บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ผู้ทำการวิจัยได้ศึกษาแนวคิด ทฤษฎี หลักการ ตลอดจนเอกสารงานวิจัยที่เกี่ยวข้องเพื่อเป็นพื้นฐานความรู้รวมทั้งเป็นแนวทางสำหรับการทำวิจัย ดังนี้

แฮชชิง (Hashing)

แฮชชิง เป็นวิธีการเกี่ยวกับการจัดเก็บข้อมูลที่มีจำนวนมาก ๆ ไว้ในตาราง โดยตารางที่ใช้เก็บข้อมูลนั้นจะมีชื่อเรียกว่า ตารางแฮช (Hash Table) ส่วนเทคนิคที่ใช้หาตำแหน่งจัดเก็บข้อมูล จะเรียกว่า แฮชฟังก์ชัน (Hash Function) ซึ่งเป็นฟังก์ชันการคำนวณทางคณิตศาสตร์

โครงสร้างข้อมูลตารางแฮช (Hash Table Structure)

ตารางแฮช (Hash Table) คือ ตารางที่ใช้ในการเก็บข้อมูล ซึ่งภายในตารางจะมีช่องที่ใช้เก็บข้อมูลแต่ละตัวเอาไว้ โดยแต่ละช่องเก็บข้อมูลจะมีตำแหน่งที่อยู่ (Address) กำกับเอาไว้เพื่อใช้อ้างอิงตำแหน่งโครงสร้างข้อมูล ตารางแฮทสามารถสร้างบนโครงสร้างอาร์เรย์หรือลิงค์ลิสต์ก็ได้

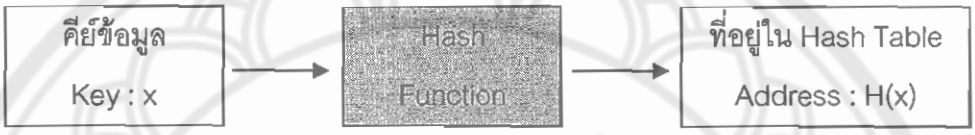
ลักษณะเด่นของโครงสร้างข้อมูลแบบตารางแฮช คือ สามารถเพิ่ม (Insert) ลบ (Delete) และค้นหา (Retrieve) ข้อมูลได้โดยใช้เวลาเฉลี่ยคงที่ (Constant Average Time) ซึ่งถือว่ามีประสิทธิภาพดีมากเมื่อเทียบกับโครงสร้างข้อมูลแบบอื่น (ขนิษฐา นามี, 2548)

หลักการพื้นฐานของแฮชชิง (Hashing Basic)

แฮชชิง (Hashing) คือ การแปลงค่าคีย์ข้อมูลให้กลายเป็นตำแหน่งที่อยู่เพื่อที่จะสามารถเก็บค่าคีย์ข้อมูลนั้น ๆ ลงในตารางแฮชได้ และสามารถที่จะค้นหาคีย์ข้อมูลนั้นในภายหลังได้ ดังภาพที่ 1 แสดงเทคนิควิธีที่นำมาใช้ในการแปลงค่าก็คือแฮชฟังก์ชัน ส่วนผลลัพธ์ของตำแหน่งที่อยู่ที่ได้จะแทนด้วยสัญลักษณ์ H(x) เมื่อ x คือ คีย์ข้อมูล (ขนิษฐา นามี, 2548) หรืออาจกล่าวได้ว่า หน้าที่หลักของการแฮชชิงก็คือ การพยายามหาตำแหน่งที่อยู่เพื่อเก็บข้อมูลลงไปในตารางให้ได้ ลักษณะของแฮชฟังก์ชันที่ดีก็คือ สามารถกระจายคีย์ข้อมูลที่ป้อนเข้ามาให้ไปอยู่

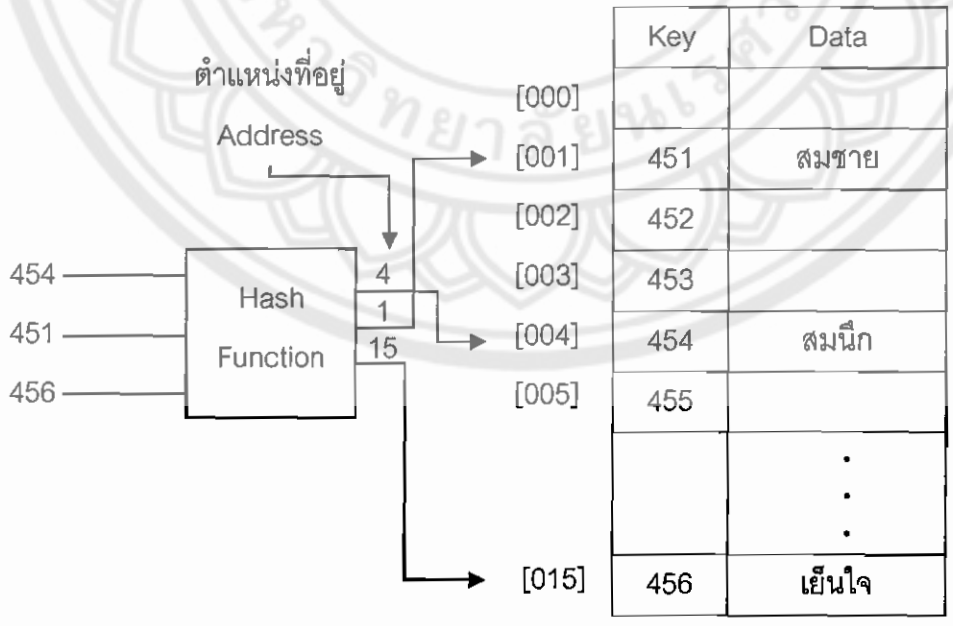
ในตำแหน่งที่อยู่ที่ไม่เกิดการชนกันเลยในตารางแฮช แต่ถ้าจำเป็นต้องชนกันจะต้องหาวิธีนำคีย์ข้อมูลตัวหลังที่ชนไปเก็บในตารางแฮชตำแหน่งอื่น ๆ ดังนั้นสามารถสรุปประเด็นหลัก ๆ ที่จำเป็นจะต้องคำนึงถึงในการออกแบบ เพื่อที่จะนำมาใช้กับโครงสร้างข้อมูลตารางแฮช ให้เกิดประสิทธิภาพสูงสุดได้ 2 ประเด็น คือ

1. ต้องเลือกใช้แฮชฟังก์ชันที่ดีเหมาะสมกับลักษณะข้อมูล
2. ต้องมีมาตรการมารองรับกรณีเกิดการชนกัน



ภาพ 1 หลักการพื้นฐานของแฮชชิง (Hashing)

สำหรับภาพ 2 เป็นการแสดงตัวอย่างลักษณะการทำงานทั่วไป ๆ ของโครงสร้างข้อมูลตารางแฮช ในตัวอย่างกำหนดให้สร้างตารางแฮช เพื่อเก็บข้อมูลของรายชื่อผู้ป่วยของโรงพยาบาลแห่งหนึ่ง คีย์ข้อมูลจะเป็นรหัสประจำตัวผู้ป่วย เมื่อผ่านแฮชฟังก์ชันออกมา จะเป็นตำแหน่งที่อยู่ที่จะนำข้อมูลไปเก็บไว้ที่ตารางแฮช



ภาพ 2 ตัวอย่างการทำแฮชชิงลงโครงสร้างข้อมูลตารางแฮช

การดำเนินการกับโครงสร้างข้อมูลตารางแฮช

การดำเนินการกับโครงสร้างข้อมูลตารางแฮช มีอยู่ด้วยกัน 3 อย่าง คือ

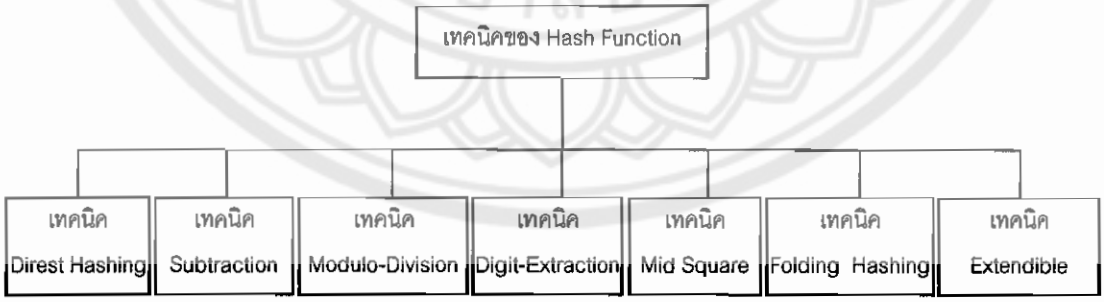
1. **การเพิ่มข้อมูล** (Insert) คือ การใส่ข้อมูลลงไปในตารางแฮช หรือการเพิ่มข้อมูลใหม่เข้าไปในฐานข้อมูล

2. การค้นหาข้อมูล (Retrieve) คือ การค้นหาข้อมูลที่ต้องการในตารางแฮชว่ามีข้อมูลที่ต้องการเก็บอยู่หรือไม่ และเก็บอยู่ ณ ตำแหน่งใดในตารางแฮช วิธีการในการค้นหาข้อมูลส่วนมากจะใช้หลักการเดียวกันกับการเพิ่มข้อมูล

3. การลบข้อมูล (Delete) คือ การค้นหาข้อมูลที่ต้องการขึ้นมาก่อน แล้วจึงลบข้อมูลนั้น เวลาที่ใช้ (Run Time) ในการลบข้อมูลจะใกล้เคียงกับเวลาที่ใช้ในการค้นหาข้อมูล

แฮชฟังก์ชัน (Hash Function)

เนื่องจากในการทำแฮชชิงนั้นการเลือกใช้แฮชฟังก์ชันเป็นสิ่งสำคัญส่งผลถึงประสิทธิภาพในการทำงานของโครงสร้างข้อมูล ดังนั้น ควรจะต้องเรียนรู้ถึงเทคนิคพื้นฐานต่าง ๆ ของแฮชฟังก์ชัน เพื่อที่จะสามารถเลือกเทคนิคที่เหมาะสมได้อย่างถูกต้องอันจะนำมาซึ่งการสืบค้นที่มีประสิทธิภาพสูงสุด เทคนิคพื้นฐานของแฮชฟังก์ชันมีด้วยกันหลายรูปแบบ ภาพ 3 แสดงตัวอย่างแฮชฟังก์ชัน ที่นิยมใช้งานกันในปัจจุบัน (ขนิษฐา นามี, 2548)



ภาพ 3 เทคนิคพื้นฐานของแฮชฟังก์ชัน

เทคนิค Direct Hashing

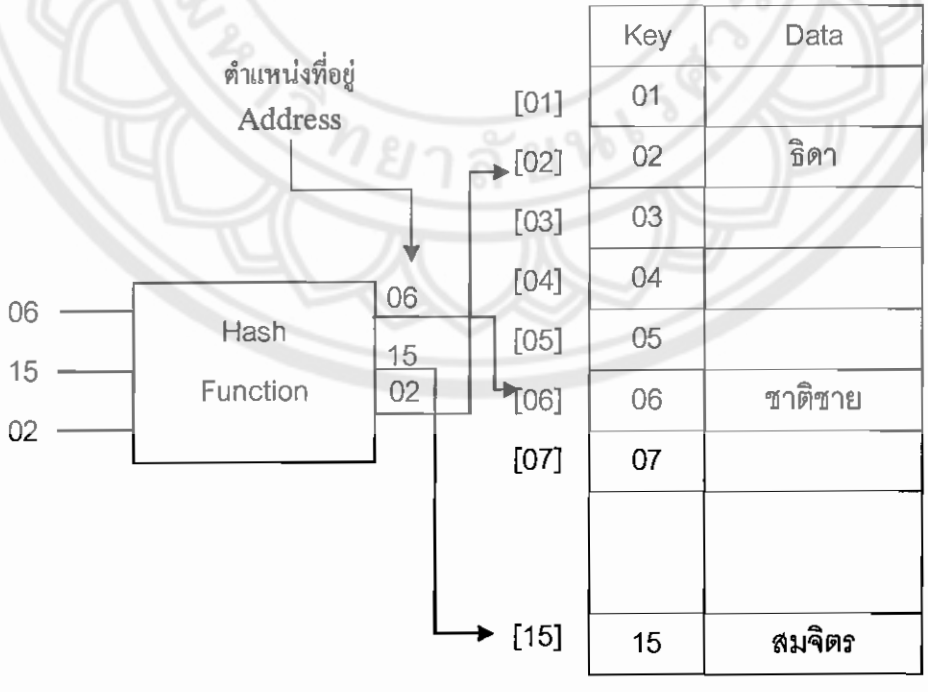
เทคนิค Direct Hashing **คือ การกำหนดให้คีย์ข้อมูลที่รับเข้ามามีค่าเป็นตำแหน่งที่อยู่เลย โดยไม่ต้องผ่านอัลกอริธึมใด ๆ ทั้งสิ้น ซึ่งถือว่าเทคนิคนี้เป็นเทคนิคที่ง่าย และมีประสิทธิภาพ** ในการทำงานสูงมาก เนื่องจากสามารถรับประกันได้ว่าจะไม่เกิดการชนกันของคีย์ข้อมูลแน่นอน แต่จะมีข้อจำกัดในเรื่องคีย์ข้อมูลคือ จะต้องเป็นคีย์ข้อมูลที่มีขอบเขตที่ชัดเจน นอกจากนั้นเทคนิคนี้ยังเหมาะสมกับคีย์ข้อมูลที่มีค่ากระจายระหว่างคีย์ไม่มากจนเกินไป และความแตกต่างระหว่างปริมาณข้อมูลกับพื้นที่ของตารางไม่ควรแตกต่างกันมากจนเกินไป เช่น ถ้าคีย์ข้อมูลมีค่าตั้งแต่ 0000 – 1000 แต่ปริมาณข้อมูลที่จะเก็บมีประมาณ 250 ข้อมูล ก็ถือว่าการใช้เทคนิคนี้จะไม่เหมาะสม เนื่องจากมีพื้นที่ว่างในตารางแฮชมากจนเกินไป ส่งผลให้สูญเสียเนื้อที่ในการเก็บข้อมูลมากโดยใช่เหตุ ดังในภาพ 4 แสดงการใช้เทคนิค Direct Hashing ในการเก็บข้อมูลโดยกำหนดให้คีย์ข้อมูลมีค่าระหว่าง 01 – 15 ดังนั้น ตำแหน่งที่อยู่จะมีค่าเป็น 01 – 15 เช่นกัน

