แบบฝึกหัด

วิชาโครงสร้างข้อมูล (Data Structure)

ชื่อ – สกุล .....

รหัสนักศึกษา ..... หมู่เรียน .....

สาขาวิชา .....คณะ.....

มหาวิทยาลัยราชภัฏมหาสารคาม

สอนโดย อาจารย์สุนันทา กลิ่นกาวร

เค้าโครงการสอน

รหัสวิชา : 4122202 **ผู้สอน** อาจารย์สุนันทา กลิ่นถาวร

ชื่อวิชา : โครงสร้างข้อมูล **ภาคเรียนที่** 1/2552

หน่วยกิต / ชั่วโมงการเรียน : 3 (2 – 2)

---

1. คำอธิบายรายวิชา

ศึกษาหลักการของโครงสร้างข้อมูลแบบคัดย่อ (Abstract Data Type -ADT) การแทนที่ข้อมูลและโครงสร้างข้อมูลในหน่วยความจำ มาตรวัดสำหรับประสิทธิภาพของอัลกอริทึม (algorithm) โครงสร้างข้อมูลสำหรับการโปรแกรมที่นอกเหนือจากโครงสร้างข้อมูลพื้นฐาน ที่มีมากับภาษาโปรแกรมต่าง ๆ ได้แก่ ข้อมูลแบบสแตก (stack) คิว (queue) ลิสต์ (list) ต้นไม้หรือทรี (tree) ตารางแฮช (hash table) กราฟ (graph) และการประยุกต์ ตลอดจนการเรียงลำดับและการค้นหาข้อมูล (sorting and searching) โดยการนำเสนอด้วยหลักการของโครงสร้างข้อมูลแบบคัดย่อ

2. วัตถุประสงค์การเรียนรู้

เมื่อเรียนจบวิชานี้แล้ว นักศึกษามีความคุณลักษณะดังนี้

2.1 เพื่อให้นักศึกษาเรียนรู้ถึงชนิดข้อมูลแบบต่าง ๆ ที่เป็นพื้นฐานที่จำเป็นสำหรับการเรียนในวิชาอื่น ๆ ที่สูงขึ้นไปที่เกี่ยวข้องกับสาขาวิทยาการคอมพิวเตอร์ และเป็นชนิดข้อมูลที่ไม่ได้เป็นชนิดข้อมูลมาตรฐานที่มากับโปรแกรมภาษาในยุคที่ 3 (โปรแกรมภาษาสูง) ชนิดข้อมูลที่ศึกษาได้แก่ สแตก คิว ลิสต์ ซึ่งมีโครงสร้างข้อมูลแบบเชิงเส้น (linear) ต้นไม้ ซึ่งมีโครงสร้างแบบลำดับชั้น (hierachical) และกราฟ ซึ่งมีโครงสร้างข้อมูลแบบเครือข่าย (network)

2.2 เพื่อศึกษาหลักการของวิธีเรียงลำดับข้อมูล และหลักการของการค้นหาข้อมูล ซึ่งเป็นกิจกรรมที่สำคัญมากในการประยุกต์การใช้งานกับข้อมูล

2.3 เพื่อนำเสนอการศึกษาชนิดข้อมูลและโครงสร้างข้อมูลแบบต่าง ๆ ด้วยหลักการของชนิดข้อมูลแบบคัดย่อ (Abstract Data Type -ADT) ซึ่งเป็นรากฐานที่สำคัญของการโปรแกรมเชิงวัตถุ (Object Oriented Programming)

3. เนื้อหาการสอน

จากสาระในเนื้อหาวิชาทั้งหมด ได้จัดแบ่ง เป็น 15 บทเรียน ดังนี้

บทที่ 1	พื้นฐานความรู้ทั่วไป
บทที่ 2	โครงสร้างข้อมูลแบบ Array
บทที่ 3	โครงสร้างข้อมูลแบบ Stack และ Qeue
บทที่ 4	โครงสร้างข้อมูลแบบ Linked List

บทที่ 5 โครงสร้างข้อมูลแบบ Tree

บทที่ 6 โครงสร้างข้อมูลแบบ Graph

บทที่ 7 การเรียงลำดับข้อมูล

บทที่ 8 การค้นหาข้อมูล

4. การประเมินผลงาน

การประเมินผลงานในวิชานี้ ยึดพื้นฐานขึ้นอยู่กับหลักฐานการปฏิบัติงานต่าง ๆ ของนักศึกษาที่แสดงต่อผู้สอน และยึดนโยบายความซื่อสัตย์สุจริตเป็นสำคัญ

4.1	การมีส่วนร่วมในชั้นเรียน	- การมีส่วนร่วมในกิจกรรมกลุ่ม - การเข้าชั้นเรียน	10 %
4.2	แบบฝึกหัด	- แบบฝึกหัด	30 %
4.3	การสอบกลางภาคเรียน		30 %
4.4	การสอบปลายภาค		30 %

หมายเหตุ *** ระดับคะแนนที่นักศึกษาได้รับนั้น ใช้ระบบอิงเกณฑ์ มีค่าระดับคะแนนดังนี้

คะแนน	80 - 100	เกรด	A
คะแนน	75 - 79	เกรด	B+
คะแนน	70 - 74	เกรด	B
คะแนน	65 - 69	เกรด	C+
คะแนน	60 - 64	เกรด	C
คะแนน	55 - 59	เกรด	D+
คะแนน	50 - 54	เกรด	D
คะแนน	0 - 49	เกรด	E

แบบฝึกหัดบทที่ 1

ความรู้พื้นฐานเกี่ยวกับโครงสร้างข้อมูล

วิชาโครงสร้างข้อมูลและอัลกอริธึม เป็นการศึกษาในเรื่องของการจัดการกับข้อมูล และการประมวลผลในคอมพิวเตอร์ วิชานี้เป็นพื้นฐานที่มีความสำคัญเป็นอย่างยิ่งในการออกแบบและพัฒนาโปรแกรม ให้มีประสิทธิภาพ ตัวอย่างเช่นการการจัดการเกี่ยวกับการค้นหาข้อมูลที่มีขนาดใหญ่อย่างเว็บไซ ต์ เสิร์ชเอ็นจิ้น google.com ซึ่ง มีการจัดเก็บข้อมูลที่ไม่ต่ำกว่าหกพันล้านหน้า และเมื่อต้องการค้นหาข้อมูลที่มีในฐานข้อมูลทั้งหมด หากไม่มีวิธีการจัดเก็บและค้นหาที่มีประสิทธิภาพแล้ว เราคงต้องใช้เวลาในการค้นหาเป็นปีทีเดียว

