น้ำ โดยใช้เรือข้ามไปยังอีกฝั่งหนึ่ง โดยเรือที่พาข้ามสามารถบรรทุกได้ครั้งละสองสิ่ง โดยถ้าหมาป่าอยู่ตามลำพังกับแกะหมาป่าจะกินแกะ ถ้าแกะอยู่ตามลำพังกับผักแกะจะกินผัก เราจะช่วยชายคนนี้ให้สามารถพา หมาป่า และ ผักข้ามแม่น้ำให้ปลอดภัยทั้งหมดได้อย่างไร

**ปัญหา** ช่วยชายคนหนึ่งพาหมาป่า และ ผักข้ามแม่น้ำให้ปลอดภัย

**เงื่อนไข** เรือที่พาข้ามสามารถบรรทุกได้ครั้งละสองสิ่ง,

ถ้าหมาป่าอยู่ตามลำพังกับแกะหมาป่าจะกินแกะ, ถ้าแกะอยู่ตามลำพังกับผักแกะจะกินผัก

**ข้อมูลที่รับเข้า** ชายหนึ่งคน หมาป่าหนึ่งตัว แกะหนึ่งตัว และผัก อยู่บนฝั่งหนึ่งของแม่น้ำ

**ผลลัพธ์** ชายหนึ่งคน หมาป่าหนึ่งตัว แกะหนึ่งตัว และผัก ข้ามไปอยู่อีกฝั่งหนึ่งของแม่น้ำ

**ขั้นตอนการแก้ปัญหา** ???

**ปัญหาที่ 2**

มีตัวเลขจำนวนเต็มชุดหนึ่ง จงหาตัวเลขที่มีค่าน้อยที่สุด

**ปัญหา** หาค่าน้อยที่สุดจากเลขจำนวนเต็ม  $n$  ตัว

**เงื่อนไข** เลขจำนวนเต็มที่ต้อง น้อยกว่าหรือเท่ากับเลขจำนวนเต็มทุกตัวที่รับเข้ามา

**ข้อมูลที่รับเข้า** เลขจำนวนเต็ม  $n$  ตัว

**ผลลัพธ์** เลขจำนวนเต็มที่มีค่าน้อยที่สุดจากเลขจำนวนเต็ม  $n$  ตัว

**ขั้นตอนการแก้ปัญหา** ???

**ปัญหาที่ 3**

มีกระดาษกว้าง 8 ซม. ยาว 12 ซม. จะสามารถตัดเป็นรูป 4 เหลี่ยมจัตุรัส ขนาดใหญ่ที่สุดได้ทั้งหมดกี่รูป

**ปัญหา** หาจำนวน 4 เหลี่ยมจัตุรัส ที่มีขนาดใหญ่ที่สุดที่เป็นไปได้

**เงื่อนไข** ความกว้างของรูปสี่เหลี่ยมจะต้องเท่ากับความยาว

**ข้อมูลที่รับเข้า** จำนวนเลขจำนวนเต็ม 2 จำนวน เป็น ความกว้างและความยาวของกระดาษ

**ผลลัพธ์** จำนวน 4 เหลี่ยม (เลขจำนวนเต็ม) ขนาดใหญ่ที่สุดที่สามารถตัดได้

**ขั้นตอนการแก้ปัญหา** ???

Greatest Common Divisor (GCD)

Euclid's GCD, [300 B.C.]

$$\text{gcd}(m, n) = \text{gcd}(n, m \bmod n)$$

$$\text{เช่น } \text{gcd}(60, 24) = \text{gcd}(24, 12) = \text{gcd}(12, 0) = 12$$

ตัวหารร่วมมากของ Euclid

ขั้นตอนการแก้ปัญหา → Algorithm

**Algorithm** คือ กระบวนการแก้ปัญหาที่สามารถอธิบายออกมาเป็นขั้นตอนที่ชัดเจน โดยสามารถรับ Input ที่ได้นิยามไว้อย่างดีและทำการเปลี่ยนเป็น Output ที่ถูกต้อง ในเวลาที่จำกัดได้อย่างมีประสิทธิภาพ

สูตรทำกล้วยทอด เป็น Algorithm ???