สมการแฮชฟังก์ชันแบบ Direct Hashing มีสมการดังนี้

แฮชฟังก์ชัน : ตำแหน่งที่อยู่ = คีย์ข้อมูล

หรือ

H(x) = x



ภาพ 4 เทคนิค Direct Hashing

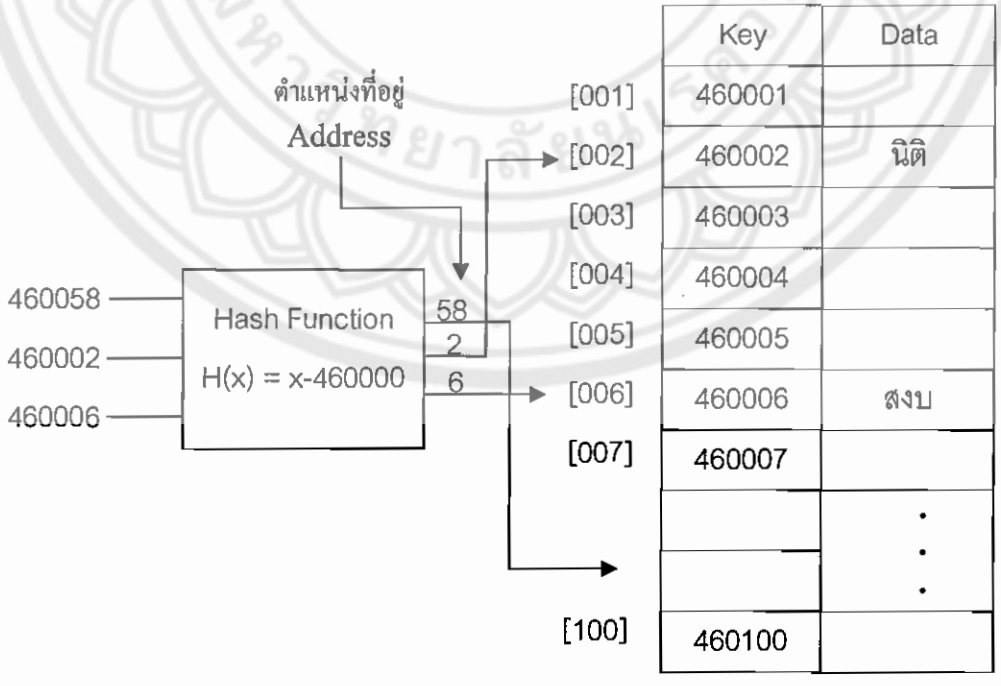
เทคนิค Subtraction

เทคนิค Subtraction มีลักษณะการทำงานที่คล้ายกันกับเทคนิค Direct Hashing ต่างกันเพียงแค่เทคนิค Subtraction ใช้สำหรับกรณีที่คีย์ข้อมูลไม่ได้เริ่มตั้งแต่ค่า 01 แต่เริ่มต้นที่ค่าอื่น วิธีแก้ก็คือ หาค่าคงที่ค่าหนึ่งมาหักลบออกไปเท่านั้นเอง ซึ่งก็จะถือว่าแฮชฟังก์ชันที่ใช้คือการนำค่าคงที่มาหักลบออกจากค่าคีย์ข้อมูล ก็จะได้ค่าตำแหน่งที่อยู่ในตารางออกมาเทคนิคนี้ก็ถือได้ว่าเทคนิคที่ง่ายและมีประสิทธิภาพในการทำงานอีกเทคนิคหนึ่ง เนื่องจากสามารถรับประกันได้ว่าจะไม่มีการชนกันของคีย์ข้อมูลเกิดขึ้น ตัวอย่างดังภาพที่ 5 แสดงการเก็บรายชื่อผู้ป่วย กำหนดให้คีย์ข้อมูลคือ รหัสประจำตัวผู้ป่วยซึ่งมีค่าตั้งแต่ 460001 – 460100 ดังนั้น กรณีนี้แฮชฟังก์ชัน ก็คือ การนำ 460000 มาลบออกนั่นเอง ก็จะได้ค่าตำแหน่งที่อยู่ที่มีค่าระหว่าง 1 - 100 สมการแฮชฟังก์ชัน สามารถเขียนได้ดังนี้

แฮชฟังก์ชัน : ตำแหน่งที่อยู่ = คีย์ข้อมูล - 460000

หรือ

H(x) = x - 460000



ภาพ 5 ตัวอย่างเทคนิค Subtraction

เทคนิค Modulo Division

เทคนิค Modulo Division **เป็นเทคนิคที่นิยมใช้กันมาก เนื่องจากมีความยืดหยุ่นสูง วิธีการทำงานของ เทคนิค** Modulo-Division **คือ การหาตำแหน่งที่อยู่โดยการนำค่าคีย์ข้อมูลมา** หารแบบ Modulo ด้วยขนาดของตาราง (การหารแบบ Modulo หรือ MOD คือการหารโดย ผลลัพธ์ที่ได้คือ ค่าเศษที่เหลือจากการหาร เช่น 10 MOD 3 = 1) โดยปกติแล้วนิยมกำหนดให้ ขนาดของตารางมีค่าเป็นจำนวนเฉพาะ (Prime Number) เพราะจะส่งผลให้เกิดการชนกันของคีย์ ข้อมูลน้อยลง เช่น ตารางแฮชมีค่า 10 ช่อง ก็มักจะเลือกใช้ขนาดตารางเท่ากับ 7 แฮชฟังก์ชัน ของเทคนิคนี้สามารถสร้างได้ 2 รูปแบบคือ

1. ค่าของตำแหน่งที่อยู่ได้มาจาก ค่าคีย์ข้อมูล Modulo ด้วยขนาดของตารางที่เป็นค่า จำนวนเฉพาะ ผลลัพธ์ของค่าตำแหน่งที่อยู่ที่ได้จะมีค่าระหว่าง {000 ถึง (ขนาดตาราง – 1)} แฮชฟังก์ชันของรูปแบบนี้สามารถเขียนได้ดังสมการ

แฮชฟังก์ชัน : ตำแหน่งที่อยู่ = คีย์ข้อมูล MOD ขนาดตาราง

หรือ

H(x) = x MOD ขนาดตาราง

2. ค่าของตำแหน่งที่อยู่ได้มาจาก ค่าคีย์ข้อมูล Modulo ด้วยขนาดของตารางที่เป็นค่า จำนวนเฉพาะเช่นกัน แล้วบวกเพิ่มเข้าไปอีก 1 รูปแบบนี้ผลลัพธ์ของค่าตำแหน่งที่อยู่ที่ได้จะมีค่า ระหว่าง {001 ถึง ขนาดตาราง} แฮชฟังก์ชันของรูปแบบนี้สามารถเขียนได้ดังสมการ

แฮชฟังก์ชัน : ตำแหน่งที่อยู่ = (คีย์ข้อมูล MOD ขนาดตาราง) + 1

หรือ

H(x) = (x MOD ขนาดตาราง) + 1

โดยรูปแบบที่ 1 และ 2 ตารางที่ได้ออกมาจะมีลักษณะคล้ายกันมาก ต่างกันเพียง ตำแหน่งที่อยู่ของข้อมูล กล่าวคือ รูปแบบที่ 1 ตำแหน่งที่อยู่ของข้อมูลจะเริ่มต้นที่ 000 ส่วน รูปแบบที่ 2 ตำแหน่งที่อยู่ของข้อมูลจะเริ่มต้นที่ 001

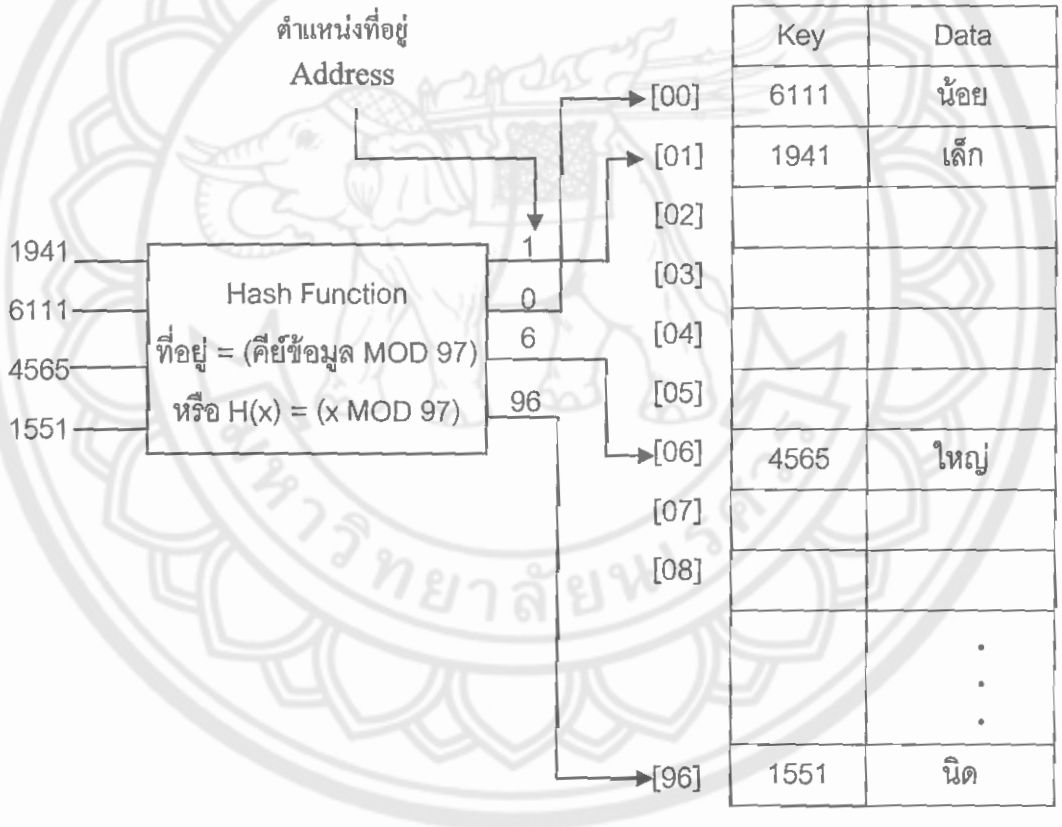
ตัวอย่างการหาค่าตำแหน่งที่อยู่ในตารางแฮชแบบเทคนิค Modulo Division **โดย กำหนดให้ต้องการสร้างตารางแฮชสำหรับเก็บชื่อของผู้ป่วย โดยต้องการตารางขนาดประมาณ 100 ช่องข้อมูล แต่เนื่องจากขนาดตารางควรเป็นค่าจำนวนเฉพาะ ดังนั้น จึงเลือกขนาดตารางที่ 97 ช่องข้อมูล ถ้าต้องการสร้างแฮชฟังก์ชันตามรูปแบบที่ 1 จะได้แฮชฟังก์ชันดังสมการ**

H(x) = x MOD 97

ดังแสดงผลของฟังก์ชันในตาราง 1 และภาพ 6

ตาราง 1 แสดงการหาค่าตำแหน่งที่อยู่ในตารางแฮชแบบที่ 1

คีย์ข้อมูล	ที่อยู่
1941	H(1941) = 1941 MOD 97 = 1
6111	H(6111) = 6111 MOD 97 = 0
4565	H(4565) = 4565 MOD 97 = 6
1551	H(1551) = 1551 MOD 97 = 96



ภาพ 6 ตัวอย่างเทคนิค Modulo Division ตามรูปแบบที่ 1

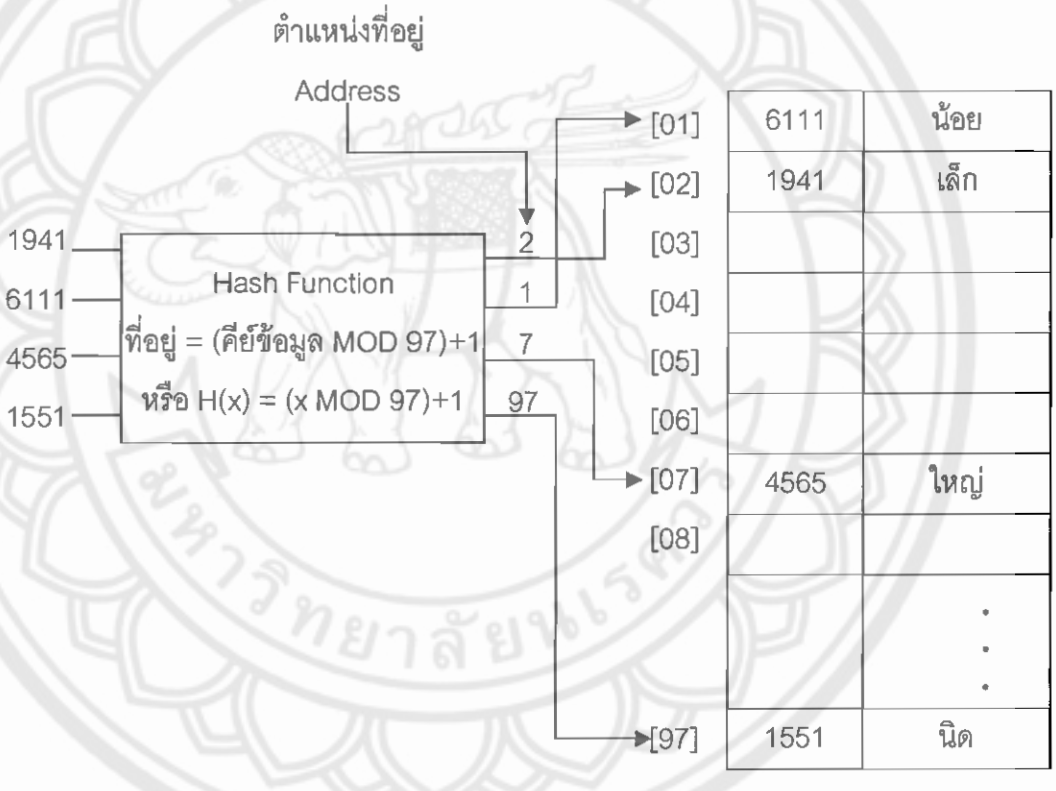
ถ้าต้องการสร้างแฮชฟังก์ชันตามรูปแบบที่ 2 จะได้แฮชฟังก์ชันดังสมการ

H(x) = (x MOD 97) + 1

และแสดงผลของฟังก์ชัน ดังตาราง 2 และภาพ 7

ตาราง 2 แสดงการหาค่าตำแหน่งที่อยู่ในตารางแฮชแบบที่ 2

คีย์ข้อมูล	ที่อยู่
1941	(1941 MOD 97) + 1 = 2
6111	(6111 MOD 97) + 1 = 1
4565	(4565 MOD 97) + 1 = 7
1551	(1551 MOD 97) + 1 = 97