การ พัฒนาโปรแกรมทางด้านคอมพิวเตอร์เพื่อให้มีประสิทธิภาพสูงนั้นจำเป็นที่จะ ต้องมีวิธีการในการจัดเก็บข้อมูล การประมวลผลที่มีประสิทธิภาพ คือใช้เวลาน้อยในการประมวลผล และสามารถประมวลผลกับข้อมูลขนาดใหญ่ ๆ ได้ ซึ่งสามารถที่จะช่วยลดค่าใช้จ่ายได้โดยเฉพาะอย่างยิ่งข้อมูลในปัจจุบันมี ขนาดที่ใหญ่มากขึ้นเรื่อย ๆ หากไม่มีวิธีจัดการกับข้อมูลที่ดีแล้ว อาจจะเสียทั้งเงินทองและเวลาได้ ฉะนั้นการศึกษาถึงพื้นฐานของโครงสร้างข้อมูล และอัลกอริธึมจึงมีความสำคัญอย่างยิ่ง ที่จะช่วยให้นักศึกษาเข้าใจถึงปัญหาเพื่อที่จะนำเอาความรู้ที่ศึกษาไป ประยุกต์ใช้เพื่อแก้ปัญหาได้อย่างมีประสิทธิภาพ

พื้นฐานที่นักศึกษาควรจะมีคือพื้นฐานการเขียนโปรแกรมเช่น การใช้ array การใช้ conditions ต่าง ๆ ภายในเนื้อหานี้ผู้สอนไม่ได้เน้นที่ภาษาใดภาษาหนึ่ง ขอเพียงให้เข้าใจถึงวิธีการ ก็สามารถที่จะนำไปใช้ในภาษาอื่น ๆ ได้ **Array** เป็นโครงสร้างข้อมูลที่ประกอบด้วยกลุ่มของตัวแปรที่เป็นค่าอะไรก็ได้ แต่ต้องมีชนิดเดียวกัน ข้อมูลแต่ละตัวใน Array เรียกว่าสมาชิกของ array การอ้างถึงจะเรียกชื่อตัวแปร array แล้วตามด้วยหมายเลขดัชนี (index) ของสมาชิกที่ต้องการอ้างถึง โดยสมาชิกตัวแรกจะแทนด้วยลำดับหมายเลข 0 สมาชิกตัวที่2 แทนด้วยลำดับหมายเลข 1 ..จนถึงลำดับที่ n

1. จงอธิบายคำศัพท์ต่อไปนี้พอสังเขป พร้อมยกตัวอย่าง

   1.1 Entity Set ……

   1.2 Data Elements ……

   1.3 Data Structure ……

   1.4 Subscripted Variable ……

   1.5 Pointer ……

1.6 Assignment Statement .....

1.7 Stack .....

1.8 Queue .....

2. จงอธิบายถึงความสัมพันธ์ของ Field, Record, Files พร้อมยกตัวอย่าง

.....

.....

.....

.....

.....

.....

3. องค์ประกอบที่สำคัญของการจัดโครงสร้างข้อมูลมีอะไรบ้าง พร้อมทั้งอธิบาย

.....

.....

.....

.....

.....

.....

.....

4. ประเภทของโครงสร้างข้อมูลสามารถจำแนกออกได้เป็นกี่ประเภท และแต่ละประเภทมีกระบวนการอะไรบ้าง ให้อธิบายมาพอสังเขป

.....

.....

.....

.....

.....

.....

.....

.....

5. ให้นักศึกษานำตัวเลือกในคอลัมน์ B มาใส่ไว้ข้างหน้าคอลัมน์ A โดยเลือกให้ถูกต้องที่สุด

คอลัมน์ A	คอลัมน์ B
……..5.1 อายุ	ก. Subscrip
……..5.2 ข้อมูลหรือรายการย่อย ๆ	ข. Algorithms
……..5.3 โครงสร้างข้อมูลที่ซับซ้อน	ค. (↑)
……..5.4 เข้าหลังออกก่อน	ง. Data Elements
……..5.5 A[n]	จ. Pascal
……..5.6 Step by Step	ฉ. Fields
……..5.7 Var (คำสั่ง)	ช. Stack
……..5.8 การยกกำลัง	ซ. 2^n
	ฌ. Linked List
	ญ. Search Tree

แบบฝึกหัดบทที่ 2

โครงสร้างข้อมูลแบบ ARRAY

อาร์เรย์ เป็นแบบหนึ่งของโครงสร้างที่เรียกว่า Linear List ซึ่งมีจำนวนรายการ (Element) จำกัด และข้อมูลที่เก็บอยู่ในอาร์เรย์แต่ละช่องจะต้องเป็นข้อมูลชนิดเดียวกัน อยู่ภายใต้ตัวแปรชื่อเดียวกัน โดยขนาดของแต่ละช่องต้องเท่ากันหมด การอ้างถึงข้อมูลในแต่ละช่องของของอาร์เรย์ ต้องอาศัยตัวห้อย Subscript เช่น กำหนดให้ Array A มีขนาด 100 รายการ A[5] จะหมายถึง ค่าของอาร์เรย์ตำแหน่งที่ 5 ในอาร์เรย์นั้น ซึ่ง Subscript ก็คือ เลข 5 จำนวน Subscript ที่ต้องการใช้เวลาเรียกใช้ค่าใน Array เรียกว่า มิติ ไดเมนชั่น (Dimention) ของ Array นั้น

ความหมายของอาร์เรย์

โครงสร้างข้อมูลแบบอาร์เรย์ (Array) หรือตัวแปรชุด มี 2 ความหมาย คือ

1. ความหมายโดยทั่วไปอาร์เรย์ หมายถึงโครงสร้างที่นำข้อมูลชนิดเดียว กันมาจัดเรียงกันเป็น n มิติ เป็นโครงสร้างตารางรูปสี่เหลี่ยมผืนผ้า

2. ความหมายทางคอมพิวเตอร์อาร์เรย์ หมายถึง กลุ่มของช่วงความจำ ในหน่วยความที่ใช้เก็บข้อมูลชนิดเดียวกันและ ทุกช่องต้องมีขนาดเท่ากัน ภายใต้ตัวแปรเดียวกัน

การสร้าง Array ขึ้นมาใช้งานนั้น ต้องคำนึงถึง

1. ชื่อของ Array
2. ขนาดของ Array แต่ละช่อง และมิติของ Array
3. ค่าสูงสุด (Upper Bound) และค่าต่ำสุด (Lower Bound) ในแต่ละมิติ

1. จงเขียนอาร์เรย์ทั้ง 2 รูปแบบเมื่อกำหนดให้อาร์เรย์ A โดยกำหนดจำนวนช่อง จัดเก็บ 5 ช่อง ตำแหน่งที่ 5 ระบุข้อมูลมีค่าเป็น 10

.....

.....

.....

.....

.....

.....

.....

2. การเก็บคะแนนสอบของนักศึกษาเลขประจำตัว 1741 – 1750 ได้คะแนนดังนี้ 50, 60, 70, 60, 55, 71, 62, 59, 41 จงเขียนค่าแสดงรูปแบบของอาร์เรย์

……..

……..

……..

……..

……..

……..

