บทที่ 4

รายละเอียดของการออกแบบและการพัฒนา โครงสร้างข้อมูลสำหรับพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

รายละเอียดการออกแบบโครงสร้างข้อมูลสำหรับเก็บพจนานุกรม จะแสดงพร้อมอัลกอริทึมการสืบค้น การเพิ่มคำศัพท์ และตัวอย่างประกอบ เพื่อให้เข้าใจได้รวดเร็วยิ่งขึ้น

ลักษณะโครงสร้างข้อมูลที่ศึกษาจากงานวิจัย

ในบทที่ 2 ได้กล่าวถึงงานวิจัยเกี่ยวกับรูปแบบโครงสร้างข้อมูลสำหรับเก็บพจนานุกรมอิเล็กทรอนิกส์ ซึ่งโครงสร้างข้อมูลแต่ละรูปแบบมีข้อดี ข้อด้อยที่แตกต่างกัน การเพิ่มประสิทธิภาพการทำงานของโครงสร้างข้อมูลดังกล่าวกระทำได้หลายวิธี เช่น การปรับวิธีการสืบค้นคำศัพท์ให้รวดเร็วยิ่งขึ้น การลดขนาดข้อมูล (Data Compression) เป็นต้น แต่การเพิ่มประสิทธิภาพโดยใช้วิธีการอย่างหนึ่งอาจมีผลกระทบต่อประสิทธิภาพอีกอย่างหนึ่ง เช่น ฟังก์ชันการลดขนาดข้อมูลนั้นเป็นวิธีการที่แปลงคำศัพท์ให้อยู่ในรูปแบบพิเศษเพื่อเก็บในสื่อบันทึกข้อมูล เมื่อต้องการใช้ข้อมูลจึงอ่านข้อมูลจากสื่อบันทึกข้อมูลแล้วแปลงข้อมูลดังกล่าวให้อยู่ในรูปแบบที่สามารถเข้าใจได้ จากตัวอย่างดังกล่าวเห็นได้ว่า ขั้นตอนการแปลงคำศัพท์ก่อนบันทึกหรือหลังจากอ่านจากสื่อบันทึกนั้นต้องใช้เวลาช่วงหนึ่ง ซึ่งจะมีผลทำให้เวลาที่ใช้ในการสืบค้นคำศัพท์นานขึ้น ดังนั้นในการเลือกวิธีการใดวิธีการหนึ่งเพื่อเพิ่มประสิทธิภาพการทำงานจะต้องพิจารณาสิ่งอื่น ๆ ประกอบด้วย เช่น วิธีการที่ใช้มีผลกระทบต่อการทำงานอย่างอื่นหรือไม่อย่างไร หรือถ้าจะเพิ่มขั้นตอนดังกล่าวแล้วจะแทรกไว้ขั้นตอนใดจึงเหมาะสม เป็นต้น

การศึกษาโครงสร้างข้อมูลที่จะนำมาใช้เก็บพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยสำหรับวิทยานิพนธ์ครั้งนี้ แบ่งเป็น 2 ส่วน ส่วนแรกทำการศึกษารูปแบบโครงสร้างข้อมูลที่ใช้เก็บพจนานุกรมอิเล็กทรอนิกส์ โดยศึกษาจากรูปแบบของโครงสร้างข้อมูลที่มีผู้ทำการวิจัยไว้แล้วนำโครงสร้างเหล่ามาเป็นเค้าโครงในการพิจารณาว่าเหมาะกับลักษณะของคำไทยหรือไม่ นอกจากนั้นยังศึกษาโครงสร้างข้อมูลอื่นอีกด้วย ในส่วนที่ 2 เป็นการศึกษายุทธวิธีปฏิบัติการกับโครงสร้างข้อมูลนั้น ๆ ทั้งการสืบค้นข้อมูล การเพิ่มข้อมูล และการลบข้อมูลว่า

