



Chapter 1

Introduction to Data Structures and Algorithms

จุดประสงค์

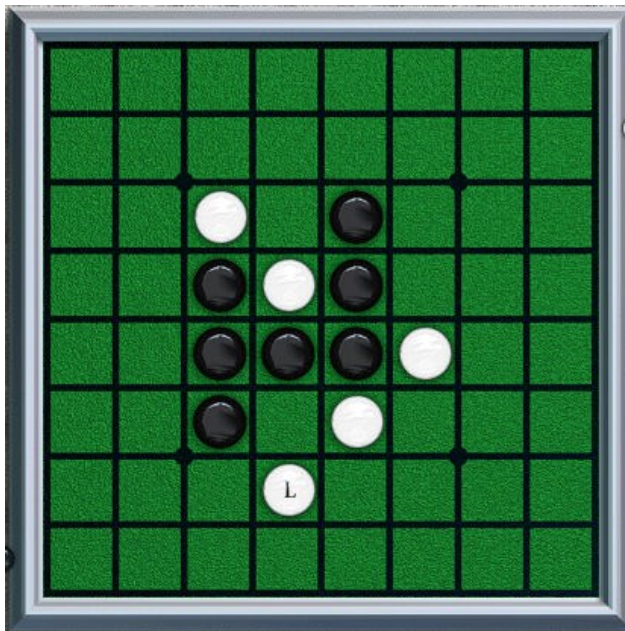
- เข้าใจความสำคัญของโครงสร้างข้อมูลและอัลกอริทึม
- เข้าใจแนวคิดของชนิดข้อมูลและโครงสร้างข้อมูล

ทำไมต้องเรียนวิชานี้ ?

- เพื่อนำไปใช้เพิ่มประสิทธิภาพในการพัฒนาโปรแกรมเพื่อแก้ปัญหาที่ซับซ้อน
 - ทำงานได้รวดเร็ว
 - ประหยัดทรัพยากร
- ประเด็นที่ต้องพิจารณา
 - โครงสร้างข้อมูล : ข้อมูลที่ใช้นั้นถูกจัดเก็บและมีความสัมพันธ์อย่างไร
 - อัลกอริทึม : ขั้นตอนวิธีในการแก้ปัญหา