……..

3. การเก็บคะแนนสอบของนักศึกษาเลขประจำตัว 1741 – 1750 ได้คะแนนดังนี้ 50, 60, 70, 60, 55, 71, 62, 59, 41 จงคำนวณหาจำนวนช่องของอาร์เรย์

……..

……..

……..

……..

……..

……..

……..

4. ร้านวีดีโอแห่งหนึ่งต้องการสำรวจยอดจำนวนสมาชิกที่เข้ามาใช้บริการในแต่ละปี โดยต้องการเก็บข้อมูลตั้งแต่ พ.ศ. 2540 ถึง พ.ศ 2550 จงหาจำนวนช่องตารางของอาร์เรย์ชุดนี้ เมื่อระบุให้อาร์เรย์มีชื่อว่า VDO

……..

……..

……..

……..

……..

……..

……..

……..

……..

5. อาร์เรย์ TEST จองพื้นที่ไว้ 5 ช่อง คือ TEST [3:7] ใช้ความกว้างของแต่ละช่องจัดเก็บ 4 ไบต์ เก็บในพื้นที่หน่วยความจำตำแหน่งแรกคือ 500 จงหา add(a[5])

……..

……..

……..

……..

6. จงหาที่อยู่ของ S[3,5] ในแบบ column major

I	=	แถวของ array ที่ต้องการหา	= 3
J	=	หลักของ array ที่ต้องการหา	= 5
l1	=	ขอบเขตล่างของแถว	= -2
u1	=	ขอบเขตบนของแถว	= 4
l2	=	ขอบเขตล่างของหลัก	= 2
u2	=	ขอบเขตบนของหลัก	= 6
C	=	เนื้อที่ของสมาชิกแต่ละตัวใช้	= 4

……..

……..

……..

……..

……..

……..

……..

……..

……..

……..

……..

……..

แบบฝึกหัดบทที่ 3

โครงสร้างข้อมูลแบบ STACK และ QUEUE

สแตคเป็นโครงสร้างข้อมูลที่มีลักษณะแบบลำดับ (sequential) คือการกระทำกับข้อมูลจะกระทำที่ปลายข้างเดียวกันที่ส่วนปลายสุดของสแตค การกระทำกับข้อมูลของสแตคประกอบไปด้วยการนำเข้าข้อมูลเข้า (PUSH) ที่ส่วนบนสุดของสแตค และการนำข้อมูลออก (POP) ที่ส่วนบนสุดของสแตคเช่นกัน ในการจะ Push ข้อมูลเข้าก็ต้องตรวจสอบด้วยว่าข้อมูลในสแตคเต็มหรือไม่ หากสแตคเต็มก็จะไม่สามารถ Push หรือนำข้อมูลเข้าได้ เช่นเดียวกับการ Pop ข้อมูลออกก็ต้องตรวจสอบด้วยว่ามีข้อมูลอยู่ในสแตคหรือไม่ หากไม่มีข้อมูลอยู่ในสแตคหรือสแตคว่าง (empty stack) ก็ไม่สามารถ pop ได้ การนำข้อมูลเข้า-ออก จากสแตค (push , pop) จะมีลักษณะแบบเข้าหลัง ออกก่อน (LIFO : Last In , First Out) คือ ข้อมูลที่เข้าไปในสแตคลำดับหลังสุด จะถูกนำข้อมูลออกจากสแตคเป็นลำดับแรก ยกตัวอย่างการทำงานแบบ LIFO เช่น การวางจานซ้อนกัน

Operation ของสแตค

1.การเพิ่มข้อมูลลงในสแตค การเพิ่มข้อมูลลงในสแตค คือ การนำเข้ามูลเข้าสู่สแตคโดยทับข้อมูลที่อยู่บนสุดของสแตค ข้อมูลจะสามารถนำเข้าได้เรื่อยๆ จนกว่าสแตคจะเต็ม สมมติว่าสแตคจองเนื้อที่ไว้ N ตัว ถ้าหากค่า TOP เป็น 0 แสดงว่าสแตคว่าง หากค่า TOP = N แสดงว่าสแตคเต็มไม่สามารถเพิ่มข้อมูลลงในสแตคได้อีก

2.การดึงข้อมูลออกจากสแตค ก่อนที่จะดึงข้อมูลออกจากสแตคต้องตรวจสอบก่อนว่าสแตคมีข้อมูลอยู่หรือไม่ หรือว่าเป็นสแตคว่าง (Empty Stack)

โครงสร้าง : คิว จะมีโครงสร้างแบบเชิงเส้น และอันดับของการนำสมาชิกเข้าออกจากคิวมีความสำคัญ คือ สมาชิกที่เข้าไปอยู่ในคิวก่อนจะออกจากคิวก่อนสมาชิกที่เข้าไปในคิวทีหลัง นั่นคือการเข้าก่อนออกก่อน (First In First Out หรือ FIFO) การดำเนินงาน: มีการดำเนินงานทั้งหมด 6 การดำเนินงาน ได้แก่ สัญลักษณ์ : ให้ Q-pre เป็นสถานะของคิวก่อนการดำเนินงาน ให้ Q-post เป็นสถานะของคิวหลังการดำเนินงาน Crete Pre: ไม่มี Post: มีคิวว่างเกิดขึ้น

Enqueue (E : StdElement)Pre: คิวยังไม่เต็ม Post: คิวจะมี E เป็นสมาชิกล่าสุดของคิว และ E จะถูกใส่ลงในท้ายคิว (tail) Serve (VAR E : StdElement) Pre: คิวต้องไม่ว่าง Post: E เป็นสมาชิกแรกสุด (head) ของ Q-pre คิวหลังการดำเนินงานหรือ Q-post จะไม่มี E เป็นสมาชิกอีกต่อไปEmpty : booleanPre: ไม่มี Post: ถ้า คิวไม่มีสมาชิกอยู่เลย แล้ว Empty จะเป็นจริง ไม่เช่นนั้น Empty จะเป็นเท็จ

Full : booleanPre: ไม่มี Post: ถ้า คิวมีสมาชิกเต็ม แล้ว Full จะเป็นจริง ไม่เช่นนั้น Full จะเป็นเท็จClearPre: ไม่มี Post: คิวจะเป็นคิวว่างLength : IntegerPre: ไม่มี Post: Length จะเป็นจำนวนสมาชิกในคิว

1. จงอธิบายถึงสิ่งที่เหมือนกันและสิ่งที่ต่างกันระหว่างสแตคกับคิว

.....

.....

.....

.....

2. “ท็อบของสแตค” คืออะไร จงอธิบาย

.....

.....

.....

.....