ภาพ 7 ตัวอย่างเทคนิค Modulo Division ตามรูปแบบที่ 2

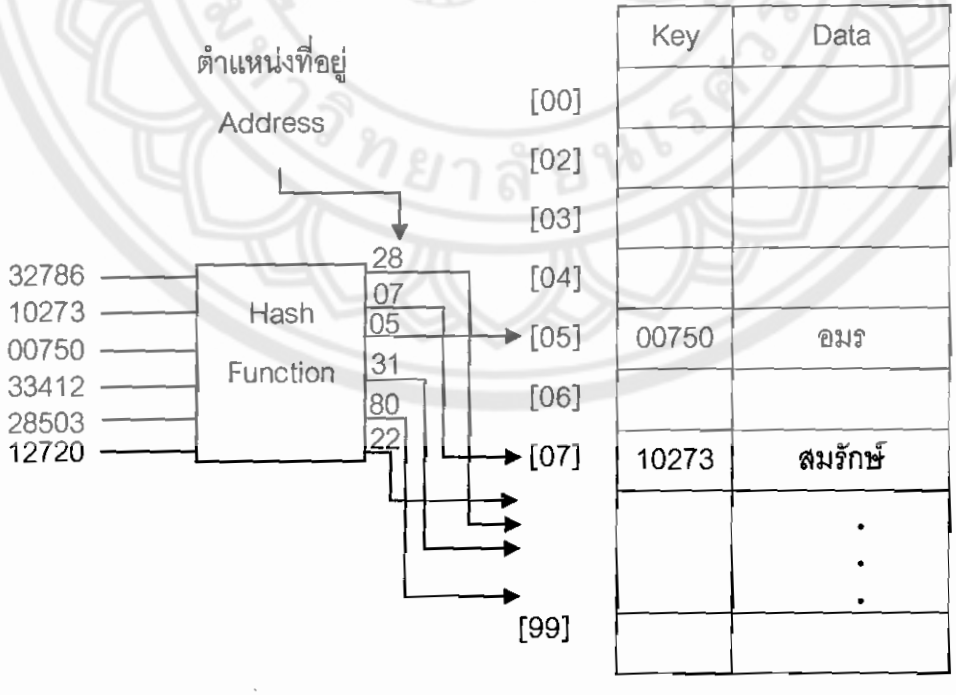
เทคนิค Digit Extraction

เทคนิค Digit Extraction คือ การนำตัวเลขคีย์ข้อมูลมาเลือกเพียงบางตำแหน่งเพื่อใช้เป็นตำแหน่งที่อยู่ในตารางแฮช โดยที่ตำแหน่งตัวเลขที่เลือกนั้นจะต้องเป็นตำแหน่งที่มีตัวเลขซ้ำกันน้อยที่สุด เพื่อลดปัญหาการชนกันของตำแหน่งที่อยู่ ตัวอย่างเช่น กำหนดให้คีย์ข้อมูลที่แฮชชิง

มีค่า {32786, 10273, 01750, 33412, 28503, 12720} เมื่อพิจารณาค่ากลุ่มคีย์ข้อมูลแล้วตัวเลขคีย์ข้อมูลที่ซ้ำกันน้อย ๆ จะอยู่ตำแหน่งที่ 2 และ 4 ดังนั้น จึงนำตัวเลขในตำแหน่งที่ 2 และ 4 มาจับคู่ เพื่อกำหนดเป็นตัวเลขที่อยู่ในตาราง (ในรูปแสดงผลการจับคู่ได้ออกมาเป็นตำแหน่งที่ 28, 07, 15, 31, 80 และ 22) ดังแสดงในตาราง 3 และภาพ 8

ตาราง 3 แสดงผลการจับคู่ตัวเลขคีย์ข้อมูลที่ซ้ำกันน้อยๆ

ตำแหน่งที่ :	1	2	3	4	5
คีย์ข้อมูล :	3	2	7	8	6
	1	0	2	7	3
	0	0	7	5	0
	3	3	4	1	2
	2	8	5	0	3
	1	2	7	2	0



ภาพ 8 ตัวอย่างเทคนิค Digit Extraction

เทคนิค Mid Square

หลักการของเทคนิค Mid Square คือ การนำคีย์ข้อมูลมายกกำลังสอง จากนั้นเลือกตำแหน่งตรงกลางมาใช้เป็นตำแหน่งที่อยู่ในตารางแฮช เช่น ถ้าผลของค่ายกกำลังออกมาเป็น 145673 อาจจะเลือกตัวเลขตำแหน่งที่ 3 และ 4 มาใช้ นั่นคือ ได้ค่าตำแหน่งที่อยู่เป็น 56 เป็นต้น ตาราง 4 แสดงตัวอย่างการทำงานของเทคนิคนี้ ค่าคีย์ข้อมูลประกอบด้วย {312, 124, 542, 056} ซึ่งในที่นี้กำหนดให้ตำแหน่งที่อยู่ คือ ตัวเลขตำแหน่งที่ 3 4 และ 5 ของผลลัพธ์ที่ได้จากการคำนวณด้วยตารางแฮช สำหรับตัวอย่างนี้จะมีค่าตำแหน่งที่อยู่ตั้งแต่ 000 – 999

ตาราง 4 แสดงการทำงานของเทคนิค Mid Square

คีย์ข้อมูล	ยกกำลังสอง	ผลลัพธ์ของเลขยกกำลัง	ตำแหน่งที่อยู่ (เลือกตำแหน่งที่ 3, 4, 5)
312	312^2	097344	734
124	124^2	015376	537
542	542^2	293764	376
056	056^2	003136	313

แต่เนื่องจากเทคนิคนี้มีข้อจำกัดอยู่ตรงที่ คีย์ข้อมูลจะต้องไม่มีค่ามากจนเกินไป เพราะถ้าคีย์ข้อมูลมีค่ามากเมื่อนำมายกกำลังสองแล้ว ค่าผลลัพธ์ที่ได้จะยิ่งมีค่ามากขึ้นไปอีก เช่น ถ้าคีย์ข้อมูลที่มีขนาด 7 หลัก เมื่อนำมายกกำลังสองก็จะมีขนาดสูงถึง 14 หลัก ซึ่งจะส่งผลกระทบต่อการเก็บข้อมูลในหน่วยความจำ เนื่องจากคอมพิวเตอร์ส่วนบุคคลส่วนใหญ่จะเก็บเลขจำนวนเต็มได้ 9 หลัก จึงเพียงพอสำหรับคีย์ข้อมูลที่มีขนาดสูงสุด 4 หลักเท่านั้น เพื่อแก้ไขข้อจำกัดนี้ เทคนิค Mid Square จึงเพิ่มทางเลือก โดยกรณีที่คีย์ข้อมูลมีขนาดใหญ่ให้เลือกตำแหน่งเพียงบางตำแหน่งในคีย์ข้อมูลมายกกำลังสอง จากนั้นจึงเลือกตำแหน่งตรงกลางของผลลัพธ์มาเป็นตำแหน่งที่อยู่ในตารางแฮช ตัวอย่างตาราง 5 กำหนดให้ค่าที่เลือกออกมาจากคีย์ข้อมูลเพื่อยกกำลังสองคือ ตำแหน่งที่ 1 2 และ 3 ส่วนตำแหน่งที่อยู่ที่เลือกออกมาจากผลลัพธ์ของเลขยกกำลังคือ ตำแหน่งที่ 3 4 และ 5

ตาราง 5 แสดงการทำงานของเทคนิค Mid Square ที่เปลี่ยนแปลง

คีย์ข้อมูล	ค่าที่เลือก	ยกกำลังสอง	ผลลัพธ์ของเลขยกกำลัง	ตำแหน่งที่อยู่
1356424	135	135^2	018225	822
2654121	265	265^2	070225	022
3642132	364	364^2	132496	249
3251324	325	325^2	105625	562

เทคนิค Folding Hashing

หลักการของเทคนิค Folding Hashing คือ การกำหนดค่าคีย์ที่รับมาแบ่งจำนวนของตัวเลขตามจำนวนแอดเดรสแล้วรวมค่าคำนวณได้ตัวเลขตามที่ต้องการ โดยมีวิธีการหาแอดเดรสของคีย์อยู่ 2 วิธี (วิวัฒน์ อภิสิทธิ์ภิญโญ และ อมร มุสิกสาร,2542) คือ

1. วิธี Fold Shift ใช้การแบ่งจำนวนคีย์รวมจำนวนเมื่อได้ค่าตำแหน่งแอดเดรสที่เกินให้ตัดออก เช่น รับค่าคีย์ 123456789 ต้องการแอดเดรสจำนวน 3 ตำแหน่ง คือ 000 – 999 ได้ค่าแอดเดรสด้วยการแบ่งตัวเลขคีย์เป็น

1 2 3
4 5 6
7 8 9
รวมค่า 1 3 6 8 ทำการตัดค่าตัวเลขที่เกินแอดเดรสออก

คือ 1 แอดเดรสที่ต้องการของคีย์นี้คือ 3 6 8

2. วิธี Fold Boundary ใช้รูปแบบหาค่าแอดเดรส เช่นเดียวกับวิธี Fold Shift แต่จะทำการแบ่งคีย์ด้วยการกลับตำแหน่ง เช่น ค่าคีย์ 123456789 ได้ค่าแอดเดรสด้วยการแบ่งตัวเลขคีย์เป็น

3 2 1 (กลับค่า 1 2 3)
4 5 6 (ตำแหน่งกลางคงเดิม)
9 8 7 (กลับค่า 7 8 9)
รวมค่า 1 7 6 4 ทำการตัดค่าตัวเลขที่เกินแอดเดรสออก

คือ 1 แอดเดรสที่ต้องการของคีย์นี้คือ 7 6 4

เทคนิค Extendible Hashing

เทคนิคที่กล่าวมาข้างต้นทั้งหมด เป็นการแฮชชิงแบบ Static นั่นคือ ตารางแฮชจะมีขนาดคงที่ ซึ่งจะต้องกำหนดไว้ล่วงหน้าและหากข้อมูลมีการเกาะกลุ่มกันในตารางแฮช อาจทำให้เกิดการชนกันได้มาก และในที่สุดอาจต้องทำการปรับโครงสร้างทั้งหมด (Reorganization) ซึ่งทำให้เสียเวลาในการดำเนินการมาก เทคนิคต่อไปนี้จะเป็นการจัดการตารางแบบ Dynamic เรียกว่าเทคนิคเอ็กเทนดิเบิล แฮชชิง

หลักการของเทคนิคเอ็กเทนดิเบิล แฮชชิง เป็นวิธีการทำแฮชชิงอีกรูปแบบหนึ่ง ซึ่งการแฮชชิงด้วยวิธีนี้เหมาะสำหรับกรณีที่มีข้อมูลเป็นจำนวนมาก ๆ จนไม่สามารถที่จะเก็บในหน่วยความจำหลัก (Main Memory) ที่มีขนาดจำกัดได้หมดในระหว่างที่ประมวลผล ในข้อมูลที่มีปริมาณมากๆปัญหาหลักที่ตามมาคือ การแก้ปัญหาการชนกันของข้อมูล และทำให้ต้องเสียเวลาสูงมากในการค้นหาข้อมูล นอกจากนั้นถ้าตารางแฮชมีข้อมูลเก็บอยู่ใกล้จะเต็มก็ต้องมีการทำรีแฮชชิง (Rehashing) ซึ่งเป็นขั้นตอนที่เสียเวลามากที่สุด เพราะต้องใช้เวลาในการดำเนินการมีค่า O(n) วิธีเอ็กเทนดิเบิล แฮชชิงสามารถเข้าถึงข้อมูลได้อย่างรวดเร็ว ช่วยลดเวลาที่ต้องใช้ในการเพิ่มและการค้นหาข้อมูลลงได้มาก การแฮชชิงด้วยวิธีนี้จะใช้หลักการของ B-tree เข้ามาช่วยวิธีการเก็บข้อมูลของโครงสร้างเอ็กเทนดิเบิล แฮชชิง มีลักษณะดังภาพ 9 ซึ่งมีคุณสมบัติดังนี้

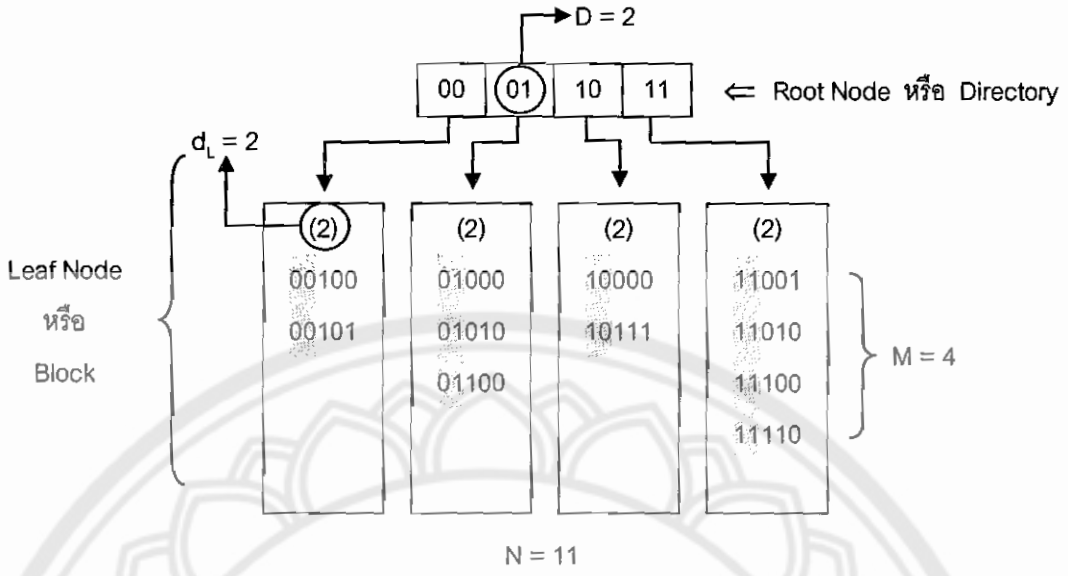
- โครงสร้างข้อมูลจะแบ่งออกเป็น 2 ส่วนคือ ส่วนที่เป็นรูทโหนด (Root Node) ซึ่งขณะประมวลผลจะเก็บไว้ในหน่วยความจำหลักจะทำหน้าที่เหมือนเป็นดรรชนี (Index) ชี้บอกตำแหน่งข้อมูล และส่วนที่เป็นลีฟโหนด (Leaf Node) เป็นส่วนที่ใช้เก็บข้อมูล

- รูทโหนดเป็นส่วนที่เป็นตัวบ่งบอกเพื่อใช้แบ่งแยกลีฟโหนดแต่ละอันออกจากกันให้ชัดเจน โดยจะมีตัวเชื่อมชี้ไปยังลีฟโหนดที่สัมพันธ์กับตัวมัน

- ลีฟโหนดเป็นส่วนที่ใช้เก็บข้อมูล โดยมีโครงสร้างข้อมูลที่มีลักษณะเป็นกล่อง ในที่นี้จะขอเรียกว่าบล็อค (Block)