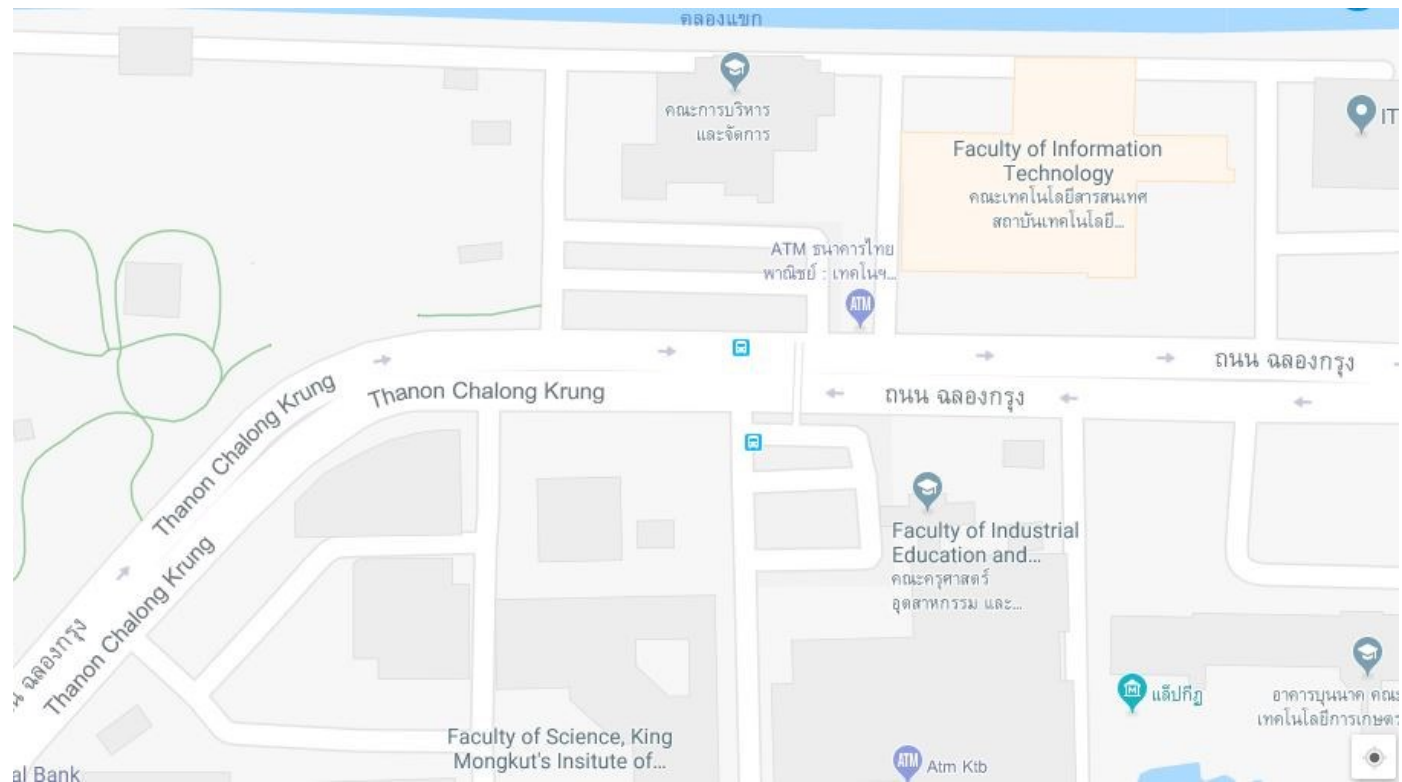
โครงสร้างข้อมูล

- ข้อมูลในโปรแกรมคอมพิวเตอร์ มีการจัดเก็บอย่างไร



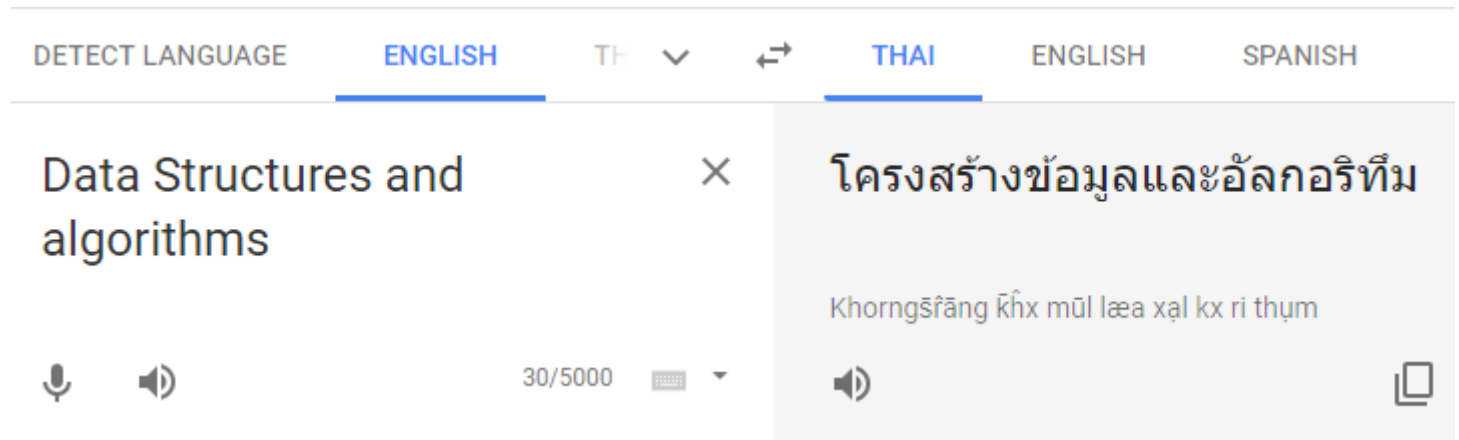
โครงสร้างข้อมูล

- ข้อมูลในโปรแกรมคอมพิวเตอร์ มีการจัดเก็บอย่างไร



โครงสร้างข้อมูล

- ข้อมูลในโปรแกรมคอมพิวเตอร์ มีการจัดเก็บอย่างไร



อัลกอริทึม

- ขั้นตอนวิธีหรือลำดับการทำงานเพื่อใช้ในการแก้ปัญหาหนึ่งๆ
 - Google Maps : แนะนำเส้นทางได้อย่างไร
 - Google Translate : แปลภาษาอย่างไร
- ปัญหาหนึ่งๆ อาจมีอัลกอริทึมในการแก้ปัญหาได้หลายแบบ
 - มีวิธีแก้ปัญหาแบบใดได้บ้าง
 - วิธีการแก้ปัญหาที่ดี มีประสิทธิภาพเป็นอย่างไร
 - Ex : https://www.youtube.com/watch?v=cVMKXKoGu_Y