3. จงวาดภาพสแตคที่เกิดขึ้น เมื่อเกิดการดำเนินการเหล่านี้ตามลำดับ

3.1 push (S,F) 3.2 push (S,D) 3.3 stacktop (S)

3.4 pop (S) 3.5 push (S,F) 3.6 empty (S)

3.7 pop (S)

4. ถ้ามีข้อมูลตามลำดับดังนี้ 10 คะแนน

ลำดับ	1	2	3	4	5	6
ข้อมูล	A	D	S	U	W	X

4.1 ถ้านำข้อมูลนี้ไปเก็บโดยใช้ Queue แล้วทำการนำข้อมูลออกจาก Queue ให้ตอบคำถาม

ข้อมูลที่ออกจาก Queue ลำดับแรกคือ.....

ค่าของ Rear คือ.....

ค่าของ Front คือ.....

4.2 ถ้านำข้อมูลนี้ไปเก็บโดยใช้ Stack แล้วทำการ pop Stack 1 ครั้ง ให้ตอบคำถาม

ข้อมูลที่ออกจาก Stack ลำดับแรก คือ.....

top of stack คือ.....

5. จงแปลงนิพจน์ต่อไปนี้เป็นนิพจน์ Postfix

5.1 A+B/C*D-E

อินพุต	เอาท์พุท	Operator stack

5.2 A+B*(C**D*E/F)-G

อินพุต	เอาท์พุท	Operator stack

6. จงเปรียบเทียบข้อดี ข้อเสียของโครงสร้างข้อมูลสแตค คิววงกลม คิวเส้นตรง

.....

.....

.....

.....

.....

.....

แบบฝึกหัดบทที่ 4

โครงสร้างข้อมูลแบบ LINKED LIST

ลิงค์ลิสต์เป็นการจัดเก็บชุดข้อมูลเชื่อมโยงต่อเนื่องกันไปตามลำดับ ซึ่งอาจอยู่ในลักษณะแบบเชิงเส้นตรง (linear) หรือ ไม่เป็นเส้นตรง (nonlinear) ก็ได้ ซึ่งในลิสต์จะประกอบไปด้วยข้อมูลที่เรียกว่าโหนด (node) ในหนึ่งโหนดจะประกอบด้วยส่วนของข้อมูลที่ต้องการจัดเก็บ เรียกว่าส่วน Info และส่วนที่เป็นพอยน์เตอร์ที่ชี้ไปยังโหนดถัดไป (Link) หรือชี้ไปยังโหนดอื่นๆที่อยู่ในลิสต์ หากไม่มีโหนดที่อยู่ถัดไป ส่วนที่เป็นพอยน์เตอร์หรือ Link จะเก็บค่า NULL หรือ NILL ใช้สัญลักษณ์ ^

Singly Linked list

Singly Linked list จะประกอบด้วยโหนดที่มีพอยน์เตอร์ชี้ไปในทิศทางเดียว คือชี้ไปยังโหนดถัดไป

Linked list ประกอบไปด้วย node ซึ่งมีข้อมูลของแต่ละ node อยู่ประกอบไปด้วย ข้อมูลและตัวที่บ่งชี้ถึงโหนดต่อไป หากเป็นโหนดสุดท้ายจะมีค่าเป็น null ในภาษา java เราสามารถอ้างถึง reference ได้โดย object จึงสามารถที่จะสร้าง linked list ได้เช่นกันเหมือนกับภาษา C หรืออื่น ๆ

Class ListNode { Object element; Node next; }

เปรียบเสมือนกับ element เป็นตัวเก็บข้อมูล และ next ทำหน้าที่ในการชี้ถึงโหนดอื่น ๆ ต่อไป

การท่องไปใน Linked list

สมมติมี Linkedlist หากต้องการดูข้อมูลที่อยู่ใน Linked list สามารถหาได้โดยการค้นหาแบบลำดับ (sequential search) ในส่วนของพอยน์เตอร์ H (Head) เราจะไม่เลื่อนพอยน์เตอร์ เนื่องจากถ้าเลื่อนพอยน์เตอร์ H จะทำให้กลับไปเริ่มตั้งต้นที่หัวลิสต์ไม่ได้ ดังนั้นจะใช้พอยน์เตอร์อีกตัวมาช่วย (auxiliary pointer) ในการท่องเข้าไปในลิสต์

การสร้าง Linked list

วิธีสร้าง Linked list คือการนำข้อมูลที่จะจัดเก็บเข้า Linked list เพิ่มตรงโหนดตำแหน่งสุดท้ายของลิสต์ ฉะนั้นจึงต้องมี External พอยน์เตอร์ที่คอยชี้โหนดสุดท้ายของลิสต์ ในที่นี้ใช้ L (Last)

การลบข้อมูลที่ต้น list

เนื่องจากขั้นตอนของการลบข้อมูลที่ header นั้นจะมีปัญหาที่ยุ่งยากกว่าเมื่อ design ด้วย oop(java) เราสามารถที่จะแก้ปัญหานี้ได้โดยการใส่ header node ที่ว่าง ๆ ไว้ข้างหน้าของ linked list เพื่อที่จะทำหน้าที่เป็นชี้ว่าเป็นหัวโหนดโดยที่ไม่ต้องมี pointer คอยชี้ที่ header และเมื่อเราต้องการที่จะเปลี่ยนแปลงข้อมูลใด ๆ บนหัวสามารถที่จะทำได้โดยการแทรก node เข้าไปดังตัวอย่างของการแทรกข้อมูลข้างล่าง

การแทรกข้อมูลลงในโหนดที่ต้องการ

Circular Linked list

ในปกติ Linked list เมื่อถึงโหนดสุดท้าย ค่าในฟิลด์ LINK หรือ โหนดถัดไปจะมีค่าเป็น NULL ซึ่งเราสามารถใช้ประโยชน์ได้โดยการเปลี่ยนให้ ค่า NULL ในฟิลด์ LINK เป็นตำแหน่งของโหนดแรกในลิสต์หรือชี้ไปที่ต้นลิสต์ใหม่นั่นเอง ซึ่งเราจะเรียก Linked list แบบนี้ว่า Circular Linked list
Circular Linked list ใช้ ประโยชน์เมื่อต้องการให้ข้อมูลมีลักษณะเป็นวนรอบหรือลูป โดยแต่ละขั้นตอนการทำงานภายในลูป จะมีการย้ายตำแหน่งของพอยน์เตอร์ไปยังโหนดถัดไปใน Linked list ในลักษณะแบบวงกลม

Doubly Linked list

Doubly linked list ประกอบด้วยส่วนของ Info และ พอยน์เตอร์ที่ชี้ไป 2 ทิศทาง คือ ชี้ไปยังโหนดถัดไปและชี้ไปยังโหนดก่อนหน้า ดังนั้นเราจึงสามารถทำการอ่านข้อมูลได้ 2 วิธี คือ การอ่านไปข้างหน้า และอ่านไปทางข้างหลัง

1. จงอธิบายคุณลักษณะของลิงค์ลิสต์

.....

.....

.....

.....

.....

.....

2. ลิงค์สิสต์เดี่ยวและลิงค์สิสต์คู่มีส่วนที่เหมือนกัน/แตกต่างกันอย่างไร

.....