- แต่ละบล็อคมีขนาดเท่า ๆ กัน และสามารถเก็บข้อมูลได้จำนวนเท่า ๆ กันส่วนมากใน**การเพิ่มข้อมูลเข้าไปในบล็อคแต่ละบล็อค จะมีการเรียงลำดับข้อมูลจากน้อยไปมากด้วย เพื่อช่วยให้รวดเร็วยิ่งขึ้นในการค้นหาข้อมูล**

- ถ้ามีการเพิ่มข้อมูลเพิ่มในบล็อคที่เต็มแล้ว ก็สามารถที่จะแตกบล็อคเพิ่มออกมาได้หรือที่เรียกว่าการแบ่ง (Split)



ภาพ 9 ลักษณะของ Extendible Hashing

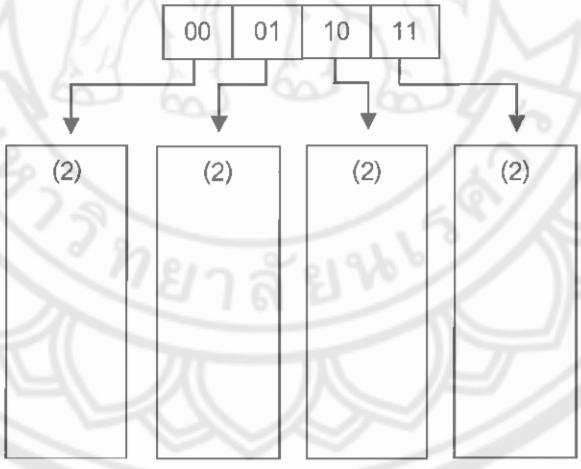
ภาพ 9 แสดงตัวอย่างการเก็บข้อมูลแบบเอ็กเทนดิเบิล แฮชชิง โดยจะเป็นตัวอย่างการเก็บข้อมูลของเลขจำนวนเต็ม 5 บิต ในที่นี้สมมติให้ข้อมูลมีจำนวนทั้งสิ้น N ข้อมูล และใน 1 Block สามารถเก็บข้อมูลได้สูงสุด M ข้อมูล โดยแต่ละ Block จะแบ่งแยกกันด้วยเลขบิตสองตัวแรกที่เหมือนกัน ทำหน้าที่เหมือนเป็น Index เลขบิตที่ใช้แบ่งแยกจะถูกเก็บในรูทโหนด กำหนดให้ D แทนจำนวนบิตที่ใช้ในรูทโหนด หรือบางครั้งจะเรียกว่าไดเร็คทอรี (Directory) ส่วนจำนวนบิตของข้อมูลที่เหมือนดรรชนีในแต่ละบล็อค (ในรูปคือ บิตที่อยู่บนแถบทึบ) แทนด้วย d_L ดังนั้น ในภาพ 9 D มีค่าเท่ากับ 2 และ d_L ก็มีค่าเท่ากับ 2 เช่นกัน

ขนิษฐา นามี (2548) กล่าวว่า ความสัมพันธ์ของค่าต่าง ๆ ในเอ็กเทนดิเบิล แฮชชิง สามารถสรุปได้ดังนี้

- จำนวนของสมาชิกในไดเร็คทอรีมีทั้งสิ้น 2^D คือภาพ 9 ไดเร็คทอรีมีสมาชิก = 2^2 = 4
- ในแต่ละบล็อคค่าของ $d_L \leq D$
- ค่าสูงที่สุดที่เป็นไปได้ของ D คือ (จำนวนบิตของข้อมูล $-\sqrt{M}$) เทียบกับภาพ 9 ค่าสูงสุดของ D คือ $(5-\sqrt{4})$ = 3
- **เวลาที่ใช้ในการประมวลผล คือ** $O(\log_{\frac{M}{2}} N)$
- **จำนวนของบล็อกที่เหมาะสม คือ** $\left(\frac{M}{N}\right)\log_2 e$
- **ขนาดของไดเร็คทอรีที่เหมาะสม คือ** $O\left(\frac{N}{M}^{\frac{1+1}{M}}\right)$

การแฮชชิงข้อมูลเข้าไปในเอ็กเทนดิเบิล แฮชชิง มีหลักการ คือ จะเพิ่มข้อมูลเข้าไปในบล็อคทีละค่า โดยพิจารณาเลือกบล็อคที่จะเพิ่มข้อมูลเข้าไปจากบิต d_L ที่ตรงกันในไดเร็คทอรี ถ้าบล็อคที่จะเพิ่มข้อมูลเข้าไปมีข้อมูลอื่นเก็บอยู่จนเต็ม ก็ให้แบ่งบล็อค (Split Block) นั้นออกมา ส่วนเรื่องการค้นหาข้อมูล สามารถกระทำได้อย่างรวดเร็ว คือ จะทำงาน (Access) เพียง 2 ครั้งก็จะทราบคำตอบว่าพบข้อมูลนั้น ๆ หรือไม่ โดยครั้งแรกจะทำงานตรงไดเร็คทอรีที่มี d_L ตรงกับข้อมูลที่ต้องการ จากนั้นทำงานครั้งที่สองจะค้นหาข้อมูลในที่เชื่อมบล็อคไว้ ดังอย่าง การเพิ่ม (Insert) ข้อมูลเลขจำนวนเต็มขนาด 8 บิตลงในเอ็กเทนดิเบิล แฮชชิง กำหนดให้ M มีค่าเท่ากับ 4 {10110111, 01010011, 00010111, 11010000, 01000111, 11000111, 01000001, 10110001, 11110111, 11101000, 10100110, 10000011, 01011011, 11100000, 11001100, 10000111}

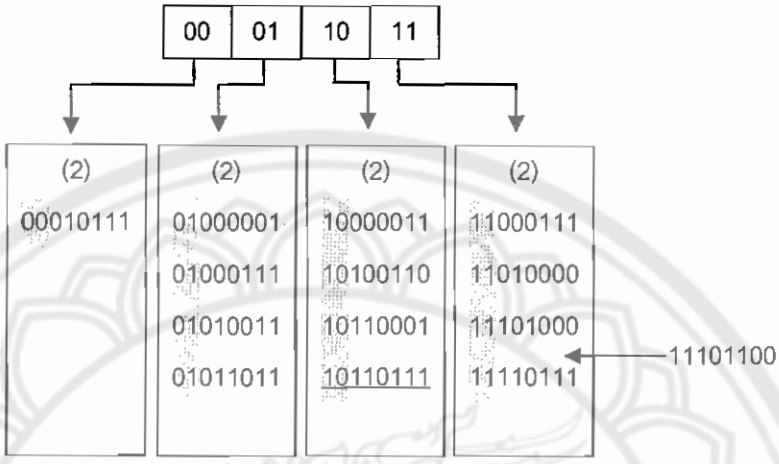
1. เริ่มต้นจะสร้างข้อมูลเอ็กเทนดิเบิล แฮชชิงให้มีขนาด M = 4, D = 2, d_L= 2 ก่อนแล้วจะแบ่งบล็อคออกภายหลังถ้าบล็อคใดเต็ม



ภาพ 10 แสดงการเริ่มต้นจะสร้างข้อมูล Extendible Hashing

2. ให้เพิ่มข้อมูลเข้าไปในบล็อคทีละค่า เริ่มจาก 10110111 วิธีดูว่าเพิ่มเข้าไปในบล็อคไหนก็คือ ดูว่าบิตที่เป็น d_L ในข้อมูลมีค่าเป็นอะไรในที่นี้ก็คือ มีค่าเป็น 10 จากนั้นก็เทียบดูว่าในส่วนของไดเร็คทอรีตำแหน่งไหนมีค่าเป็น 10 จากนั้นก็เพิ่มข้อมูลเข้าไปในบล็อคที่ 10 ชี้อยู่ ซึ่งใน

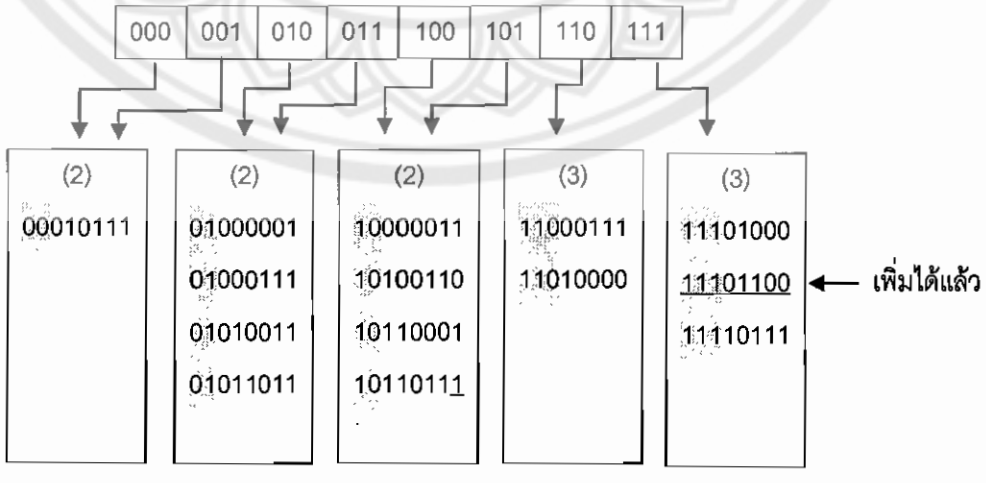
ที่นี้ก็คือ ข้อมูลบล็อคที่ 3 ให้ทำการเพิ่มข้อมูลค่าบล็อคที่ 3 ก็คือว่าเพิ่มข้อมูลเสร็จไปแล้วหนึ่งค่า ให้ทำการเพิ่มข้อมูลไปเรื่อย ๆ จนกว่าจะมีบล็อคใดบล็อคหนึ่งเต็มจึงจะต้องแบ่งบล็อค



ภาพ 11 แสดงการเพิ่มข้อมูลเข้าไปในบล็อค

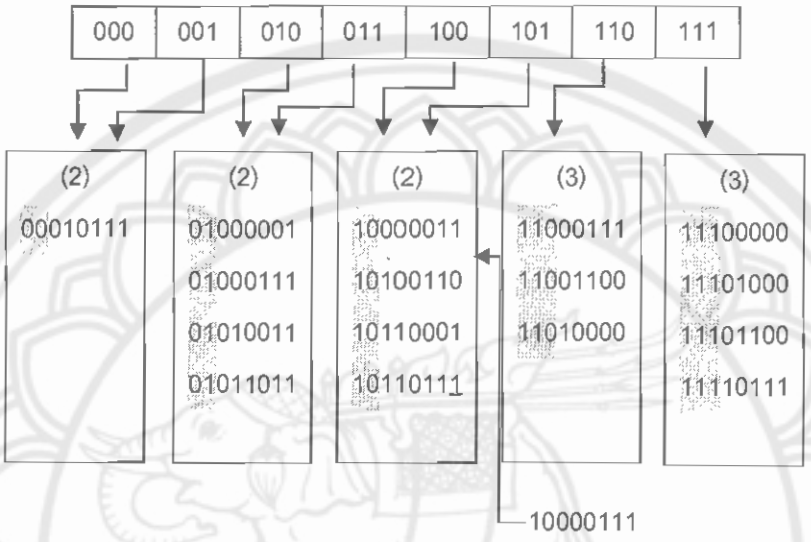
สามารถเพิ่มข้อมูล 10110111, 01010011, 00010111, 11010000, 01000111, 11000111, 01000001, 10110001, 11110111, 11101000, 10100110, 01011011 ลงไปได้ แต่พอจะเพิ่มข้อมูล 111011100 ลงในบล็อคสุดท้ายจะต้องแบ่งบล็อคออก เนื่องจากบล็อคเต็ม

3. การแบ่งบล็อคก็คือ การเพิ่มค่าของ D ขึ้นไปอีกหนึ่งค่า เดิม D มีค่า = 2 เมื่อแบ่งบล็อค D จึงมีค่าเป็น 3 ดังนั้นไดเร็คทอรีก็จะเพิ่มตามค่าของ D จึงกลายเป็นมีสมาชิกเท่ากับ $2^3 = 8$ โครงสร้างข้อมูลเอ็กเทนดิเบิล แฮชชิงใหม่จึงกลายเป็น



ภาพ 12 แสดงการแบ่งบล็อค

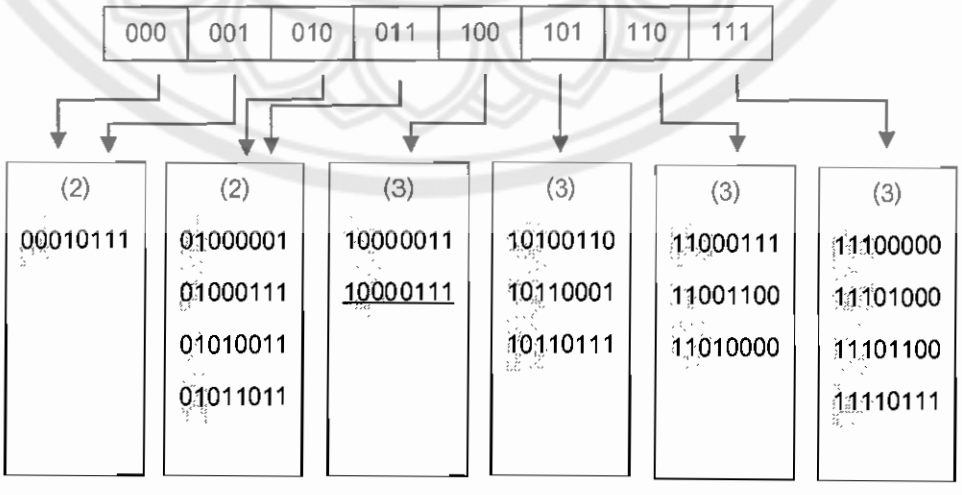
สามารถเพิ่มข้อมูล 111011000 ลงไปได้ โครงสร้างใหม่ที่แตกต่างออกไปคือ ไดเร็คทอรีมีขนาดใหญ่ขึ้น มี 8 ค่ามาจาก $2^D = 2^3 = 8$ ส่วนค่า D^L ในบางบล็อคจะกลายเป็น 3 ตัวชี้เชื่อม จะชี้ไปยังบล็อคที่มีค่าเหมือนตน จากนั้นเพิ่มข้อมูลค่าถัดไปลงในเอ็กเทนดิเบิล แฮชชิง



ภาพ 13 แสดงการเพิ่มข้อมูล กรณีทำให้เกิดการแบ่งบล็อค

สามารถเพิ่มข้อมูล 11100000, 11001100 ลงไปในโครงสร้างข้อมูลเอ็กเทนดิเบิล แฮชชิงได้ แต่เมื่อจะเพิ่มข้อมูล 10000111 ลงในบล็อคที่สามบล็อคเต็มและต้องแบ่งบล็อคออก