ตัวอย่างความสัมพันธ์

❖ การจัดหนังสือในห้องสมุด

- ข้อมูล -> หนังสือ
- โครงสร้างข้อมูล -> เก็บข้อมูลอย่างไร
 -
 -
- อัลกอริทึม -> หาหนังสืออย่างไร
 -
 -

ตัวอย่างความสัมพันธ์

❖ การจัดหนังสือในห้องสมุด

- ข้อมูล -> หนังสือ
- โครงสร้างข้อมูล -> เก็บข้อมูลอย่างไร
 - แบบ 1 เก็บหนังสือตามลำดับที่ได้รับ
- อัลกอริทึม -> หาหนังสืออย่างไร
 - แบบ 1 ไล่หาจากเล่มแรกที่มีการลงทะเบียน

ตัวอย่างความสัมพันธ์

❖ การจัดหนังสือในห้องสมุด

- ข้อมูล -> หนังสือ
- โครงสร้างข้อมูล -> เก็บข้อมูลอย่างไร
 - แบบ 1 เก็บหนังสือตามลำดับที่ได้รับ
 - แบบ 2 เก็บหนังสือตามหมวดหมู่
- อัลกอริทึม -> หาหนังสืออย่างไร
 - แบบ 1 ไล่หาจากเล่มแรกที่มีการลงทะเบียน
 - แบบ 2 หาจากหมายเลขหมวดหนังสือ

แบบไหนเร็วกว่า ?

กุญแจสำคัญ : ข้อมูล

- โปรแกรมจะมีความเกี่ยวข้องกับข้อมูล
 - รับข้อมูล (Receiving)
 - จัดการข้อมูล (Manipulating)
 - ส่งกลับข้อมูล (Returning)
- เป็นเช่นนี้เสมอ ไม่ว่าจะเป็นโปรแกรมง่ายๆ หรือโปรแกรมที่ซับซ้อน



ว่าด้วยเรื่องของ ข้อมูลทั่วไป

- ข้อมูลเชิงเดี่ยว (*Atomic data*)
 - เช่น ตัวเลขจำนวน 1234
- ข้อมูลประกอบ (*Composite data*)
 - เช่น หมายเลขโทรศัพท์

ชนิดข้อมูลในโปรแกรมคอมพิวเตอร์

- ข้อมูลเชิงเดี่ยว (*Atomic Data Type*)
 - Integer, Char, Boolean, Real
- ข้อมูลเชิงโครงสร้าง (*Structured Data Type*)
 - String, Array, File
- สอดคล้องประเภทของข้อมูลทั่วไป

ชนิดของข้อมูล (Data Type)

- ประกอบด้วย 2 ส่วน
 - กลุ่มค่าข้อมูล (A set of values)
 - การดำเนินการกับข้อมูล (A set of operations on values)
- ตัวอย่าง ข้อมูลชนิดเลขจำนวนเต็ม (Integer)
 - ค่าข้อมูล : จำนวนเต็มที่อยู่ในช่วงที่กำหนด
 - การดำเนินการกับข้อมูล : บวก, ลบ, คูณ,หาร

ชนิดของข้อมูล (Data Type) (cont.)

Type	Values	Operations
integer	$-\infty, \dots, -2, -1, 0, 1, 2, \dots, \infty$	$*, +, -, \%, /, ++, --, \dots$
floating point	$-\infty, \dots, 0.0, \dots, \infty$	$*, +, -, /, \dots$
character	$\backslash 0, \dots, 'A', 'B', \dots, 'a', 'b', \dots, \sim$	$<, >, \dots$

ชนิดข้อมูลนามธรรม

(ADT : Abstract Data Type)

- เป็นข้อมูล (แนวคิด) ที่นำมาใช้นิยามหรือกำหนดลักษณะของชนิดข้อมูลอื่นที่
- การออกแบบ ADT ทำได้โดย
 - กำหนดค่าข้อมูลที่ใช้ (Value Definition)
 - กำหนดตัวดำเนินการสำหรับข้อมูลข้างต้น (Operator Definition)
 - ซ่อนรายละเอียด (Encapsulation of data and operations)

ชนิดข้อมูลนามธรรม

(ADT : Abstract Data Type)

