

1

พื้นฐานความรู้ทั่วไป เกี่ยวกับโครงสร้างข้อมูล

นิยาม

โครงสร้างข้อมูล (Data Structure) หมายถึง การรวมประเภทข้อมูล (Data Type) เข้าไว้ด้วยกันจนกระทั่งกลายเป็นกลุ่มประเภทข้อมูล และมีการกำหนดค่านิยามของความสัมพันธ์ภายในกลุ่มข้อมูลไว้อย่างชัดเจน ซึ่งการรวมกลุ่มนั้นอาจจะเป็นการรวมกลุ่มกันระหว่างข้อมูลประเภทเดียวกัน ต่างประเภทกัน หรือต่างโครงสร้างข้อมูลกันก็ได้

ส่วนความสัมพันธ์ภายในกลุ่มข้อมูลก็คือ กลุ่มของกฎระเบียบข้อบังคับต่างๆ ที่ใช้ผูกข้อมูลเข้าไว้ด้วยกันอย่างเป็นระบบ นอกจากนั้นเรายังสามารถสร้างโครงสร้างข้อมูลขึ้นมาใหม่ซึ่งภายในประกอบไปด้วยโครงสร้างข้อมูลอื่นๆมารวมกันก็ได้ เพื่อใช้งานร่วมกับโปรแกรมที่เราเขียน

ชนิดของข้อมูล (Data Type)

ในภาษาคอมพิวเตอร์ระดับสูง (high level language)

ที่เราใช้เขียน โปรแกรมกันไม่ว่าจะเป็นภาษา

อะไรก็ตามมักจะมีชนิดของข้อมูลหลักให้เราเลือกใช้

เหมือนๆ กัน ได้แก่ integer, char, boolean, real, array

เป็นต้น บางภาษามีชนิดของข้อมูลให้ใช้

มากกว่าบางภาษา เช่น ในภาษา Pascal, C ฯลฯ

ชนิดของข้อมูล (Data Type)

ชนิดของข้อมูลเชิงเดี่ยว (Atomic Data Type)

ข้อมูลเชิงเดี่ยวคือข้อมูลซึ่งไม่สามารถจะแยกออกเป็นส่วนย่อยได้อีก ได้แก่ integer, char, boolean, real เป็นต้น เมื่อพูดถึงชนิดของข้อมูล เช่น integer เราจะนึกถึงสิ่งสามสิ่งในขณะเดียวกัน คือ ค่าของข้อมูล (*data value or data element*) ความสัมพันธ์หรือโครงสร้างของข้อมูล (*data structure*) และ การดำเนินงานกับข้อมูล (*set of operations*)

ชนิดของข้อมูล (Data Type)

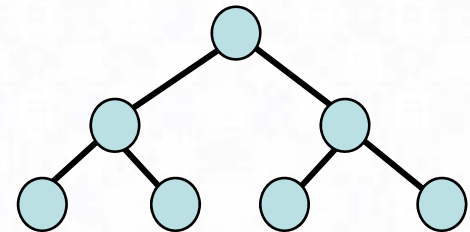
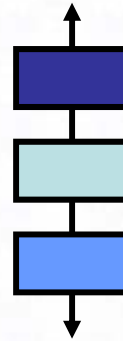
ชนิดของข้อมูลเชิงโครงสร้าง (Structured Data Type)

ชนิดของข้อมูลเชิงโครงสร้าง คือ ชนิดของข้อมูลซึ่งค่าของข้อมูล (data values) ประกอบจากเซตของข้อมูลย่อย (component) ซึ่งอาจจะเป็นข้อมูลเชิงเดี่ยวหรือข้อมูลเชิงโครงสร้างก็ได้ ตัวอย่างของข้อมูลเชิงโครงสร้างในภาษาปาสคาล ได้แก่ แถวลำดับ หรือ อะเรย์ (array), ระเบียบน หรือ เรคคอร์ด (record), ไฟล์ (file), และ เซ็ต (set) เป็นต้น

Introduction

โครงสร้างข้อมูลสามารถแบ่งประเภทได้ 2 แบบ ดังนี้

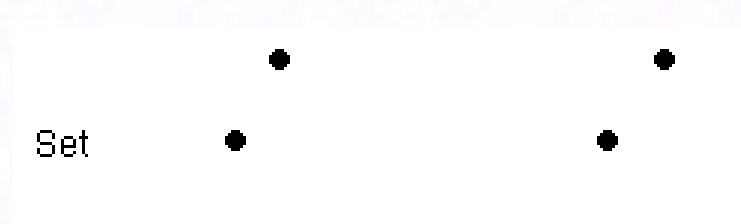
- แบบต่อเนื่อง (Linear) -> Array
 - Linked list
 - Stack
 - Queue
- แบบไม่ต่อเนื่อง (Nonlinear)
 - Tree
 - Graph