.....

.....

.....

.....

.....

.....

3. จงเขียนอัลกิริธึมการลบโหนดออกจากลิงค์ลิสต์คู่

.....

.....

.....

.....

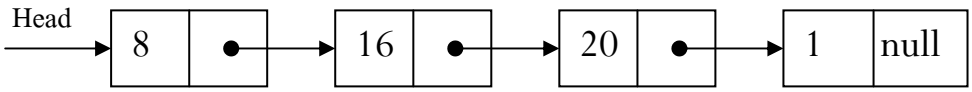
.....

.....

.....

.....

4. สำหรับลิงลิสต์ในรูปต่อไปนี้ จงวาดรูปลิงค์ลิสต์เมื่อทำตามคำสั่งต่อไปนี้



4.1 addnode (Numlist , 14)

4.2 delnode (Numlist , 16)

4.3 displaynode (Numlist)

4.4 insafter (Numlist , 14)

.....

.....

.....

.....

.....

.....

.....

.....

5. จงเลือกคำตอบที่ถูกที่สุดเพียงข้อเดียว

1. ลักษณะพิเศษของ Linked List กับข้อใด
   ก.) มีความคงที่
   ข.) เข้าใจง่าย
   ค.) ความซับซ้อนน้อย
   ง.) สามารถยืดหยุ่นได้

2. หน้าที่ของ Pointer ใน Linked List คือข้อใด
   ก.) เชื่อมโยงข้อมูล
   ข.) เชื่อมโยงข้อมูลต่อกันเป็นลำดับ
   ค.) เชื่อมโยงข้อมูลข้ามไป
   ง.) ชี้ข้อมูลที่อยู่ข้ามไป

3. Info หมายถึงข้อใด
   ก.) ส่วนที่เก็บ Pointer
   ข.) ส่วนที่เก็บ Reccord
   ค.) ส่วนเก็บข้อมูล
   ง.) ไม่มีข้อใดถูก

4. ข้อใดไม่ใช่ชุดปฏิบัติการของลิงค์ลิสต์ที่ทำงานร่วมกับ Pointer
   ก.) Node (P)
   ข.) Info (P)
   ค.) Null (P)
   ง.) Next (P)

5. โครงสร้างแบบ Linked List แบ่งเป็นที่ประเภท
   ก.) 2
   ข.) 3
   ค.) 4
   ง.) 5

6. มี Pointer ชี้ข้อมูลไปในทิศทางเดียวเป็น Linked List ประเภทใด
   ก.) Singly Linked List
   ข.) Doubly Linked List
   ค.) Three Linked List
   ง.) Foult Linked List

7. ส่วนที่เก็บ Info และ Pointer คือข้อใด
   ก.) Node
   ข.) List
   ค.) Record
   ง.) Ale

8. Beginning คืออะไร
   ก.) การเพิ่มข้อมูลขึ้นด้านหน้า
   ข.) การเพิ่มข้อมูลขึ้นด้านหลัง
   ค.) การแทรกข้อมูล
   ง.) การลบข้อมูลด้านหน้า

9. Left Pointer หมายถึงอะไร
   ก.) เก็บ Pointer ทางซ้าย
   ข.) เก็บ Pointer ทางขวา
   ค.) เก็บ Pointer ตรงกลาง
   ง.) เก็บข้อมูล

10. Circutar Doubly Linked List เป็น Linked List มีลักษณะอย่างไร
   ก.) ทิศทางเดียว
   ข.) 2 ทิศทาง
   ค.) ทิศทางเดียวแบบวงกลม
   ง.) 2 ทิศทางแบบวงกลม

แบบฝึกหัดบทที่ 5

โครงสร้างข้อมูลแบบ TREE

ทรี หรือโครงสร้างข้อมูลแบบต้นไม้ ประกอบด้วยโหนด (node) ซึ่งเป็นส่วนที่เก็บข้อมูล ในทรีหนึ่งทรีจะประกอบไปด้วยรูทโหนด (root node) เพียงหนึ่งโหนด แล้วรูทโหนดสามารถแตกโหนดออกเป็นโหนดย่อยๆ ได้อีกหลายโหนดเรียกว่าโหนดลูก (Child node) เมื่อมีโหนดลูกแล้ว โหนดลูกก็ยังสามารถแสดงเป็นโหนดพ่อแม่ (Parent Node) โดยการแตกโหนดออกเป็นโหนดย่อยๆได้อีก

Leaf Node คือ โหนดที่ไม่มีโหนดลูก

Branch Node คือ โหนดที่ไม่ใช่ Leaf Node เช่น โหนด B C D E เรียกว่า Branch Node

Degree คือ จำนวนลูกของโหนด x เช่น degree ของ โหนด A = 4 ได้แก่ B C D E จำนวน degree ของโหนด B = 2 จำนวนdegree ของโหนด F = 0 เนื่องจากโหนด F ไม่มีโหนดลูก

Direct Descendant Node คือโหนดที่มาทีหลัง

Descendant Node คือ โหนดลูกของโหนด x และโหนดที่ทุกโหนดที่แตกจากโหนดลูกของโหนด x

Direct Ancestor Node หรือโหนดที่มาก่อนทันที

Level หรือระดับ คือ หมายเลขแสดงระดับของโหนดในทรี ซึ่งรูทโหนดจะมีค่า Level = 0 ส่วนโหนดลูกของรูทโหนดก็จะมีค่า = 1 หากค่าโหนด x อยู่ในระดับ L โหนดลูกของ x ก็จะอยู่ในระดับ L + 1

Binary Tree มีลักษณะเหมือนกับ Tree ปกติแต่มีคุณสมบัติพิเศษ คือ “แต่ละโหนดจะมีโหนดลูกได้ไม่เกิน 2 โหนด” หรือพูดอีกนัยหนึ่งก็คือ แต่ละโหนดใน binary tree จะมีดีกรีได้ไม่เกิน 2

Complete Binary Tree

Complete Binary Tree หรือต้นไม้ไบนารีแบบสมบูรณ์ มีลักษณะคล้ายกับ Binary Tree แต่มีข้อพิเศษ คือ

1. ทุกโหนดที่ไม่ใช่ Leaf Node จะต้องมีโหนดลูก 2 โหนด
2. Leaf Node จะต้องอยู่ในระดับเดียวกัน

Binary Search Tree

Binary Search Tree มีลักษณะคล้ายกับ Binary Tree แต่มีลักษณะพิเศษเพิ่มเติม คือ

1. ค่าของรูทโหนดมีค่ามากกว่าค่าในต้นไม้ย่อยซ้าย (TL < R)
2. ค่าของรูทโหนดมีค่าน้อยกว่าหรือเท่ากับค่าในต้นไม้ย่อยขวา (R <= TR)