- ตัวอย่าง สร้างชนิดข้อมูล KString
 - กำหนดค่าข้อมูล
 - ประกอบด้วย ตัวอักษร a-z หรือ A-Z
 - กำหนดตัวดำเนินการ
 - กำหนดค่า ด้วยฟังก์ชัน makeStr()
 - เปรียบเทียบ ด้วยฟังก์ชัน isEqual()
 - หาความยาว ด้วยฟังก์ชัน length()

ชนิดข้อมูลนามธรรม

(ADT : Abstract Data Type)

- KString ADT

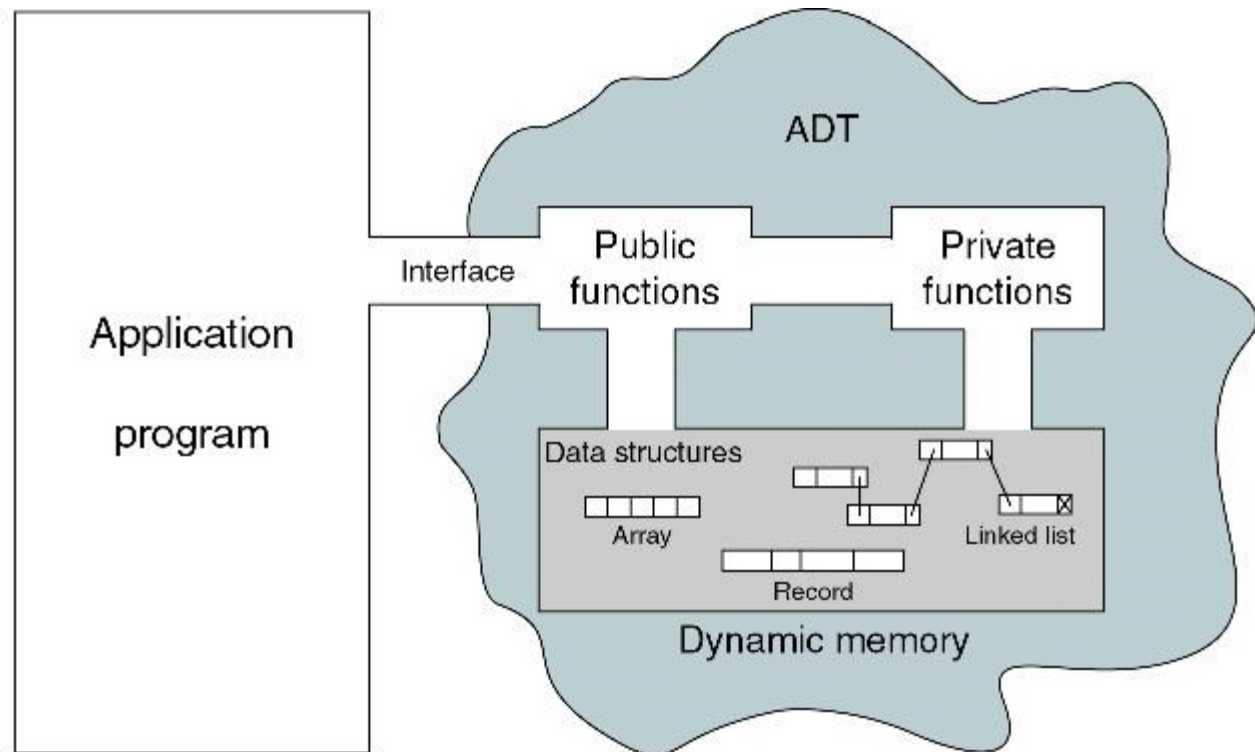
Abstract Type KString = sequence of Alpha;

Alpha = 'a'..'z' or 'A'..'Z';

Abstract Function makeStr(KString s);

Postcondition return(s);

Model for an ADT



โครงสร้างข้อมูล (Data Structure)

- คือ การรวมกันของข้อมูลเชิงเดี่ยวหรือข้อมูลเชิงโครงสร้างอื่นเข้าด้วยกัน
 - โครงสร้างข้อมูลแบบ Array, Array ช้อน Array
- ภาษาคอมพิวเตอร์มีโครงสร้างข้อมูลให้หลายรูปแบบ
- ผู้เขียนโปรแกรมสามารถสร้างโครงสร้างข้อมูลแบบใหม่ด้วยตัวเองได้

ADT vs Data Structure

- ADT = Data Structure ?