แต่ละวิธีมีข้อดีข้อด้อยอย่างไร

หลังจากที่ศึกษารายละเอียดต่าง ๆ แล้วจึงนำมาพัฒนาโปรแกรมและทดสอบประสิทธิภาพการทำงาน โดยทดสอบด้านความเร็วในการสืบค้นคำศัพท์ และปริมาณเนื้อที่ที่ใช้สำหรับเก็บพจนานุกรม การทำวิทยานิพนธ์ครั้งนี้ได้นำโครงสร้างแบบดับเบิลอะเรย์มาใช้แล้วปรับวิธีการสืบค้น การเพิ่ม และการลบคำศัพท์ที่เหมาะกับลักษณะคำไทย

โครงสร้างข้อมูลสำหรับเก็บพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยที่ใช้ในวิทยานิพนธ์นี้ใช้วิธีการสืบค้นข้อมูลแบบดิจิตอลที่มีความเร็วเทียบเท่ากับการสืบค้นแบบสถิต นอกจากนี้ประสิทธิภาพในการปรับให้ทันกาลของข้อมูลยังไม่ด้อยกว่าขั้นตอนวิธีการแบบพลวัต ทั้งนี้ยังสอดคล้องกับลักษณะคำไทยก็ตรงที่การปฏิบัติการกับคำศัพท์นั้นพิจารณาคำศัพท์ครั้งละตัวอักษร โดยโครงสร้างที่นำมาใช้คือโครงสร้างข้อมูลแบบดับเบิลอะเรย์ แล้วนำเอาลักษณะเด่นของคำไทยตรงที่เป็น "คำโดด" มาทำการประยุกต์กับวิธีการสืบค้น และการเพิ่มคำศัพท์ของโครงสร้างแบบดับเบิลอะเรย์

รูปแบบโครงสร้างข้อมูลสำหรับพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

รูปแบบโครงสร้างข้อมูลที่นำเสนอในวิทยานิพนธ์นี้ เป็นโครงสร้างที่ประกอบด้วย มัลติโหนด เซพพาเรทโหนด ซิงเกิลโหนด และเส้นทางเดินระหว่างโหนด โดยที่เส้นทางเดินระหว่างโหนดของมัลติโหนดหรือเซพพาเรทโหนดเก็บในอะเรย์ 1 มิติ 2 อะเรย์ คือ อะเรย์เบส อะเรย์เช็ค ซิงเกิลสตริงจะจัดเก็บไว้ในเทล และค่าของคำโดดเก็บในอะเรย์เก็บค่า รูปที่ 4.1 แสดงความสัมพันธ์ของ อะเรย์ต่าง ๆ

1. <u>อะเรย์เบส</u> อะเรย์เบสประกอบด้วย 2 ส่วนคือ ดรรชนีและค่าในอะเรย์ ดรรชนีใช้แทนหมายเลขมัลติโหนดหรือเซพพาเรทโหนด ส่วนค่าในอะเรย์มี 2 ประเภทคือ ถ้ามีค่าบวกหมายถึงตัวเลขที่ใช้สำหรับคำนวณเส้นทางเดินของโหนดหนึ่งไปยังอีกโหนดหนึ่ง แต่ถ้าค่าเป็นลบหมายถึงตำแหน่งของซิงเกิลสตริงในอะเรย์เทล