การสร้างและเพิ่มข้อมูลใน Binary Search Tree

1. ถ้า Binary Search Tree ยังไม่มีข้อมูล ให้โหนดที่เข้ามาใหม่เป็นรูทโหนดของ Binary Search Tree
2. ถ้า Binary Search Tree มีข้อมูลอยู่ ให้เพิ่มโหนดที่เข้ามาดังนี้

2.1 ถ้าค่าของโหนดใหม่ที่เข้ามา มากกว่า ค่าของรูทโหนด ให้เพิ่มโหนดใหม่ลงในต้นไม้ย่อยด้านขวา

2.2 ถ้าค่าของโหนดใหม่ที่เข้ามา น้อยกว่า ค่าของรูทโหนด ให้เพิ่มโหนดใหม่ลงในต้นไม้ย่อยด้านซ้าย

การลบข้อมูลใน Binary Search Tree

กรณีที่ 1 หากโหนดที่ต้องการลบเป็น Leaf Node สามารถลบได้ทันที

กรณีที่2 ถ้า โหนดที่ต้องการลบมีต้นไม้ย่อยเพียงด้านเดียว เมื่อลบโหนดที่ไม่ต้องการทิ้งแล้ว ให้นำค่ารูทโหนดของต้นไม้ย่อยไปแทนที่โหนดที่ลบทิ้งไป

กรณีที่3 ถ้าโหนดที่ต้องการลบมีต้นไม้ย่อยทั้งสองด้าน เมื่อลบโหนดที่ไม่ต้องการแล้ว มีวิธีให้เลือกทำอยู่ 3 วิธี คือ

1. นำค่าที่น้อยที่สุดของต้นไม้ย่อยขวาไปแทนที่โหนดที่ลบทิ้งไป
2. นำค่าที่มากที่สุดของต้นไม้ย่อยซ้ายไปแทนที่โหนดที่ลบทิ้งไป
3. นำค่ารูทโหนดของต้นไม้ย่อยขวาไปแทนที่โหนดที่ลบทิ้งไป แล้วนำค่ารูทโหนดของต้นไม้ย่อยซ้ายไปเป็นโหนดลูกทางซ้ายของโหนดที่มีค่าน้อย ที่สุดของต้นไม้ย่อยขวา

การท่องไปในทรี (Tree Traversal)

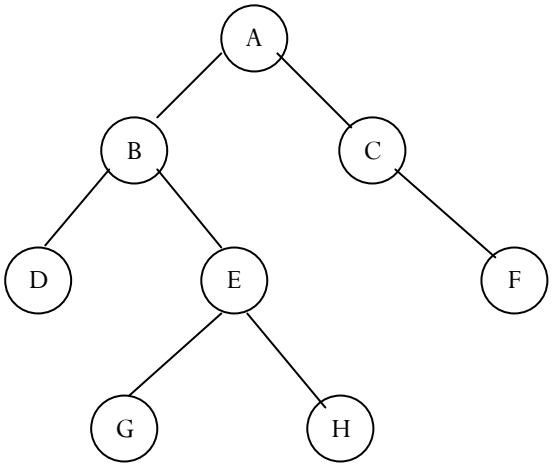
สามารถท่องเข้าไปในทรีเพื่อดูข้อมูล ได้ 3 วิธีด้วยกันคือ

1. Preorder
2. Inorder
3. Postorder

ในการท่องเข้าไปในทรีแต่ละแบบจะใช้สัญลักษณ์ดังนี้

Root = root node Left = ต้นไม้ย่อยซ้ายของ Root Right = ต้นไม้ย่อยขวาของ Root

1. จงตอบคำถามต่อไปนี้



1.1 โหนดรูทของต้นไม้คือ .....

1.2 โหนด G, H มีความสัมพันธ์กับ E อย่างไร .....

1.3 ต้นไม้นี้มีกี่ระดับ (Level) .....

1,4 โหนด D, G, H, F มีชื่อเรียกว่าอะไร .....

1,5 ดีกรีของต้นไม้นี้เป็นอย่างไร .....

1.6 ผลลัพธ์ของข้อมูลเมื่อเดินเข้าไปในโครงสร้างแบบ Pre-order คือ .....

.....

1.7 ผลลัพธ์ของข้อมูลเมื่อเดินเข้าไปในโครงสร้างแบบ In-order คือ .....

.....

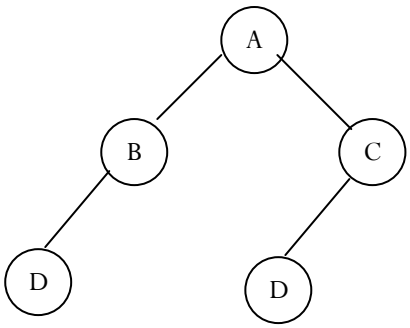
1.8 ผลลัพธ์ของข้อมูลเมื่อเดินเข้าไปในโครงสร้างแบบ Post-order คือ .....

.....

2. จงสร้าง Binary Tree จากลำดับของ postorder ที่กำหนดให้ ซึ่งกำกับด้วยค่าดีกรีแบบต้นไม้ไบนารี

A	D	C	F	E	B
0	1	1	0	2	1

3. จงวาดภาพ Inorder Threaded Binary tree สำหรับต้นไม้ต่อไปนี้



4. จงอธิบายถึงความสัมพันธ์ของโหนดต่างๆ ในโครงสร้างแบบต้นไม้

.....

.....

.....

.....

.....

.....

.....

.....

แบบฝึกหัดบทที่ 6

โครงสร้างข้อมูลแบบ GRAPH

กราฟเป็นโครงสร้างข้อมูลประเภทหนึ่งที่แสดงความสัมพันธ์ระหว่าง vertex และ edge กราฟจะประกอบด้วยกลุ่มของ vertex ซึ่งแสดงในกราฟด้วยสัญลักษณ์รูปวงกลม และ กลุ่มของ edge (เส้นเชื่อมระหว่าง vertex) ใช้แสดงถึงความสัมพันธ์ระหว่าง vertex หากมี vertex ตั้งแต่ 2 vertex ขึ้นไปมีความสัมพันธ์กัน ใช้สัญลักษณ์เส้นตรงซึ่งอาจมีหัวลูกศร หรือไม่มีก็ได้

กราฟสามารถเขียนแทนด้วยสัญลักษณ์ ดังนี้

G = (V , E) G คือ กราฟ V คือ กลุ่มของ vertex E คือ กลุ่มของ edge

ประเภทของกราฟ

แบ่งเป็น 2 ประเภทโดยแบ่งตามประเภทของ edge ได้ดังนี้

Direct Graph (กราฟแสดงทิศทาง) เป็นกราฟที่แสดงเส้นเชื่อมระหว่าง vertex โดแสดงทิศทางของการเชื่อมต่อด้วย