4. โครงสร้างข้อมูลเอ็กเทนดิเบิล แฮชชิง จึงกลายเป็น



ภาพ 14 แสดงการเพิ่มข้อมูลลงไปใน Extendible Hashing ได้ครบทุกค่า

5. ในการค้นหาข้อมูล (Searching) ถ้าต้องการจะค้นหาข้อมูล 10110111 วิธีการคือ

5.1 จะทำงานไดเร็คทอรีตรงดรรชนี 101 (เนื่องจาก D = 3 จึงยึด 3 บิตแรก)

5.2 ทำงานครั้งที่สองตรงบล็อคที่ดรรชนี 101 เชื่อมมา

5.3 ค้นหาข้อมูลในบล็อคนั้น ก็จะทราบคำตอบว่ามีข้อมูลที่ต้องการเก็บอยู่ในเอ็กเทนดิเบิล แฮชชิงนี้หรือไม่

6. ในการลบข้อมูล (Delete) ถ้าต้องการลบข้อมูลค่า 10110111 มีวิธีการดังนี้ คือ

6.1 ทำตามขั้นตอนที่ 5.1 - 5.3 เพื่อค้นหาตำแหน่งข้อมูลตัวที่ต้องการลบ

6.2 ถ้าพบก็จะลบค่านั้นทิ้ง

6.3 ถ้าไม่พบก็จะรายงานผลว่า ไม่มีข้อมูลที่ต้องการลบในเอ็กเทนดิเบิล แฮชชิงนี้

โครงสร้างข้อมูลเอ็กเทนดิเบิล แฮชชิง สามารถเพิ่มข้อมูลได้ไปอีกเรื่อย ๆ เนื่องจากบล็อคยังคงแบ่งออกมาได้อีก การที่จะทราบได้ว่าโครงสร้างข้อมูลเอ็กเทนดิเบิล แฮชชิงนั้นเต็มหรือยัง สามารถทราบได้จากขนาดของไดเร็คทอรี นั่นคือ ค่า D จะมีค่าสูงสุดเท่ากับผลต่างของจำนวนบิตของข้อมูล ($-\sqrt{M}$) ถ้าค่า D มีค่าสูงสุดและทุกบล็อคมีข้อมูลเก็บอยู่เต็ม นั่นหมายความว่าโครงสร้างข้อมูลเอ็กเทนดิเบิล แฮชชิงนั้น ๆ เต็มแล้ว จากตัวอย่างภาพ 14 ค่า M มีค่าเท่ากับ 4 และข้อมูลมีขนาด 8 บิต ดังนั้นค่า D มีขนาดสูงสุดที่ $(8-\sqrt{4}) = 6$ บิต ทำให้ไดเร็คทอรีมีดรรชนีมากถึง $2^6 = 64$ ดรรชนี และแต่ละดรรชนีชี้ไปยังบล็อคที่สามารถเก็บข้อมูลได้สูงสุด 4 ข้อมูลต่อบล็อค (ตามค่า M) ดังนั้นข้อมูลทั้งหมดที่สามารถเก็บในเอ็กเทนดิเบิล แฮชชิงนี้ได้ คือ 64 x 4 = 256 จะเห็นได้ว่า ถ้าค่า M และ D มีค่ามากขึ้น จำนวนข้อมูลที่เก็บได้ก็จะมากขึ้นไปอีก ดังนั้นจะเห็นได้ชัดเจนว่าโครงสร้างข้อมูลเอ็กเทนดิเบิล แฮชชิง จะมีประสิทธิภาพสูงมากๆ เมื่อใช้เก็บข้อมูลที่มีปริมาณมาก

การแก้ปัญหาการชนกันของคีย์ (Collision)

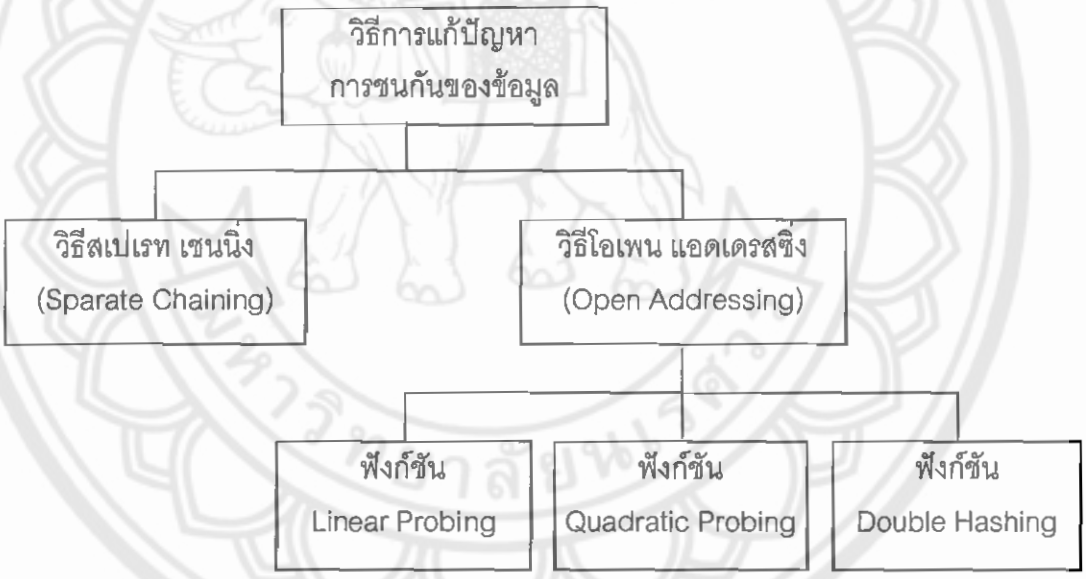
เมื่อเลือกแฮชฟังก์ชันที่จะใช้กับโครงสร้างข้อมูลแฮชชิง สิ่งที่จำเป็นต้องคำนึงถึงอีกอย่างก็คือ การหามาตรการในการแก้ไขปัญหาการชนกันของคีย์ กล่าวคือ เมื่อคีย์ข้อมูลที่ผ่านแฮชฟังก์ชันออกมาแล้วได้ค่าตำแหน่งที่อยู่ตรงกับตำแหน่งที่มีข้อมูลอื่นเก็บอยู่ก่อนแล้ว (เกิดการชนกัน) จึงจำเป็นต้องหาที่อยู่ใหม่ให้กับคีย์ข้อมูลที่เป็นฝ่ายมาชน

ขนิษฐา นามี (2548) กล่าวว่า สิ่งที่จะบอกเราว่าตารางแฮชที่สร้างไว้แล้วนั้น ถ้าต้องการจะใส่ข้อมูลเพิ่มเข้าไป จะมีโอกาสเกิดการชนกันมากน้อยเพียงใด หรือในขณะนี้

ตารางแฮชมีข้อมูลเก็บอยู่มากน้อยเพียงใด จะทราบได้จากค่า ๆ หนึ่ง ค่านั้นคือโหลดแฟ็คเตอร์ (Load Factor) ซึ่งเป็นค่าเปอร์เซ็นต์บอกปริมาณความหนาแน่นของข้อมูลในตาราง สัญลักษณ์ที่ใช้แทนความหมายว่าเป็นค่าโหลดแฟ็คเตอร์ก็คือค่า λ สามารถคำนวณหาค่าได้ตามสมการ

$$\lambda = \frac{\text{จำนวนข้อมูลที่มีข้อมูลเก็บอยู่}}{\text{จำนวนช่องข้อมูลทั้งหมด}} \times 100$$

ดังนั้น การเปรียบเทียบประสิทธิภาพวิธีการแก้ปัญหาการชนกันของคีย์สามารถเปรียบเทียบจากค่าโหลดแฟ็คเตอร์ได้เช่นกัน วิธีการแก้ปัญหาการชนกันของคีย์มีด้วยกันหลายวิธี ดังภาพ 15 สามารถแบ่งได้เป็น 2 วิธีหลัก ๆ คือ วิธีสเปเรท เชนนิ่ง (Sparate Chaining) และ วิธีโอเพน แอดเดรสซิ่ง (Open Addressing)



ภาพ 15 วิธีการแก้ปัญหาการชนกันของคีย์ที่ได้ค่าแอดเดรสเดียวกัน

วิวัฒน์ อภิสิทธิ์ภิญโญและอมร มุสิกสาร (2548) กล่าวว่า ในการแปลงค่าคีย์ด้วยฟังก์ชัน หรือใช้วิธีต่าง ๆ ในบางกรณีอาจพบปัญหาเกี่ยวกับการชนกันของคีย์ คือ ได้ค่าแอสเดรสเดียวกัน จึงแบ่งวิธีการแก้ปัญหาการชนกันของคีย์ ออกเป็น 3 วิธีคือ

1.การแก้ปัญหาการชนกันของคีย์ด้วยการปรับตำแหน่ง

2.การแก้ปัญหาการชนกันของคีย์ด้วยลิงค์ลิสต์

3.การแก้ปัญหาการชนกันของคีย์ด้วยบักเก็ต

1. การแก้ปัญหาการชนกันของคีย์ด้วยการปรับตำแหน่ง

การแก้ปัญหาการชนกันของคีย์ด้วยวิธีการนี้คือ

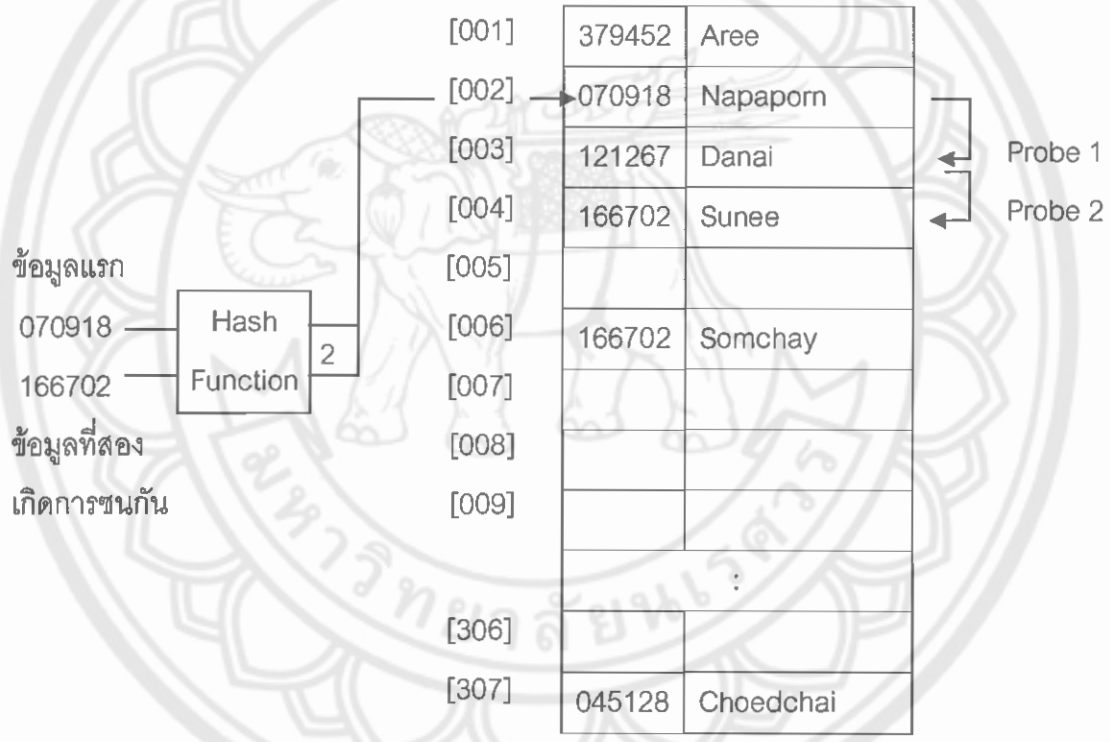
1.1 วิธีลิเนียร์โพรบ (Linear Probe)

1.2 วิธีควอร์ตดราติคโพรบ (Quadratic Probe)

1.3 วิธีซูโดแรนดอม (Pseudo Random)

1.4 วิธีคีย์ออฟเซต (Key Offset)

วิธีลิเนียร์โพรบ จะใช้วิธีการเลื่อนตำแหน่งของคีย์ที่มีการชนกันย้ายลงไปยังแอดเดรสที่ยังว่างอยู่ดังภาพ 16



ภาพ 16 การแก้การชนกันของคีย์ด้วยวิธี Linear Probe

จากภาพ 16 เมื่อมีการแปลงค่าคีย์ 070918 จะได้แอดเดรสที่ 002 ซึ่งสามารถลงข้อมูลได้โดยไม่มีการชนกันของคีย์ เพราะยังไม่มีข้อมูลแต่เมื่อมีคีย์ 166702 รับเข้ามาแปลงค่าคีย์ได้แอดเดรส 002 เช่นกัน แต่ในแอดเดรสนี้มีข้อมูลอยู่แล้ว จึงต้องเลื่อนมาตรวจสอบหาแอดเดรสที่ว่างครั้งที่ 1 แต่ตำแหน่งที่ 003 มีค่าข้อมูลของคีย์ 121267 อยู่แล้ว จึงต้องตรวจสอบหาแอดเดรสที่ว่าง ครั้งที่ 2 จึงได้แอดเดรส 004 เพื่อใช้ในการเก็บข้อมูล

ข้อเสียของวิธีการนี้ คือ การจัดวางข้อมูลลงยังแอดเดรสนั้น ไม่สามารถที่จะกำหนดตำแหน่งที่แน่นอนได้เพราะต้องอาศัยการตรวจสอบแอดเดรสที่ใกล้ตำแหน่งเดิมก่อน แล้วจึงเลื่อนไปยังตำแหน่งต่อไปจนกว่าจะพบแอดเดรสว่างสำหรับจัดเก็บข้อมูล ทำให้การค้นหาข้อมูลจะต้องตรวจสอบและเทียบค่ากับตำแหน่งหลายครั้งจนกว่าจะพบ

วิธีควอร์ตดราติคโพรบ วิธีการนี้จะทำการคำนวณตำแหน่งใหม่ให้กับคีย์ที่เกิดการชนกันจากสูตร

$$\text{New Address} = (K \pm j^2) \text{ MOD } N$$

K : ตำแหน่งแอดเดรสที่เกิดการชนกันของคีย์

J : ลำดับเลขของการชนกันโดยเริ่มนับตั้งแต่ 1

N : จำนวนตารางแฮชที่ใช้ในการเก็บข้อมูล

โดยวิธีการนี้จะใช้การตรวจสอบจำนวนของการชนกันของคีย์ก่อนแล้วนำมายกกำลังสอง จากนั้นบวกลบกับตำแหน่งแอดเดรสที่มีการชนกัน หารด้วยจำนวนช่องของตาราง ดังตาราง 6

ตาราง 6 การแก้การชนกันของคีย์ด้วยวิธี Quadratic Probe (N = 100)

ลำดับการชนกัน (j)	ตำแหน่งแอดเดรสที่ชนกัน (K)	j^2	ตำแหน่งแอสเดรสใหม่
1	1	1	02
2	2	4	06
3	6	9	15
4	15	16	31
5	31	25	56
6	56	36	92
7	92	49	41