ค่าที่เก็บในอะเรย์ที่ 1 คือดรรชนีสูงสุดของอะเรย์เทล

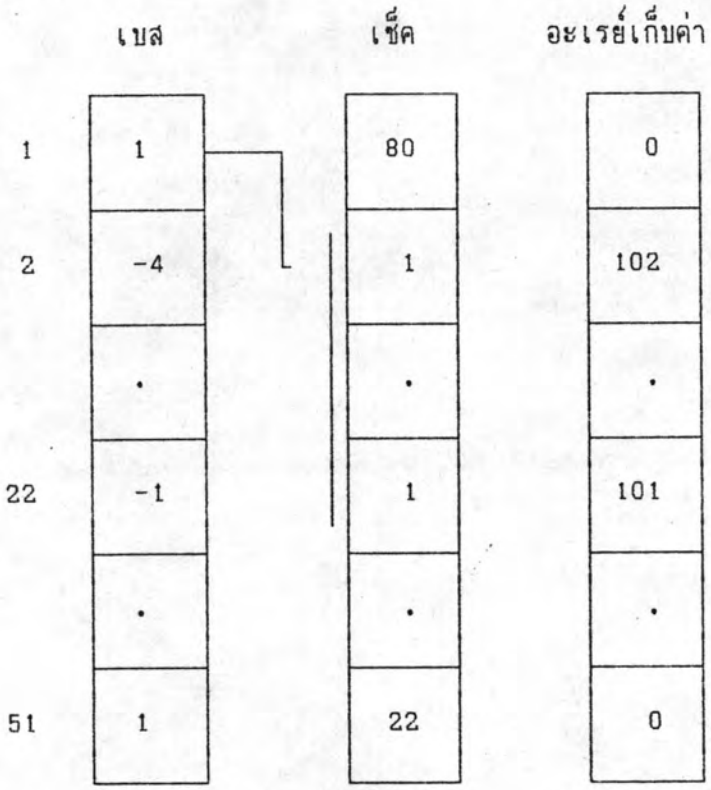
2. <u>อะเรย์เช็ค</u> อะเรย์เช็คประกอบด้วย 2 ส่วนคือ ดรรชนีและค่าในอะเรย์ ดรรชนีใช้แทนหมายเลขโหนด ส่วนค่าในอะเรย์คือหมายเลขพาเรนท์โหนด (parent node) เช่น โหนดหมายเลข 5 สร้างเส้นทางเดินไปยังโหนดที่ 12 เพราะฉะนั้นค่าของ อะเรย์เช็คที่ 12 มีค่าเป็น 5

ค่าที่เก็บในอะเรย์ที่ 1 คือดรรชนีสูงสุดของอะเรย์เบสและเช็ค

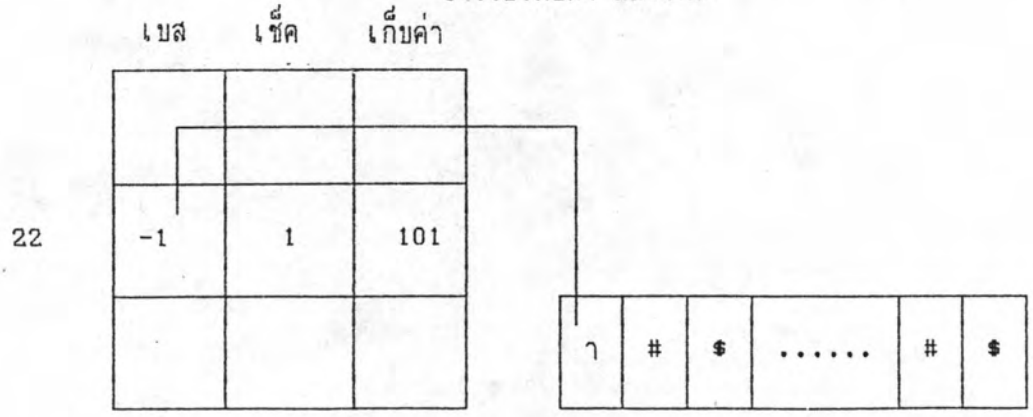
3. อะเรย์เทล เก็บซิงเกิลสตริง ประกอบด้วยตัวอักษร สัญญลักษณ์สิ้นสุดคำ "#" สัญญลักษณ์แบ่งซิงเกิลสตริง "$" และสัญญลักษณ์ระบุตำแหน่งที่ไม่ได้ใช้งาน (Garbage Symbol) "?"