Undirected Graph กราฟที่แสดงเส้นเชื่อมต่อระหว่าง vertex แต่ไม่แสดงทิศทางของการเชื่อมต่อ Cyclic Graph กราฟที่มีเส้นเชื่อมต่อระหว่าง vertex ที่ทำให้ vertex มีลักษณะเป็นวงจรปิด (Cycle) เส้นเชื่อมต่อระหว่าง vertex อาจจะแสดงทิศทางหรือไม่แสดงทิศทางการเชื่อมต่อก็ได้

การแทนที่กราฟด้วยเมตริกซ์

โครงสร้างข้อมูลประเภทกราฟสามารถใช้เมตริกซ์มาแสดงแทนได้ โดยกราฟที่ประกอบด้วย vertex จำนวน N vertex สามารถแทนที่ด้วยเมตริกซ์ขนาด N x N โดยค่าในเมตริกซ์จะประกอบด้วย ค่า 0 และ 1

ค่า 0 จะใช้แทนไม่มี edge ความยาว 1 เชื่อมต่อจากต้นทางไปปลายทาง และ

ค่า 1 จะใช้แทนมี edge ความยาว 1 เชื่อมต่อจากต้นทางไปปลายทาง

จงตอบคำถามต่อไปนี้

1. ข้อใดหมายถึงเซตของเส้น

ก.) Point

ข.) Line

ค.) Edge

ง.) Arc

2. Vertice หมายถึงข้อใด

ก.) จุด

ข.) เส้น

ค.) โหนด

ง.) กราฟ

3. ข้อใดไม่เป็นเส้นทางไซเคิล

ก.) P(a,a) = (a,a)

ข.) P(b,b) = (b,c) (c,b)

ค.) P(b,b) = (b,c) (c,d)

ง.) P(d,d) = (d,b) (c,b)

4. ข้อใดไม่ใช่แนวทางการสร้างกราฟ

ก.) Adjacency Martrix

ข.) Node Directory

ค.) Multi-List

ง.) Out-Degree

5. แอปพลิเคชันการไหล หมายถึง ข้อใด

ก.) ปริมาณความจุ

ข.) โหนดของกราฟ

ค.) ท่อส่ง

ง.) น้ำหนักของเอจ

6. Breath-first Traversal หมายถึงข้อใด

ก.) การวิ่งตามแนวนอน

ข.) การวิ่งตามแนวกว้าง

ค.) การวิ่งตามแนวขนาน

ง.) การวิ่งตามแนวทแยง

7. ขั้นตอนแรกของ Breath – first คือข้อใด

ก.) วิ่งผ่านไปยังโหนด

ข.) สร้างคิวเก็บโหนด

ค.) ทำเครื่องหมายโหนด

ง.) เก็บโหนด

8. ขั้นตอนที่สองในการหาเส้นทางที่ดีที่สุด คือข้อใด

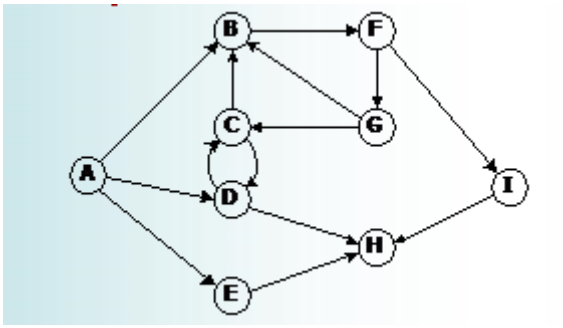
ก.) วิ่งผ่านไปยังโหนด N

ข.) กำหนดระยะทางเท่ากับ 0

ค.) สร้างคิวเก็บโหนด N

ง.) ลบโหนด N ออกจากคิว

จากรูป จงตอบคำถามข้อ 9-10



9. ในลำดับแรก ค่าใดจะถูกเก็บในคิว

ก.) A

ข.) B

ค.) D

ง.) E

10. เมื่อกิ่งโหนด E จากคิวเพื่อหาโหนดที่เชื่อมกันคือ H ในคิวจะเก็บข้อมูลใด

ก.) E,F,C,H

ข.) F,C,H

ค.) H,I,G

ง.) C,H,I,G

แบบฝึกหัดบทที่ 7

การเรียงลำดับข้อมูล

การเรียงลำดับข้อมูล (Sorting) เป็นเทคนิคที่สำคัญอย่างหนึ่งในกระบวนการประมวลผลข้อมูล ไม่ว่าจะเป็นการดำเนินงานทางด้านธุรกิจ วิทยาศาสตร์ หรือ สังคมศาสตร์ ก็ตาม ข้อมูลที่มีการเรียงกันเป็นที่เรียบร้อย ย่อมเป็นระเบียบ และง่ายต่อการค้นหา เช่น ระเบียนคนไข้ของโรงพยาบาล หรือรายชื่อหนังสือในหอสมุด หากข้อมูลไม่มีการเรียงกันเลย การค้นหาข้อมูลจะยากลำบากและต้องใช้เวลามาก หรืออาจไม่สามารถค้นหาได้เลย

การเรียงลำดับข้อมูลจะประสบความสำเร็จรวดเร็วแค่ไหน ขึ้นอยู่กับปัจจัย 3 ประการ ปัจจัยประการแรกขึ้นอยู่กับฮาร์ดแวร์ ฮาร์ดแวร์ที่มีความไวกว่าย่อมทำงานได้เร็วกว่าแน่นอน ซึ่งเราจะไม่สนใจปัจจัยนี้ เพราะไม่ใช้วิธีการแก้ปัญหา ที่ดีที่สุด ปัจจัยที่สอง ขึ้นอยู่กับจำนวนหรือขนาดของข้อมูล ถ้าข้อมูลมีเป็นจำนวนมาก การเรียงลำดับข้อมูลก็ย่อมเสียเวลาเพิ่มขึ้นเป็นธรรมดา แต่การเสียเวลาที่เพิ่มขึ้นนั้น เพิ่มขึ้นอย่างไร ? มีความสัมพันธ์อย่างไรกับจำนวนข้อมูลและสัมพันธ์ในลักษณะไหน สมมุติให้ N เป็นจำนวนข้อมูล เวลาที่ใช้ในการเรียงลำดับข้อมูล เป็นสัดส่วนโดยตรงกับ N^2 หรือที่เรียกว่า Big O(N^2) หมายความว่า เวลาที่ใช้ในการเรียงลำดับข้อมูลจะมากขึ้น ในลักษณะ กราฟพาราโบล่า (Parabola) ซึ่งจะมีการเปลี่ยนแปลงเร็วมาก โดยจะแปรผันเป็นกำลังสอง เช่น ถ้า N=10, N^2 จะเป็น 100 แต่ถ้า N=100, N^2 จะเป็น 10000. หรือถ้าเวลาในการเรียงลำดับข้อมูลขึ้นอยู่กับจำนวนข้อมูล ในลักษณะ Big O($N\log_2 N$) หมายความว่าจะมีความ สัมพันธ์ในลักษณะ N คูณ $\log_2 N$ ซึ่งการเพิ่มเวลา ในการเรียงลำดับข้อมูล จะช้ากว่าวิธีแรก แน่นอน