ป RC 91 ส944ก 1550
21 พ.ย. 2550
i. 3871748
สำนักหอสมุด

วิธีซูโดแรนดอม สำหรับวิธีการนี้อาจเรียกได้ว่าเป็นการแก้ปัญหาการชนกันของคีย์ด้วยการทำแฮชซ้ำ 2 ครั้ง (Double Hashing) โดยเมื่อคำนวณหาตำแหน่งในครั้งแรกแล้วเกิดการชนกันก็จะคำนวณหาตำแหน่งใหม่จากสูตร

New Address = (Ax+c) MOD N+1

A : ตำแหน่งแอดเดรสที่มีการชนกัน

x : จำนวนครั้งของการทำซ้ำ

c : ค่าคงที่กำหนดขึ้นซึ่งต้องไม่ใช่ค่าของ N

N : จำนวนตารางแฮช

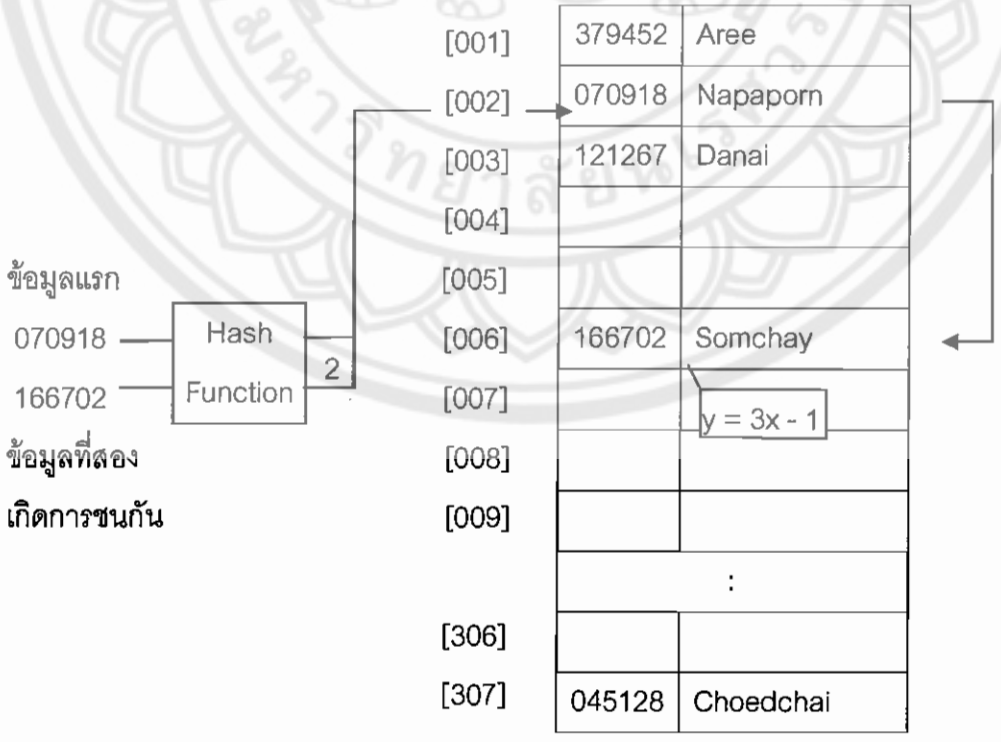
ตัวอย่าง การแก้การชนกันของแอดเดรส ในตำแหน่งที่ 3 และเป็นครั้งที่ 2 ของการทำซ้ำกันของตำแหน่ง กำหนดค่า c = -1 และกำหนดค่า N = 307

New Address = (Ax+c) MOD N+1

= (3 x 2 + (-1)) MOD 307 + 1

= 6

จะได้ตำแหน่งใหม่ของการจัดเรียงคือ 006 ดังภาพ 17



ภาพ 17 การแก้การชนกันของคีย์ด้วยวิธี Pseudo Random

วิธีคีย์ออฟเซต สำหรับการแก้ปัญหาการชนกันของคีย์ด้วยวิธีการนี้ จะนำค่าคีย์ที่มีการชนกันในแอดเดรสมาทำการออฟเซต (Offset) ก่อน คือ แปลงค่าด้วยการหารด้วยจำนวนช่องตารางแฮช จากนั้นจึงนำคีย์ที่ได้ไปหาแอดเดรส ซึ่งถ้ามีการชนกันอีกก็จะวนรอบการทำงานอีก

Offset = Key/N

New Address = ((Offset + Old Address) MOD N) + 1

ตัวอย่างจากภาพ 17 ค่าคีย์ 166702 มีการชนกันของคีย์ในแอดเดรส 002 ในการหาค่าของแอดเดรสด้วยวิธีการคีย์ออฟเซตในครั้งที่ 1 และครั้งที่ 2 เมื่อตารางแฮชเก็บข้อมูลได้ 307 ช่อง สามารถหาค่าได้ดังต่อไปนี้

หาแอดเดรส ครั้งที่ 1

Offset = Key/N

= 166702/307

= 543

New Address = ((543 + 002) MOD 307) + 1

= 239

หาแอดเดรส ครั้งที่ 2

Offset = Key/N

= 166702/307

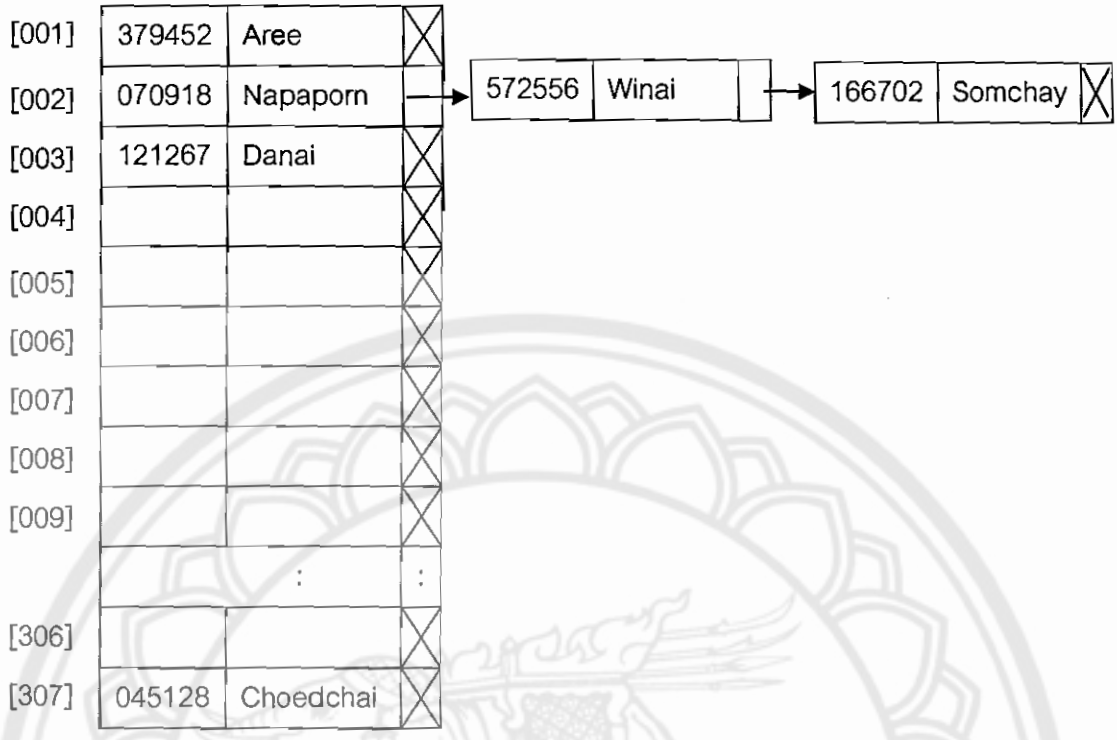
= 543

New Address = ((543 + 239) MOD 307) + 1

= 169

2. การแก้ปัญหาการชนกันของคีย์ด้วยลิงค์ลิสต์ (Linked List)

สำหรับวิธีการแก้ปัญหาในรูปแบบนี้จะนำเอาโครงสร้างแบบลิงค์ลิสต์มาใช้ในการเชื่อมโยงรายการโดยเมื่อตารางแฮชจัดเก็บข้อมูลที่มีการชนกันของคีย์ จะทำการเพิ่มเรคอร์ดในการจัดเก็บคีย์ โดยใช้วิธีการกำหนดพอยน์เตอร์ในการลิงค์ไปยังข้อมูลที่เกิดการชนกันของคีย์ หรือค่าคีย์ที่มีแอดเดรสตรงกันนั่นเอง ดังภาพ 18

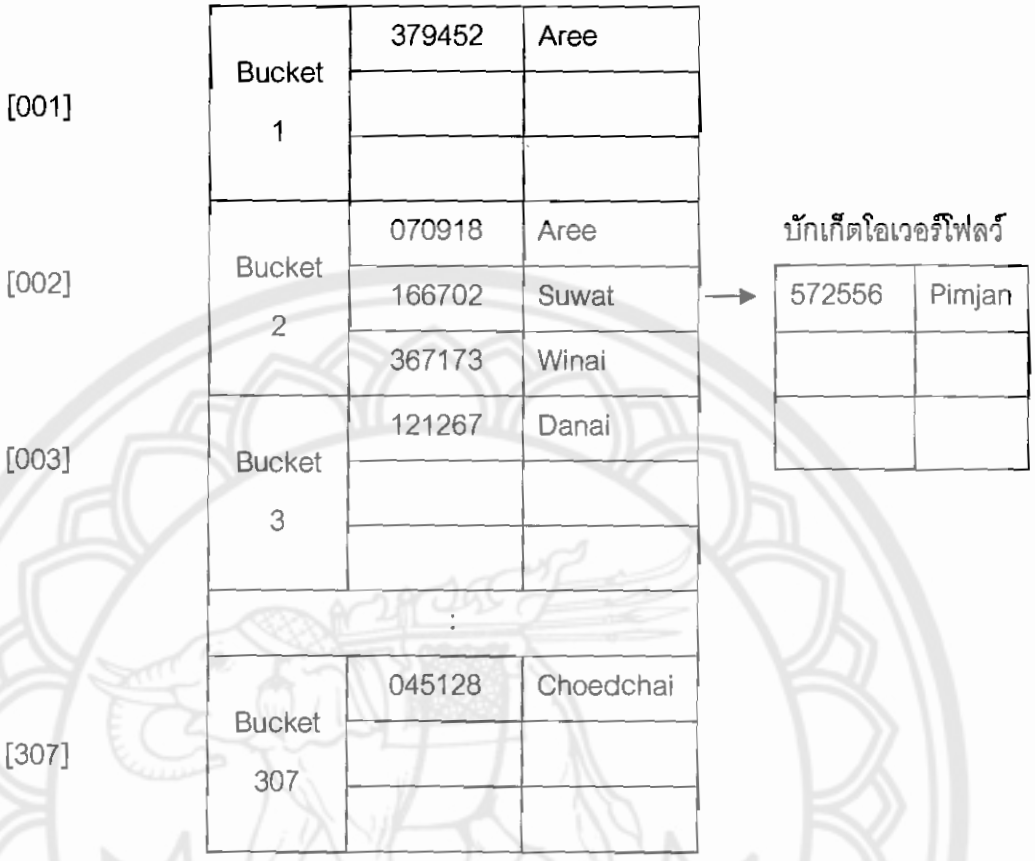


ภาพ 18 การแก้การชนกันของคีย์ด้วยลิงค์ลิสต์

จากภาพ 18 ในแอดเดรส 002 จัดเก็บข้อมูล Napaporn ไว้แล้วเมื่อคีย์ต่อไป คือ คีย์ 572556 เป็นข้อมูลของ Winai และคีย์ 166702 ซึ่งเป็นข้อมูลของ Somchay ได้ค่าแอดเดรส 002 เช่นกัน ก็ใช้การเชื่อมโยงด้วยพอยน์เตอร์ และเมื่อมีคีย์ต่อไปที่มีแอดเดรสตรงกัน ก็จะเพิ่มข้อมูลในโครงสร้างลิงค์ลิสต์ต่อไปอีก

3. การแก้ปัญหาการชนกันของคีย์ด้วยบักเก็ต (Bucket)

วิธีการนี้เหมาะสมกับข้อมูลที่คาดว่าจะเกิดการชนกันของคีย์ที่มีปริมาณพอ ๆ กัน จึงมีการเตรียมบักเก็ตเอาไว้ เพื่อใช้ใส่ข้อมูลในกลุ่มเดียวกัน โดยบรรจุไว้ในอะเรย์ ซึ่งไม่จำเป็นต้องเชื่อมต่อกันด้วยพอยเตอร์ และหากมีบางบักเก็ตที่เกิดการชนกันมากเกินกว่าจะรับได้ก็ยังสามารถ**ใช้วิธีการของลิงค์ลิสต์เพื่อเชื่อมต่อบักเก็ตใหม่ได้อีก เรียกบักเก็ตนี้ว่า บักเก็ตโอเวอร์โฟลว์**



ภาพ 19 การแก้การชนกันของคีย์ด้วย บักเก็ต แฮชชิง (Bucket Hashing)

จากภาพ 19 ในแต่ละบักเก็ตของคีย์จะถูกแบ่งออกเป็น 3 ช่อง และเก็บค่าคีย์ในช่องแรก แต่ในกรณีที่มีการซ้ำกันของคีย์มากกว่าจำนวนช่อง เช่น แอดเดรส 002 ค่าคีย์ 572556 จะต้องถูกจัดอยู่ในแอดเดรส 002 แต่บักเก็ตเต็มจึงนำลงมาเก็บในบักเก็ตที่โอเวอร์โฟลว์แทน

ฐานข้อมูล (Database)

ฐานข้อมูล (Database) คือ ชุดของสารสนเทศที่มีโครงสร้างที่มีแบบแผน ชุดของสารสนเทศใด ๆ ก็อาจเรียกว่าเป็นฐานข้อมูลได้ คำว่าฐานข้อมูลนี้มักใช้อ้างถึงข้อมูลที่ประมวลผลด้วยคอมพิวเตอร์ แต่บางครั้งคำนี้ก็ถูกใช้เพื่ออ้างถึงข้อมูลที่ยังมิได้ประมวลผลด้วยคอมพิวเตอร์เช่นกัน ในแง่ของการวางแผนให้ข้อมูลดังกล่าวสามารถประมวลผลด้วยคอมพิวเตอร์ได้

ฐานข้อมูลถูกพัฒนาเป็นครั้งแรกในทศวรรษ 1960 ซึ่งผู้บุกเบิกในสาขานี้คือ ชาลส์ บากแมน แบบจำลองข้อมูลสำคัญสองแบบเกิดขึ้นในช่วงเวลานี้ ซึ่งเริ่มต้นด้วยแบบจำลอง