4. อะเรย์เก็บค่า เป็นอะเรย์ที่เพิ่มขึ้นเพื่อใช้เก็บค่าของคำโดดในขณะเก็บคำศัพท์ และสืบค้นคำศัพท์

ขณะเริ่มต้นทำงานนั้นค่าที่เก็บในอะเรย์เบส อะเรย์เช็ค อะเรย์ที่ 1 เป็น 1



รูปที่ 4.1 (ก) แสดงความสัมพันธ์ระหว่างอะเรย์เบส อะเรย์เช็ค อะเรย์เก็บค่า และเทล

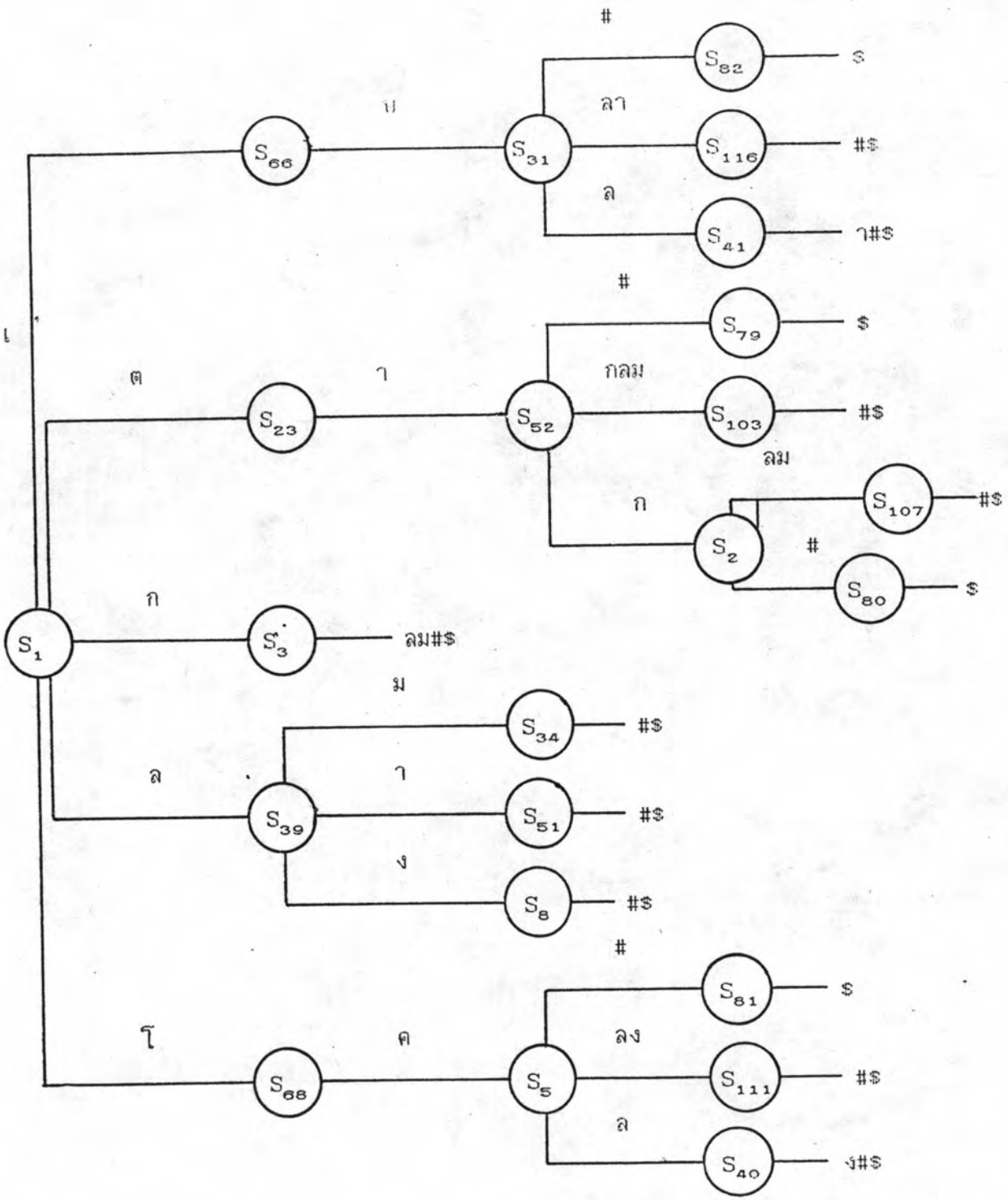


รูปที่ 4.1 (ข) แสดงความสัมพันธ์ระหว่างอะเรย์เบส อะเรย์เช็ค อะเรย์เก็บค่า และเทล

ดรรชนี	ค่าอะเรย์เบส	ค่าอะเรย์เช็ค	ค่าอะเรย์เก็บค่า
1	2	116	0
2	2	52	0
3	-4	1	102
5	3	68	0
8	-20	39	108
23	2	1	0
31	4	66	0
34	-12	39	105
39	1	1	0
40	-24	5	110
41	-34	31	115
51	-30	39	113
52	1	23	0
66	1	1	0
68	1	1	0
79	-1	52	101
80	-10	2	104
81	-17	5	107
82	-27	31	112
103	-8	52	0
107	-15	2	0
111	-22	5	0
116	-32	31	0

เทล	$??ลม#$#$$?#$?#$$??#$#$ง#$$??#$#$า#$

รูปที่ 4.2 แสดงรูปแบบการเก็บข้อมูลของโครงสร้างข้อมูลสำหรับเก็บพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย



รูปที่ 4.3 โหนดต่างๆ ของโครงสร้างข้อมูลสำหรับเก็บพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

ตามที่ได้กล่าวในตอนต้นแล้วว่า โครงสร้างข้อมูลที่จะนำเสนอในวิทยานิพนธ์นี้ได้นำลักษณะเด่นของคำภาษาไทยคือ "คำโดด" มาใช้ในการปรับวิธีการสืบค้น และเพิ่มคำศัพท์ รูปที่ 4.2 และ 4.3 เป็นตัวอย่างของโครงสร้างข้อมูลสำหรับพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยที่เพิ่มคำศัพท์ชุดหนึ่งที่ประกอบด้วยคำเหล่านี้คือ เพลา, เพ ลา, ตาก ลม, ตา กลม, โคลง, โค ลง