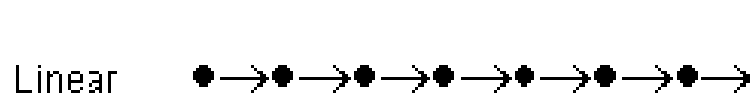
โครงสร้างข้อมูล แสดงถึงความสัมพันธ์ระหว่างข้อมูล

ความสัมพันธ์หลักใหญ่ ๆ 4 แบบ ได้แก่

- โครงสร้างข้อมูลแบบเซต (set) เป็นโครงสร้างที่ข้อมูลแต่ละตัวไม่มีความสัมพันธ์กันเลย



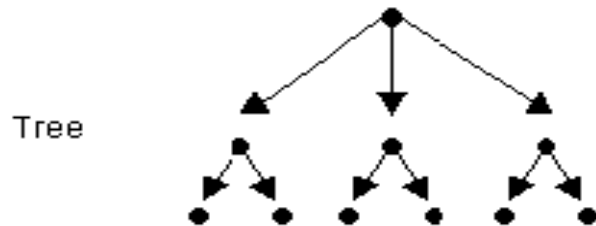
- โครงสร้างแบบเชิงเส้น (linear) เป็นโครงสร้างที่ข้อมูลมีความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one-to-one relationship) นั่นคือเราสามารถระบุถึงข้อมูลตัวถัดไปของข้อมูลได้



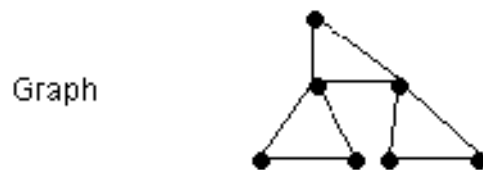
โครงสร้างข้อมูล แสดงถึงความสัมพันธ์ระหว่างข้อมูล

ความสัมพันธ์หลักใหญ่ ๆ 4 แบบ ได้แก่

- โครงสร้างแบบต้นไม้หรือแบบลำดับชั้น (tree or hierarchical) เป็นโครงสร้างที่ข้อมูลมีความสัมพันธ์กันแบบหนึ่งต่อหลาย (one-to-many relationship) นั่นคือข้อมูลตัวหนึ่งสามารถมีความสัมพันธ์กับข้อมูลในลำดับรองลงไปได้หลายตัว



- โครงสร้างแบบกราฟหรือเครือข่าย (graph or network) เป็นโครงสร้างที่ข้อมูลมีความสัมพันธ์กันแบบหลายต่อหลาย (many-to-many relationship) นั่นคือข้อมูลตัวหนึ่ง ๆ อาจจะมีการสัมพันธ์กับข้อมูลตัวอื่น ๆ ก็ได้



ตัวอย่าง 1.1

เปรียบเทียบโครงสร้างข้อมูลชนิดจำนวนเต็มและจำนวนจริง ข้อมูลชนิดจำนวนเต็มมีโครงสร้างทั้งแบบอันดับและแบบเชิงเส้น (ordered and linear) นั่นคือ เราสามารถจะเปรียบเทียบข้อมูลจำนวนเต็มสองจำนวนได้ และเราสามารถจะบอกอันดับถัดไป (succ) หรืออันดับก่อนหน้า (pred) หรืออันดับ (ord) ของข้อมูลที่กำหนดให้ได้ ส่วนข้อมูลชนิดจำนวนจริงมีโครงสร้างแบบอันดับอย่างเดียวไม่เป็นแบบเชิงเส้น (ordered but not linear) คือ เราสามารถจะเปรียบเทียบข้อมูลจำนวนจริงสองตัวได้ แต่ถ้ากำหนด ข้อมูลจำนวนจริงให้ตัวหนึ่งเราไม่สามารถจะบอกได้ว่า ข้อมูลจำนวนจริงตัวถัดไปคืออะไร เพราะว่าระหว่างจำนวนจริงสองจำนวน ไม่ว่าจะใกล้กันเท่าใดจะมีจำนวนจริงจำนวนไม่ถ้วนอยู่ระหว่างนั้น

การแทนโครงสร้างข้อมูลเชิง เส้น ในหน่วยความจำ

- วิธีที่ 1: เก็บข้อมูลในหน่วยความจำในลักษณะต่อเนื่องกัน
 - Sequential access เช่น Array
- วิธีที่ 2: เก็บข้อมูลไว้ ณ แอดเดรสที่เป็นฟังก์ชัน ของอักขรตัวแรกของชื่อตัวแปร
 - Random access
 - ไม่ทราบตำแหน่งของข้อมูลตัวถัดไป
- วิธีที่ 3: $2 +$ ตำแหน่งของข้อมูลตัวถัดไป
 - Linked List