ข่ายงาน (Network Model) และตามด้วยแบบจำลองเชิงลำดับชั้น (Hierarchical Model) แบบจำลองทั้งสองแบบนี้ในภายหลังถูกนำไปใช้น้อยลงเพราะแบบจำลองเชิงสัมพันธ์ (Relational Model) ได้ถูกพัฒนาขึ้นและเป็นที่นิยมมากกว่าแบบจำลองเชิงสัมพันธ์ มีพื้นฐานมาจากทฤษฎีเซต (Set Theory) ได้มีการเสนอแบบจำลองดัดแปลงซึ่งใช้ทฤษฎีเซตคลุมเครือ (Fuzzy Set) ขึ้นเป็นอีกทางเลือกหนึ่ง (วิกิพีเดียสารานุกรมเสรี, 2549. เว็บไซต์) ปัจจุบันได้มีการนำความคิดด้านเชิงวัตถุมาใช้สร้างแบบจำลองฐานข้อมูล ที่กำลังเข้าสู่ความนิยมนั่นคือ ฐานข้อมูลเชิงวัตถุ (OODB: Object Oriented Database)

ระบบจัดการฐานข้อมูล (Database Management Systems)

ซอฟต์แวร์สำหรับจัดการฐานข้อมูล (DBMS - Database Management System) มีทั้งแบบสำหรับฐานข้อมูลขนาดเล็กที่มีผู้ใช้คนเดียวและฐานข้อมูลขนาดใหญ่ที่มีผู้ใช้จำนวนมาก ปกติจะประกอบด้วยโปรแกรมหลายโปรแกรมด้วยกัน และโดยทั่วไปส่วนใหญ่จะใช้สถาปัตยกรรมแบบให้บริการ (Client/Server)

โปรแกรมส่วนหน้า (Front-end) ของซอฟต์แวร์สำหรับจัดการฐานข้อมูล (ได้แก่ โปรแกรมรับบริการ) จะเกี่ยวข้องเฉพาะการนำเข้าข้อมูล การตรวจสอบ และการรายงานผลเป็นสำคัญ ในขณะที่โปรแกรมส่วนหลัง (Back-end) ซึ่งได้แก่ โปรแกรมให้บริการ จะเป็นชุดของโปรแกรมที่ดำเนินการเกี่ยวกับการควบคุม การเก็บข้อมูล และการตอบสนองการร้องขอจากโปรแกรมส่วนหน้า โดยปกติแล้วการค้นหาและการเรียงลำดับ จะดำเนินการโดยโปรแกรมให้บริการ รูปแบบของระบบฐานข้อมูล มีหลากหลายรูปแบบด้วยกัน นับตั้งแต่การใช้ตารางอย่างง่ายที่เก็บในแฟ้มข้อมูลแฟ้มเดียว ไปจนกระทั่งฐานข้อมูลขนาดใหญ่มาก ที่มีระเบียนหลายล้านระเบียน ซึ่งเก็บในห้องที่เต็มไปด้วยดิสก์ไดรฟ์ (Disk Drive) หรืออุปกรณ์หน่วยเก็บข้อมูลอิเล็กทรอนิกส์รอบข้าง (Peripheral) อื่น ๆ

ภาษาสอบถามเชิงโครงสร้าง (Structured Query Language :SQL)

ภาษาสอบถามเชิงโครงสร้าง เป็นภาษาทางด้านฐานข้อมูลที่ถูกพัฒนาขึ้นจากแนวคิดของ Relational Calculus **และ** Relational Algebra **เป็นหลัก (กิตติ ภักดีวัฒนะกุล และ จำลอง ครูอุตสาหะ,2542) และเป็นภาษาที่ได้รับความนิยมอย่างมากเนื่องจาก**

1. เป็นภาษาซึ่งใกล้เคียงกับภาษาพูด ทำให้ง่ายต่อการทำความเข้าใจ
2. สามารถทำงานร่วมกับภาษาอื่นได้ โดยฝังรวมไว้กับภาษาอื่น ๆ เพื่อทำงานร่วมกันได้
3. เป็นภาษามาตรฐาน ซึ่งมีรูปแบบในการเขียนคำสั่งคล้าย ๆ กันไม่ว่าจะใช้บนโปรแกรมใดหรือ DBMS ใด

ภาษาสอบถามเชิงโครงสร้างสามารถแบ่งได้ 3 ส่วน (ศิริลักษณ์ โรจนกิจอำนวย,2542) ได้แก่

1. ส่วนภาษานิยามข้อมูล (Data Definition Language : DDL) เป็นส่วนของคำสั่งที่ใช้กำหนดโครงสร้างข้อมูล สร้างฐานข้อมูล สร้างตาราง กำหนดดัชนี สร้างวิวเพื่อดูผลลัพธ์ และใช้ในการกำหนดกฎเกณฑ์ต่าง ๆ สำหรับให้ DBMS ตรวจสอบความถูกต้องของข้อมูล
2. ส่วนภาษาจัดการข้อมูล (Data Manipulation Language : DML) เป็นส่วนของคำสั่งในการค้นหาข้อมูล เพิ่มเติมข้อมูล เปลี่ยนแปลงแก้ไขข้อมูล และการลบข้อมูล
3. ส่วนภาษาในการควบคุมข้อมูล (Data Control Language : DCL) เป็นส่วนของคำสั่งที่ใช้ควบคุม และการกำหนดสิทธิ์ของผู้ใช้ในการจัดการฐานข้อมูล เพื่อความปลอดภัยสำหรับฐานข้อมูล

หลักเบื้องต้นในการวินิจฉัยและการรักษาโรค

ประวัติผู้ป่วย

การถามประวัติผู้ป่วย แพทย์จำเป็นต้องทราบข้อมูลทั่ว ๆ ไปเกี่ยวกับผู้ป่วย เช่น ชื่อ เพศ อายุ เชื้อชาติ ที่อยู่ อาชีพ และสถานที่เกิด จากนั้นจึงซักถามอาการและประวัติการป่วย

อาการสำคัญ คือ อาการที่นำผู้ป่วยมาหาแพทย์หรือมาโรงพยาบาลเพื่อรับการรักษา รวมทั้งระยะเวลาที่มีอาการนั้น

ประวัติปัจจุบัน คือ รายละเอียดของเรื่องราวการเจ็บป่วยหรืออาการที่ผู้ป่วยมาหาแพทย์ในครั้งนี้ประวัติปัจจุบันอาจมีระยะเวลาสั้นหรือยาวก็ได้ขึ้นอยู่กับสภาพที่ผู้ป่วยมีอาการ ผู้ป่วยที่เป็นโรคไส้ติ่งอักเสบอาจมีประวัติเป็นมาเพียงไม่กี่ชั่วโมง ในขณะที่ผู้ป่วยเป็นโรคหลอดลม

อักเสบเรื้อรังอาจจะมีประวัติมานานเป็นปี อาการที่เกิดขึ้นที่เกี่ยวข้องกับประวัติปัจจุบันอาจจะมีมากกว่าหนึ่งอาการ และโดยมากแพทย์มักจะพยายามถามประวัติอาการดังกล่าวนั้นโดยละเอียด โดยเฉพาะระยะเวลาของแต่ละอาการที่เกิดขึ้น ลักษณะและวิธีการที่เกิดขึ้น ความถี่และความรุนแรง และการเปลี่ยนแปลงของอาการ ตลอดจนความเกี่ยวข้องกับอาการอื่น ๆ รวมทั้งการรักษาที่เกี่ยวข้องกับอาการดังกล่าวซึ่งผู้ป่วยได้รับมาก่อนนั้นด้วย นอกจากนี้ แพทย์ยังมักจะถามประวัติตามระบบซึ่งอาจจะมีความสำคัญเกี่ยวข้องกับการวินิจฉัยโรคได้ การถามประวัติตามระบบนี้แพทย์มักจะถามอาการสำคัญและเกี่ยวข้องกับระบบต่าง ๆ ของร่างกายและจิตใจ

ประวัติอดีต หมายถึง เรื่องราวการเจ็บป่วยของผู้ป่วยที่เคยเป็นมาแล้ว และไม่เกี่ยวข้องกับการเจ็บป่วยครั้งนี้ ผู้ป่วยอาจมาพบแพทย์ด้วยอาการของแผลในกระเพาะอาหาร แต่อาจจะมีประวัติในอดีตว่าเคยเป็นวัณโรคปอดเมื่อ 15 ปีก่อน และเคยผ่าตัดต่อมทอนซิลเมื่อ 10 ปีมาแล้ว เป็นต้น

ประวัติครอบครัว หมายถึง ประวัติการเจ็บป่วยของบุคคลในครอบครัวผู้ป่วย เช่น การเจ็บป่วยของบิดา มารดา บุตร และปู่ย่าตายาย เป็นต้น โดยเฉพาะประวัติของโรคที่อาจมีความเกี่ยวเนื่องกันกับการเจ็บป่วยที่ผู้ป่วยเป็นในครั้งนี้ นอกจากนี้ แพทย์ยังมักจะถามอายุ สุขภาพ สาเหตุของการถึงแก่กรรม และอายุที่ถึงแก่กรรมของสมาชิกในครอบครัว ตลอดจนอาจจะถามประวัติที่เกี่ยวข้องกับโรคที่พบได้บ่อย ๆ เช่น เบาหวาน ความดันเลือดสูง วัณโรคปอด มะเร็ง โรคหัวใจ โรคหืด และโรคแพ้ต่าง ๆ

ประวัติส่วนตัว คือ เรื่องราวส่วนตัวของผู้ป่วย เช่น คู่สมรส บุตร การทำงาน การเจริญเติบโตในวัยเด็ก และสิ่งแวดล้อม นอกจากนี้ ประวัติส่วนตัวมักรวมถึงนิสัยประจำของผู้ป่วย เช่น การดื่มสุรา การสูบบุหรี่ ยาที่ใช้เป็นประจำ ตลอดจนความสัมพันธ์กับสังคมต่าง ๆ การศึกษา รายได้และฐานะหรือปัญหาทางเศรษฐกิจ ตลอดจนอาชีพ สิ่งแวดล้อมอื่น ๆ และสถานที่ซึ่งผู้ป่วยอาศัยอยู่ หรือได้เคยอาศัย หรือได้เคยไปมาแล้ว ซึ่งอาจเกี่ยวข้องกับการเจ็บป่วยในครั้งนี้

การตรวจร่างกาย

อาการแสดงแห่งชีวิต ได้แก่ การตรวจอุณหภูมิ ชีพจร การหายใจ และความดันเลือดของผู้ป่วย นอกจากนี้ยังอาจจะรวมถึงการตรวจวัดน้ำหนักตัวและส่วนสูงด้วย

ลักษณะทั่วไป ได้แก่ การตรวจดูลักษณะโดยทั่วไป ตลอดจนกระทั่งท่าทางการเดิน ท่าทีของผู้ป่วยในขณะกำลังตรวจ ดูลักษณะว่าผอม อ้วน เตี้ยหรือสูงกว่าธรรมดา มีอาการหอบ

บวม เขียวคล้ำ และซีดหรือไม่ การสังเกตความพิการที่อาจเห็นได้ง่าย รวมทั้งความรู้สึกทางอารมณ์ของผู้ป่วยที่มีต่อการเจ็บป่วยที่เป็นอยู่

การตรวจตามระบบ ได้แก่ การตรวจตามระบบต่าง ๆ ของร่างกายให้ครบถ้วน โดยเฉพาะอย่างยิ่ง ระบบซึ่งอาจจะเกี่ยวข้องกับการเจ็บป่วยของผู้ป่วยที่กำลังเป็นอยู่

การสืบค้น

การสืบค้นประจำ ส่วนใหญ่เป็นการตรวจทางห้องปฏิบัติการง่าย ๆ หรือการตรวจที่อาจกระทำได้ข้างเตียงผู้ป่วย เช่น การตรวจเลือดเพื่อหาเฮโมโกลบินจำนวนเม็ดเลือดแดง จำนวนและชนิดของเม็ดเลือดขาว การตรวจปัสสาวะและอุจจาระ เป็นต้น

การสืบค้นพิเศษ คือ การสืบค้นอื่น ๆ ซึ่งจะช่วยสนับสนุนในการวินิจฉัยโรค หรือช่วยในการแยกโรค การสืบค้นพิเศษนี้อาจเป็นการตรวจในห้องทดลองหรือไม่ก็ได้ เช่น การตรวจทางรังสีวิทยา การวิเคราะห์หาปริมาณหรือระดับของสารเคมีต่าง ๆ ในเลือด การตรวจโดยใช้สารทึบรังสี การตัดเนื้อ (Biopsy) เพื่อนำไปตรวจด้วยกล้องจุลทรรศน์ การฉีดสารทึบรังสีเข้าหลอดเลือด (Angiography) การใช้กล้องส่องดูหลอดลม (Bronchoscopy) การใช้กล้องส่องดูกระเพาะอาหาร (Gastrosputed Tomography) เป็นต้น แพทย์มักสั่งหรือกระทำการสืบค้นพิเศษเมื่อมีข้อชี้บ่งที่แน่นอนและการสืบค้นพิเศษดังกล่าวนั้นไม่เป็นอันตราย ไม่เป็นโทษ หรือก่อให้เกิดการเจ็บปวดแก่ผู้ป่วยเกินความจำเป็น ตลอดจนต้องคำนึงถึงค่าใช้จ่ายที่จำเป็นต้องใช้เมื่อเปรียบเทียบกับผลลัพธ์ที่อาจได้จากการสืบค้น นั้น ๆ ด้วย

การรักษา

การรักษาเฉพาะ คือ การรักษาตามสาเหตุของโรคโดยการใช้ยาหรือโดยวิธีการอื่น ๆ ซึ่งมีผลเป็นการบำบัดสาเหตุและทำให้โรคนั้นหายไป เช่น การรักษาโรคไทฟอยด์ ปอดบวม โดยการให้ยาเฉพาะตามสาเหตุนั้น ๆ

การรักษาตามอาการ คือ การรักษาตามอาการของผู้ป่วย เช่น ปวดศีรษะ ก็ให้ยากินแก้ปวด ไอก็ให้ยาระงับอาการไอ การรักษาตามอาการมีจุดประสงค์เพื่อให้อาการนั้นหายไปโดยมิได้รักษาสาเหตุ ดังนั้นผู้ป่วยอาจกลับมีอาการดังเดิมถ้ามิได้ให้ยาระงับอาการนั้น ๆ ไว้ หรือยาที่ให้นั้นหมดฤทธิ์ โดยปกติแพทย์จะให้การรักษาตามอาการเมื่อจำเป็น ในขณะที่ยังไม่ทราบสาเหตุหรือกำลังหาสาเหตุอยู่ โดยหวังว่าเมื่อพบสาเหตุแล้วจะได้ให้การรักษาเฉพาะได้ การรักษาตามอาการอาจมีผลเสียได้ เช่น ผลเสียที่เกิดจากการให้ยาระงับอาการนั้นเอง โดยยาระงับอาการอาจกลบอาการทำให้แพทย์ไม่สามารถวินิจฉัยโรคได้ หรือวินิจฉัยโรคได้ช้าหรือลำบากขึ้น