ADT vs Data Structure

- ADT = Data Structure ? -> No
- ADT เป็นแนวคิด
- Data Structure --> การนำ ADT มาสร้างให้เกิดผล

โครงสร้างข้อมูลชนิดต่างๆ

- ลิสต์แบบเชิงเส้น
- ลำดับชั้น
- กราฟ

Four Basic Ways with Data Structures

- Read
- Search
- Insert
- Delete

อัลกอริทึม (Algorithm)

- ขั้นตอนวิธีที่ใช้ในการแก้ปัญหา (หาเส้นทางจาก Initial state -> End state)
- อัลกอริทึม เป็นลักษณะของแนวคิด ซึ่งสามารถแสดงออกมาได้หลายรูปแบบ
 - ซูโดโค้ด (Pseudo code)
 - แผนผัง (Flowchart)
 - โปรแกรม (Program)

ซูโดโค้ด (Pseudo Code)

Algorithm sample (pageNumber)

This algorithm reads a file and prints a report.

Pre pageNumber passed by reference

Post Report Printed

 pageNumber contains number of pages in report

Return Number of lines printed

1 loop (not end of file)

1 read file

2 if (full page)

 1 increment page number

 2 write page heading

3 end if

4 write report line

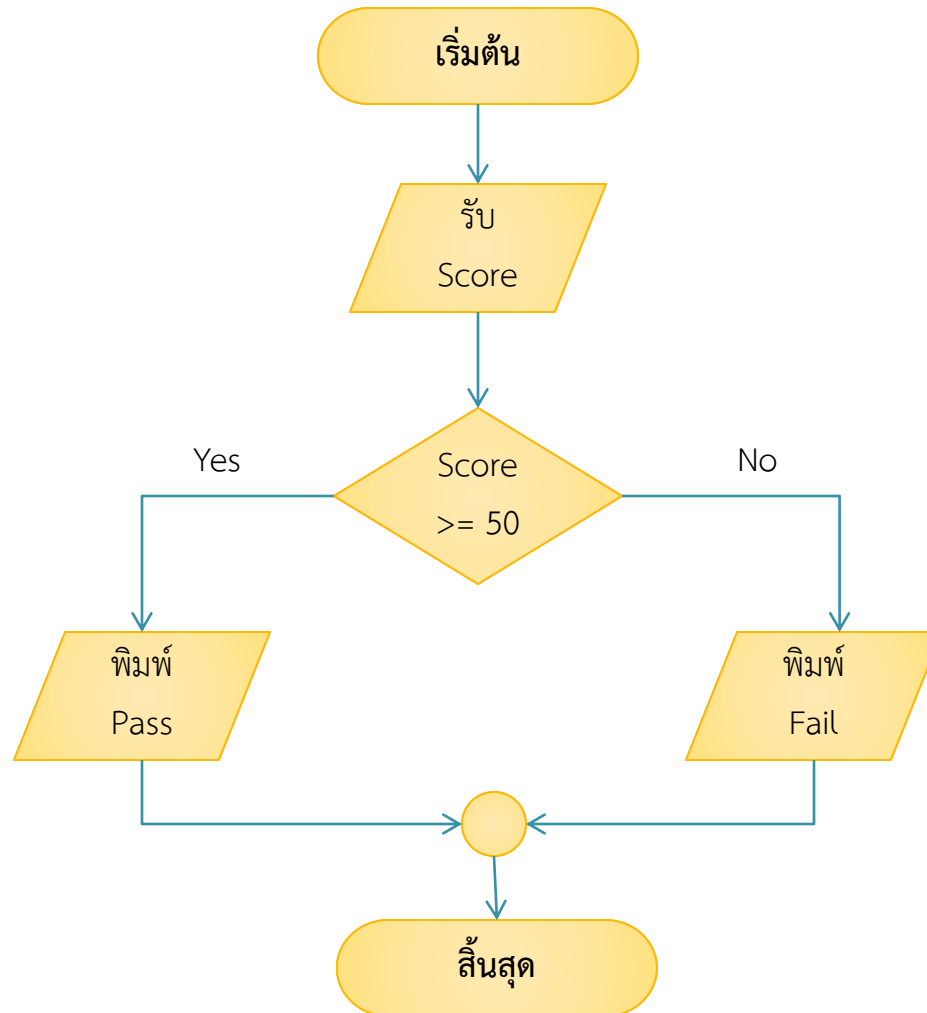
5 increment line count

2 end loop

3 return line count

end sample

แผนผัง (Flowchart)



โครงสร้างสำหรับการเขียนโปรแกรม

- แบบคำสั่งตามลำดับ (Sequential)
- แบบมีการตัดสินใจ (Selection : IF-THEN)
- แบบทำงานวนซ้ำ (Repetition : Loop)