อัลกอริทึมการสืบค้นคำศัพท์

อัลกอริทึมการสืบค้นคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยที่ได้ออกแบบในวิทยานิพนธ์นี้ เป็นอัลกอริทึมที่ให้ผลลัพธ์เป็นชุดของคำศัพท์ที่มีการแยกคำซึ่งทำให้เห็นความหมายของคำได้ชัดเจนยิ่งขึ้น แต่ความถูกต้องของผลลัพธ์ที่ได้ขึ้นกับความถูกต้องของการตัดแบ่งคำศัพท์ขณะเพิ่มคำศัพท์ในพจนานุกรม สำหรับวิธีการสืบค้นคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยมีรายละเอียดดังนี้

1. อัลกอริทึมการสืบค้นคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

อัลกอริทึมการสืบค้นคำศัพท์ที่จะกล่าวต่อไปจะอธิบายขั้นตอนการทำงานพร้อมยกตัวอย่างประกอบ โดยที่อัลกอริทึมการสืบค้นคำศัพท์ประกอบด้วยฟังก์ชันต่าง ๆ ดังนี้

1.1 ฟังก์ชัน SearchDict

เป็นฟังก์ชันที่ใช้สำหรับสืบค้นคำศัพท์ว่ามีในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยหรือไม่ หากค้นพบจะแสดงให้เห็นว่าจัดเก็บคำศัพท์อย่างไร เช่นต้องการค้นหาคำว่า "ตากลม" มีในพจนานุกรมหรือไม่ ผลลัพธ์ที่ได้จากฟังก์ชันนี้จะพบว่ามีคำศัพท์คำดังกล่าวในพจนานุกรม และถูกเก็บไว้ 2 รูปแบบคือ "ตา กลม" และ "ตาก ลม" โดยมีขั้นตอนการทำงานของฟังก์ชันดังนี้