ความสัมพันธ์ที่กล่าวมาข้างต้น จะขึ้นอยู่กับวิธีการในการเรียงลำดับข้อมูล ซึ่งถือว่าเป็นปัจจัยที่สาน และเป็นปัจจัย ที่มีความสำคัญยิ่ง ในการเรียงลำดับข้อมูล อย่างมีประสิทธิภาพ ที่เราจะต้องศึกษาค้นคว้า และเป็นปัจจัยที่เราให้ความสนในที่สุด เพราะเป็นวิธี ที่ขึ้นอยู่กับความสามารถ ของนักโปรแกรมโดยตรง สำหรับในบทนี้ เราจะกล่าวถึงการเรียงลำดับข้อมูลในลักษณะ Big O(N^2) และบางวิธี ที่คิดว่าจะเป็นประโยชน์ต่อผู้เรียนโดยตรง วิธีการเรียงลำดับข้อมูลในลักษณะ Big O(N^2) แม้จะเรียงได้ช้ากว่า การเรียงลำดับข้อมูลในลักษณะวิธีการอื่น แต่ในข้อเท็จจริง วิธีการนี้ก็ไม่เลวนักและถือว่าเป็นวิธีที่ดีเลยที่เดียว สำหรับบางกรณี เช่น ถ้าข้อมูลไม่มากมายมหาศาล ยิ่งในปัจจุบัน ฮาร์ดแวร์ มีความสามารถรวดเร็วยิ่ง การประมวลผลแทบไม่มีความแตกแต่งกัน การเรียงวิธีนี้ถึงจะช้ากว่า แต่ถ้าข้อมูลอยู่ในลักษณะที่ ใกล้จะเรียงกัน หรือเรียงกันแล้ว วิธีนี้ยังสามารถตรวจสอบ และจบการทำงานก่อนเวลา ไม่เหมือนกับบางวิธี

ที่ถือว่าเร็วกว่า แม้ข้อมูลที่เรียงกันแล้วก็ไม่สามารถตรวจสอบได้และจะตั้งหน้าตั้งตาเรียงใหม่ตลอดเวลา ที่สำคัญอีกประการวิธีการนี้ ง่าย และสั้น ในการเขียนโปรแกรม

1. จงเปรียบเทียบสิ่งที่แตกต่างกัน ในการวิเคราะห์ประสิทธิภาพของ Internal Sort และ External Sort

.....

.....

.....

.....

.....

.....

2. จงเรียงลำดับข้อมูลต่อไปนี้ จากน้อยไปมากด้วยเทคนิค Insertion Sort

23, 43, 2, 4, 26, 8, 34, 16, 9, 20, 45

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. จงเรียงลำดับข้อมูลต่อไปนี้ จากน้อยไปมากด้วยเทคนิค Insertion Sort

314, 19, 15, 415, 336, 34, 112, 137, 82, 524, 350, 451, 213, 40

.....

.....

.....

.....

.....

.....

.....

4. จงเรียงลำดับข้อมูลในข้อ 3 ด้วยเทคนิค Radix Sort

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

5. จงเปรียบเทียบข้อดี-ข้อเสียของเทคนิค Insertion Sort และ Radix Sort

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

แบบฝึกหัดบทที่ 8

การค้นหาข้อมูล

การค้นหาข้อมูล (Searching) เป็นกระบวนการที่สำคัญยิ่ง ในการประมวลผลข้อมูลด้วยคอมพิวเตอร์ การค้นหาข้อมูลในปัจจุบัน ยังคงเป็นปัญหาและยังมีการศึกษาค้นคว้า ตัวอย่างการค้นหาข้อมูล ที่พอจะคุ้นเคย ได้แก่ การค้นหาเส้นทางที่สั้นที่สุดสำหรับการเดินทาง การค้นหาทางเดินของหุ่นยนต์ การค้นหาทางเดินของหมากรุก หรือ พัสซัล (Puzzle) ต่าง ๆ การค้นหาลักษณะของหลักการกลายพันธ์ (Genetic Algorithm) การค้นหาความเหมือนกันของรูปภาพ หรือการรู้จำ เป็นต้น สิ่งเหล่านี้ล้วนเป็นความท้าทายในการค้นหา นอกจากนี้ในปัจจุบันยังได้นำความรู้ทางด้านสถิติและระบบผู้เชียวชาญ มาช่วยในการค้นหา และการพยากรณ์เหตุการณ์ที่จะเกิดขึ้นในอนาคต เช่นในสาขาวิชา เดตามายนิ่ง (Data Mining) สำหรับการค้นหาที่จะศึกษาในบทนี้เป็นเพียงการค้นหาข้อมูลเบื้องต้น การค้นหาเพื่อหาระเบียน หรือ เขตข้อมูล ที่มีการจัดเก็บในแฟ้มข้อมูล หรือในแถวลำดับ มีหรือไม่ ซึ่งในการค้นหา จะต้องมีการกำหนด เขตข้อมูล เพื่อทำหน้าที่เป็นคีย์หลัก หรือดัชนี ในการค้นหา ซึ่งอาจเป็น รหัส ชื่อ-สกุล หรือเขตข้อมูลอื่นๆ ที่สนใจ และสามารถระบุระเบียนข้อมูลได้ การค้นหาจะหาได้รวดเร็ว หรือมีประสิทธิภาพเพียงใด ขึ้นอยู่กับหลักการ การค้นหาเป็นสำคัญ สำหรับขั้นตอนวิธีการค้นหาข้อมูลเบื้องต้น ที่จะกล่าวถึงนี้ คิดว่าจะเป็นประโยชน์ในการใช้งาน และเป็นพื้นฐานในการเรียนรู้ต่อไป

1. การตัดสินในเลือกเทคนิคการค้นหาข้อมูลมาใช้งาน ควรพิจารณาจากสิ่งใดบ้าง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. เทคนิค Linear Searching ต่างจากเทคนิค Hashing อย่างไรบ้าง

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. กำหนดให้โครงสร้างข้อมูลมีดังนี้ (7, 3, 9, 13, 16, 21, 32, 38, 43, 48, 51, 54, 61, 65, 69, 72, 75, 80)

3.1 จงค้นหาข้อมูลค่า 38 ด้วยเทคนิค Sequential Searching

3.2 จงค้นหาข้อมูลค่า 18 ด้วยเทคนิค Sequential Searching

3.3 จงค้นหาข้อมูลค่า 38 ด้วยเทคนิค Indexed Sequential Searching

3.4 จงค้นหาข้อมูลค่า 18 ด้วยเทคนิค Indexed Sequential Searching

3.5 จงค้นหาข้อมูลค่า 38 ด้วยเทคนิค Binary Searching

3.5 จงค้นหาข้อมูลค่า 18 ด้วยเทคนิค Binary Searching

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....