สูตรทำกล้วยทอด

1. นำกล้วยมาปอกเปลือกและหั่นตามยาวเป็นชิ้นบางๆ หนึ่งลูกควรหั่นให้ได้ 3 ชิ้น
2. นำแป้งข้าวเจ้า, แป้งสาลี, เกลือ, ผงฟู, น้ำตาลปีบ, งามาว, มะพร้าวขูด, น้ำปูนใสและหัวกะทิ ผสมเข้าด้วยกันในชามขนาดใหญ่ คนจนแป้งและน้ำตาลละลายดี และส่วนผสมทั้งหมดเข้ากัน เป็นเนื้อเดียว
3. ใส่น้ำมันลงในกระทะ และนำไปตั้งบนไฟค่อนข้างแรง รอจนน้ำมันเดือด จึงใส่ใบเตยลงไปทอด ก่อนให้น้ำมันหอม
4. นำกล้วยที่หั่นเตรียมไว้ในขั้นตอนที่หนึ่งชุบแป้งแล้วนำไปลงทอดจนเหลืองสุกและกรอบ จึงตักออกมาสะเด็ดน้ำมัน
5. เรียงจัดใส่จาน และเสิร์ฟเป็นของว่างทานเล่น

สูตรทำกล้วยทอด

**ลักษณะของ Algorithms**

1. ต้องมีการกำหนด input ที่ชัดเจน
2. algorithm หนึ่งๆสามารถแสดงได้หลายวิธี เช่น Pseudocode, Flowchart เป็นต้น
3. อาจมีหลาย algorithm ที่สามารถใช้ในการแก้ปัญหาเดียวกัน
4. หลาย algorithms ที่ใช้ในการแก้ปัญหาเดียวกัน สามารถมี ideas ที่ต่างกันมากมายและแต่ละ ideas ก็อาจมีประสิทธิภาพหรือความเร็วในการแก้ปัญหาที่ต่างกัน

```
function gcd(a, b)
    while b ≠ 0
        t ← b
        b ← a mod b
        a ← t
    return a
```

ตัวอย่าง Euclid's algorithm

“Algorithm ไม่ใช่คำตอบแต่เป็นชุดคำสั่งที่ทำให้ได้คำตอบ”

### ขั้นตอนในการออกแบบและวิเคราะห์ Algorithm (Algorithm Design and Analysis Process)

1. ทำความเข้าใจปัญหาให้ครบถ้วน และคิดถึงกรณีพิเศษต่างๆ ของปัญหา ระบุช่วงของ instance ซึ่ง algorithm จำเป็นต้องใช้เป็น input ในการแก้ปัญหา

“Algorithm ที่ถูกต้อง ต้องทำงานได้ถูกต้องสำหรับทุกๆ input ที่กำหนดไว้”

2. เลือกการแก้ปัญหาโดยการประมาณหรือการแก้ปัญหาแบบแน่นอน (Exact & Approximate Problem Solving)

“ทำไมต้องใช้ Approximate Algorithms

บางปัญหาไม่สามารถแก้แบบแน่นอนได้ เช่น Square Root

บางครั้งการแก้ปัญหาแบบแน่นอนอาจใช้เวลานานมากจนรับไม่ได้”

3. เลือกโครงสร้างข้อมูลที่เหมาะสมกับ Algorithm
4. กำหนดวิธีการในการอธิบาย Algorithm

- Natural Language

- Pseudocode

- Flowchart

5. พิสูจน์ว่า Algorithm จะมีความถูกต้อง (ทำงานถูกต้องทุก input ที่กำหนดในเวลาที่ได้รับได้) สำหรับ Approximation Algorithm, Correctness คือ error ที่ได้จาก Algorithm จะต้องไม่มากกว่าค่าที่กำหนดไว้

6. การวิเคราะห์ Algorithm

ประสิทธิภาพ

- Time Efficiency

- Space Efficiency

Simplicity ง่ายในการอ่านของ user

Generality

7. การ Coding