การรักษาประคับประคอง คือ การรักษาเพื่อประคับประคอง หรือพยุงอาการทั่ว ๆ ไปของผู้ป่วยให้ดีขึ้น เป็นการเพิ่มความต้านทานตามธรรมชาติให้แก่ร่างกายผู้ป่วยอีกวิธีหนึ่ง ตัวอย่างการรักษาประเภทนี้ ได้แก่ การแนะนำให้พักผ่อน การให้น้ำเกลือหรือกลูโคสในผู้ป่วยที่รับประทานอาหารได้น้อยเพราะเหตุใดก็ตาม เช่น เบื่ออาหาร คลื่นไส้ เป็นต้น การรักษาโดยการให้วิตามินและแร่ธาตุจำเป็นแก่ร่างกาย การพยาบาลที่ดีและกายภาพหรือสรีระบำบัด (Physiotherapy) เป็นต้น

การรักษาบรรเทาอาการ เป็นการรักษาเพื่อบรรเทาอาการผู้ป่วยมิให้เลวลง หรือให้เลวลงช้ากว่าปกติ การรักษาประเภทนี้ใช้ในผู้ป่วยที่เป็นโรคเรื้อรังหรือโรคที่ทราบแน่นอนว่ารักษาไม่หาย เช่น การให้ยาต้านมะเร็งหรือการใช้รังสีบำบัด (Radiotherapy) ในผู้ป่วยที่เป็นโรคมะเร็ง

จิตวิทยารักษา เป็นการรักษาที่มีจุดประสงค์สำคัญเพื่อทำให้ผู้ป่วยมีจิตใจและอารมณ์ดีขึ้น ผู้ป่วยที่มีโรคของร่างกายจะมีความผิดปกติทางจิตใจด้วยเสมอและเป็นหน้าที่ของแพทย์ที่ต้องรักษาผู้ป่วยทั้งร่างกายและจิตใจ สุภาษิตโบราณอันหนึ่งกับกล่าวว่า "ไม่มีโรค มีแต่ผู้ป่วย" ซึ่งมีความมุ่งหมายที่จะให้รักษาผู้ป่วยทั้งร่างกายและจิตใจ มิใช่แต่รักษาโรคทางกายเท่านั้น วิธีหนึ่งของจิตวิทยารักษา คือ การที่แพทย์ให้ความเชื่อมั่น อธิบายธรรมชาติของโรคให้ผู้ป่วยฟังโดยใช้คำพูดง่าย ๆ เพื่อเป็นการระงับอาการวิตกกังวลจากโรคทางกายที่ผู้ป่วยเป็นอยู่ (สารานุกรมไทยสำหรับเยาวชนฯ เล่ม 10, www.Thailand Junior Encyclopedia Project.htm)

ระบบสนับสนุนการตัดสินใจ

ปัจจุบันความก้าวหน้าของเทคโนโลยีสื่อสารและโทรคมนาคม ทำให้ข้อมูลข่าวสารสามารถเดินทางได้อย่างคล่องตัวและเป็นอิสระมากขึ้น ส่งผลให้องค์กรต่าง ๆ สามารถรับส่งข้อมูลข่าวสาร และข้อสนเทศได้ในระยะเวลาที่รวดเร็ว โดยข้อมูลมีความชัดเจน ถูกต้อง และสะดวกขึ้น จึงนับว่ามีความจำเป็นอย่างยิ่งสำหรับผู้บริหารที่จะต้องปรับตัวให้ทันต่อการเปลี่ยนแปลงนี้ตลอดจนต้องพยายามฝึกฝนตนเอง โดยพัฒนาทักษะและสั่งสมประสบการณ์ในการตัดสินใจ เพื่อที่จะสามารถวิเคราะห์และตัดสินใจเลือกทางเลือกต่าง ๆ ได้อย่างแม่นยำ มีประสิทธิภาพ และสอดคล้องกับสถานการณ์ที่เกิดขึ้น ดังนั้น ในการวิจัยครั้งนี้ผู้วิจัยจึงได้นำแนวความคิดของนักวิชาการหลายท่านที่อธิบายขั้นตอนการตัดสินใจ เพื่อเป็นแนวทางในการศึกษา

แนวความคิดของ Simon (1960) ที่อธิบายขั้นตอนการตัดสินใจโดยใช้แบบจำลอง (Model) ที่ประกอบด้วยขั้นตอนหลัก 3 ประการ ดังต่อไปนี้

1. การใช้ความคิดประกอบเหตุผล (Intelligence) ผู้ตัดสินใจจะรับรู้ถึงโอกาสหรือปัญหาที่เกิดขึ้น จากนั้นผู้ทำการตัดสินใจเริ่มเก็บรวบรวมข้อมูลที่เกี่ยวข้องจากทั้งตัวปัญหาและสิ่งแวดล้อม หรือโอกาสนั้น

2. การออกแบบ (Design) ผู้ตัดสินใจจะวิเคราะห์และพัฒนาหาทางที่เป็นไปได้ในการแก้ปัญหา เพื่อนำไปใช้ประกอบการตัดสินใจเลือกทางเลือกในการปฏิบัติที่เหมาะสม การที่จะประสบความสำเร็จในขั้นตอนนี้ได้ ผู้ทำการตัดสินใจจะต้องมีความเข้าใจในปัญหา มีความคิดสร้างสรรค์ พยายามที่จะหาทางออกของปัญหา และตรวจสอบความเป็นไปได้ในการแก้ปัญหานั้น

3. การคัดเลือก (Choice) ผู้ทำการตัดสินใจจะทำการคัดเลือกแนวทางปฏิบัติที่เหมาะสมกับสถานการณ์ที่สุด เพื่อที่จะนำไปประยุกต์ใช้ต่อไป

ปกติขั้นตอนการตัดสินใจจะมีการปรับเปลี่ยนอย่างต่อเนื่อง จากขั้นตอนแรกจนถึงขั้นตอนตัดสินใจเลือกทางเลือกเพื่อนำไปปฏิบัติ อย่างไรก็ตามอาจจะมีการดำเนินการย้อนกลับไปยังขั้นตอนที่ผ่านมาแล้วในระหว่างที่ขั้นตอนใด ๆ กำลังดำเนินอยู่ เพื่อปรับปรุงให้การตัดสินใจมีผลสมบูรณ์ขึ้น นอกจากนี้ยังมีผู้วิจารณ์ว่าแบบจำลองกระบวนการตัดสินใจของ Simon ในช่วงเริ่มต้นไม่ได้กล่าวเจาะจงถึงกระบวนการต่าง ๆ หลังการคัดเลือกแนวทางปฏิบัติ เช่น การติดตามผลการวิเคราะห์ผลลัพธ์ที่เกิด ซึ่งต่อมา Rubenstien และ Haberstroh (1965) ได้เสนอแนวความคิดเกี่ยวกับขั้นตอนการตัดสินใจ มี 5 ขั้นตอนดังนี้

1. ผู้ตัดสินใจรับรู้ถึงโอกาส หรือปัญหาที่เกิดขึ้น

2. ผู้ตัดสินใจรวบรวมข้อมูลเกี่ยวกับปัญหา ศึกษาและวิเคราะห์ปัญหา และกำหนดทางเลือกที่เป็นไปได้ เพื่อการวิเคราะห์ทางเลือกในการตัดสินใจ

3. ผู้ตัดสินใจจะทำการตัดสินใจเลือกทางเลือกที่คิดว่าเหมาะสมกับลักษณะของปัญหาและสถานการณ์ เพื่อนำไปปฏิบัติต่อไป

4. ผู้ตัดสินใจจะดำเนินการ เพื่อนำผลการตัดสินใจไปปฏิบัติ

5. ภายหลังการนำผลการตัดสินใจไปดำเนินงาน ต้องทำการติดตามผลของการปฏิบัติเพื่อตรวจสอบว่าการดำเนินงานมีประสิทธิภาพเพียงใด และต้องปรับปรุงให้สอดคล้องกับสถานการณ์อย่างไร

นอกจากนี้ ยังมีแนวคิดของ Long (1989) ซึ่งได้กล่าวไว้ในหนังสือระบบสารสนเทศเพื่อการจัดการ กล่าวว่า การตัดสินใจแบ่งออกเป็น 6 ขั้นตอน ดังนี้

1. การรับรู้ถึงโอกาสหรือปัญหาที่เกิดขึ้น
2. การสำรวจขอบเขตและข้อจำกัดของการตัดสินใจ
3. การกำหนดทางเลือกในการตัดสินใจ
4. การรวบรวมสารสนเทศที่เหมาะสม เพื่อนำมาใช้ประกอบการตัดสินใจ
5. การวิเคราะห์ทางเลือกที่เป็นไปได้
6. การเลือกทางเลือกที่เหมาะสมและนำไปปฏิบัติ

เราจะเห็นได้ว่ามีการสรุปเกี่ยวกับขั้นตอนการตัดสินใจแตกต่างไปตามความเข้าใจแนวทางและเป้าหมายในการอธิบายของผู้รู้แต่ละท่าน ซึ่งก็มีส่วนที่คล้ายคลึงกันและที่แตกต่างกัน (ทวีศักดิ์ นาคม่วง, 2547)

งานวิจัยที่เกี่ยวข้อง

การวิจัยในครั้งนี้มีงานวิจัยอื่น ๆ ที่เกี่ยวข้องที่เป็นปัจจัยพื้นฐานต่อการพัฒนาระบบการเพิ่มประสิทธิภาพการสืบค้นข้อมูลด้วยแฮชฟังก์ชันในฐานข้อมูลด้านการวินิจฉัยโรค ดังนี้

ศันสนีย์ ชาตตระกูล (2542) ได้วิจัยเรื่องการพัฒนาคลังข้อมูลเพื่อระบบสนับสนุนการตัดสินใจกรณีศึกษา : ระบบแถวคอย แผนกอายุรกรรม โรงพยาบาลนครธน การวิจัยนี้เป็นการนำแนวคิดเรื่องคลังข้อมูลมาประยุกต์เข้ากับระบบสนับสนุนการตัดสินใจเพื่อใช้สำหรับโรงพยาบาล ปัจจุบันโรงพยาบาลได้มีการจัดทำระบบจัดการฐานข้อมูล ไว้เก็บข้อมูลจำนวนมาก แต่ไม่ได้นำข้อมูลเหล่านั้นมาใช้ประโยชน์อย่างเต็มที่ เนื่องจาก ไม่ได้นำข้อมูลที่มีอยู่มาทำการวิเคราะห์และสรุปผล การวิจัยนี้จัดทำรายงานสรุปทั่วไปเกี่ยวกับผู้ป่วย รายรับโรงพยาบาล และรายงานสรุปเพื่อประกอบการตัดสินใจในเรื่องระบบแถวคอย เพื่อแสดงให้ผู้บริหารทราบว่า ในปัจจุบันจำนวนแพทย์ผู้ให้การรักษามีจำนวนพอเหมาะกับจำนวนผู้ป่วยที่มารับบริการหรือไม่ผลการวิจัยทำให้ทราบถึงผลการดำเนินงานที่ผ่านมาเกี่ยวกับผู้ป่วยรายรับของโรงพยาบาล และผลการวิเคราะห์ระบบแถวคอยแสดงให้เห็นว่ามีบางช่วงเวลาที่ยังมีแถวคอยอยู่มากหรือบางช่วงแพทย์ผู้ให้บริการว่าง จากผลการวิเคราะห์นี้เป็นแนวทางให้ผู้บริหารใช้ประกอบการตัดสินใจในการเลือกวางแผนกำหนดจำนวนแพทย์ในโรงพยาบาลต่อไป โดยผู้บริหารควรนำปัจจัยอื่น เช่น ค่าจ้างแพทย์มา

ประกอบการตัดสินใจด้วย การพัฒนาคลังข้อมูลเพื่อประกอบการตัดสินใจได้แสดงผล อย่างชัดเจน ว่าสามารถทำให้การดึงข้อมูลสรุปเป็นไปอย่างรวดเร็วและนำข้อมูลที่มีอยู่มาใช้ให้เกิดประโยชน์

อุไรพร เจตนชัย (2544) ได้วิจัยเรื่องการเปรียบเทียบความเร็วในการค้นหาข้อมูลด้วยคำสั่ง SQL ในคลังข้อมูลที่ออกแบบด้วย Star Schema Model และ Snowflake Schema Model งานวิจัยนี้เสนอการเปรียบเทียบความเร็วในการค้นหาข้อมูลใน Star Schema Model และ Snowflake Schema Model ด้วยคำสั่ง SQL โดยยกฝ่ายการเงินและฝ่ายทะเบียนของมหาวิทยาลัยเทคโนโลยีมหานครเป็นกรณีศึกษาเพื่อออกแบบโมเดลทั้งสอง หลังจากที่วัดความเร็วในการค้นหาข้อมูลจากคลังข้อมูลที่ออกแบบโดยโมเดลทั้งสองแล้ว นำความเร็วที่วัดได้มาวิเคราะห์การถดถอยและสัมประสิทธิ์สหสัมพันธ์เพื่อทดสอบว่าตัวแปรแต่ละตัวมีผลต่อเวลาในการค้นหาข้อมูลมากน้อยเพียงใด สุดท้ายทำการสรุปว่าการออกแบบคลังข้อมูลด้วย Model ใด ทำให้การค้นหาข้อมูลทำได้ดีกว่า

วราภรณ์ วีระวิทยา (2538) ได้วิจัยเรื่องการพัฒนาภาษาสอบถามฐานข้อมูล SQL งานวิทยานิพนธ์นี้สร้างตัวแปลภาษาสอบถามฐานข้อมูล SQL ที่เป็นมาตรฐานของ ANSI และเพิ่มเติมบางรูปแบบของข้อความสอบถามเพื่อให้สามารถครอบคลุมการดำเนินงานสอบถามฐานข้อมูลได้มากขึ้น โดยรูปแบบของข้อความสอบถามที่เพิ่มเติมได้ออกแบบและเลือกมาจาก SQL ที่ใช้ในระบบจัดการฐานข้อมูลโอราเคิล โดยเน้นการใช้รูปแบบที่สั้นและเข้าใจ ความหมายได้ง่าย ตัวแปลภาษาที่ถูกสร้างจะครอบคลุมการตรวจสอบวากยสัมพันธ์และ การตีความหมายของแต่ละข้อความที่สั่ง

เมื่อได้ทำการศึกษางานวิจัยที่เกี่ยวข้องในการทำวิจัยครั้งนี้เพื่อเป็นพื้นฐานความรู้สำหรับการทำวิจัย ผู้วิจัยจึงได้กำหนดวิธีการดำเนินการวิจัย ดังจะกล่าวในบทต่อไป