การแทนโครงสร้างข้อมูลเชิงเส้น ในหน่วยความจำ

สมมติเรามีข้อมูลซึ่งเป็นรายชื่อของคนจำนวนหนึ่งสมมติมี 5 คน
คือ นพวรรณ วิลาวัณย์ สมบุญ
ดิเรก และ อารีรักษ์ สมมติรายชื่อเหล่านี้มีความสัมพันธ์กันแบบมี
อันดับและแบบเชิงเส้น เรามีวิธีที่จะแทนข้อ
มูลและโครงสร้างของข้อมูลนี้ 3 วิธี คือ
วิธีที่ 1 เก็บไว้ในหน่วยความจำที่ใดที่หนึ่งในลักษณะต่อกันไปได้
เลย

วิธีที่ 1 เก็บไว้ในหน่วยความจำที่ใดที่หนึ่งในลักษณะต่อกันไปได้เลย

- เก็บข้อมูลในหน่วยความจำในลักษณะต่อเนื่องกัน

หมายเลขประจำตำแหน่ง (Address)	ชื่อ (Name)
1000	นพวรรณ
1008	วิลาวณิชย์
1016	สมบุญ
1024	ศิเรก
1032	อารีรักษ์

วิธีที่ 2 เก็บข้อมูลไว้ ณ แอดเดรสที่เป็นฟังก์ชันของอักขระตัวแรก
ของชื่อ (fl)

เช่น

$1000 + 8 * [\text{ord}(\text{fl}) - \text{ord}('A')]$ ซึ่งเป็นวิธีการของ
Hashing function สมมติชื่อทั้ง 5 ชื่อดังกล่าวเก็บ
ไว้ในหน่วยความจำ ณ ตำแหน่ง

- เก็บข้อมูลในหน่วยความจำตำแหน่งที่เป็นฟังก์ชันของอักขรตัวแรกของชื่อตัวแปรหมายเลขประจำตำแหน่ง (Address) ชื่อ (Name)

หมายเลขประจำตำแหน่ง (Address)	ชื่อ (Name)
1000	ดิเรก
1008	-
1016	อารีรักษ์
1024	วิลาวัณย์
1032	สมบุญ
1104	นพวรรณ

วิธีการแทนที่ข้อมูลในลักษณะแบบนี้ไม่สามารถเก็บโครงสร้างของข้อมูลได้ นั่นคือ ถ้าเราต้องการทราบว่าใครต่อจากใครเราไม่สามารถจะบอกได้ การแทนที่ข้อมูลแบบนี้จะมีประโยชน์ตรงที่ว่า ถ้าเราอยากจะได้ค้นหาข้อมูลที่ชื่อ วิลาวรรณ เราสามารถจะทำการคำนวณได้โดยตรง (ใช้อินเวอร์สฟังก์ชันของการจัดเก็บ) ซึ่งเป็นการเข้าถึงข้อมูลแบบสุ่ม (random access) แต่ถ้าเราใช้วิธีการจัดเก็บ ตามวิธีที่ 1 เราต้องทำการค้นหาจากชื่อแรกจนพบชื่อที่ต้องการ ซึ่งเป็นการเข้าถึงข้อมูลแบบลำดับ (sequential access)

วิธีที่ 3 เก็บข้อมูลไว้ในหน่วยความจำแบบวิธีที่ 2 แต่เพิ่มฟิลด์สำหรับเก็บตำแหน่งของข้อมูล
อันดับถัดไปด้วย (ซึ่งเป็นการรวมข้อดีของทั้ง 2 วิธีเข้าด้วยกัน)

- เก็บข้อมูลในหน่วยความจำตำแหน่งที่เป็นฟังก์ชันของอักษรตัวแรกของชื่อ
ตัว

ตำแหน่ง	ชื่อ	ตำแหน่งของข้อมูลถัดไป
1000	ดิเรก	1032
1008	-	-
1016	อารีรักษ์	-
1024	วิลาวัณย์	1016
1032	สมบุญ	1000

โดยสรุปจะเห็นว่าในการแทนที่ข้อมูลในหน่วยความจำเพื่อให้
คงโครงสร้างแบบเชิงเส้นมี 2 วิธี คือ
วิธีที่ 1 ซึ่งเป็นวิธีการแบบแถวลำดับหรืออะเรย์ (array) โดยอันดับ
ของข้อมูลแสดงด้วยตำแหน่งของข้อมูลใน
อะเรย์และวิธีที่ 3 ซึ่งเป็นวิธีการแบบลิงค์ลิสต์ (link list) ซึ่งอันดับ
ของข้อมูลหาได้โดยติดตามลิงค์ ชนิดข้อมูล
แบบอะเรย์และแบบลิงค์ลิสต์ เป็นชนิดข้อมูลพื้นฐานที่ใช้สำหรับ
สร้างชนิดข้อมูลอื่น ๆ ต่อไป โดยจะมีการ
ศึกษาต่อไปในรายละเอียดในบทเรียนที่ 3 และบทเรียนที่ 4
ตามลำดับ