$SepTable_t$ แทนอะเรย์เก็บตำแหน่งสิ้นสุดคำโดด

$NodeTable_t$ แทนอะเรย์เก็บหมายเลขโหนดก่อนหน้าโหนดที่ระบุว่าเป็นโหนดสิ้นสุดคำโดด

m_t แทนดรรชนีสูงสุดของอะเรย์

1. กำหนดตำแหน่งตัวอักษรที่นำมาเป็นอินพุท เท่ากับ 0 และ หมายเลขโหนดก่อนหน้า เท่ากับ 1
2. หาตำแหน่งสิ้นสุดคำและหมายเลขโหนดก่อนหน้าโดยใช้ฟังก์ชัน RetrieveData แล้วเก็บค่าดังกล่าวไว้ใน $SepTable_1$ และ $NodeTable_1$ ซึ่งมีจำนวนข้อมูลทั้งหมดเท่ากับ m_1
3. ใช้คำศัพท์ที่ได้จาก SepTable หมายเลขโหนดจาก NodeTable โดยมีวิธีการดังนี้
   - 3.1 สร้าง $SepTable_2$ และ $NodeTable_2$ จากการตัดคำโดดคำที่ 1 ออกจากคำศัพท์โดยใช้คำโดดที่ได้จาก $SepTable_1$ แล้วใช้หมายเลขโหนดก่อนหน้าชุดแรกจาก $NodeTable_1$
   - 3.2 สร้าง $SepTable_3$ และ $NodeTable_3$ จากการตัดคำโดดคำที่ 2 ออกจากคำศัพท์ส่วนที่เหลือจากข้อ 3.1 โดยใช้คำโดดที่ได้จาก $SepTable_2$ แล้วใช้หมายเลขโหนดก่อนหน้าชุดแรกจาก $NodeTable_2$
   - 3.3 สร้าง SepTable และ NodeTable ชุดต่อ ๆ ไป ตามวิธีการข้างต้น จนคำศัพท์หมด โดยอะเรย์ชุดสุดท้ายที่สร้างได้เป็นชุดที่ n แสดงว่าคำศัพท์ที่ประกอบด้วยคำโดดชุดดังกล่าวมีในพจนานุกรม แต่ถ้าไม่สามารถสร้างได้แสดงว่าไม่มีในพจนานุกรม
   - 3.4 ย้อนการทำงานกลับมาทำงานในอะเรย์ชุดที่ n-1 โดยใช้คำโดดที่ได้จาก SepTable และ NodeTable ชุดดังกล่าวที่ยังไม่ได้ใช้งาน แล้วสร้างอะเรย์ชุดที่ n ใหม่ ต่อจากนั้นให้ใช้ข้อมูลในอะเรย์ชุดที่ n-1 จนหมด
   - 3.5 เมื่อข้อมูลในอะเรย์ชุดที่ n-1 หมดให้ย้อนการทำงานกลับมาทำงานในอะเรย์ชุดที่ n-2 โดยใช้คำโดดที่ได้จาก SepTable และ NodeTable ชุดดังกล่าวที่ยังไม่ได้ใช้งาน เพื่อสร้างอะเรย์ชุดที่ n-1 ใหม่ แล้วเลือกคำโดดที่สร้างจากอะเรย์ชุดที่ n-1 ที่จะไปสร้างอะเรย์ชุดที่ n ใหม่
   - 3.6 ย้อนการทำงานกลับมาทำงานในอะเรย์ชุดก่อนหน้าเรื่อยไปจนกระทั่งอะเรย์ชุดที่ 1 และข้อมูลในอะเรย์ชุดที่ 1 หมด

3.7 ขณะที่คำโดดที่ได้จากอะเรย์ชุด 1 คำใดสามารถสร้างชุดของอะเรย์จนกระทั่งถึงชุดที่ n ได้แสดงว่ามีคำศัพท์ที่มีการตัดคำที่ตำแหน่งดังกล่าวอยู่ในพจนานุกรม

2. ฟังก์ชัน RetrieveData

เป็นฟังก์ชันที่ค้นหาคำโดดทั้งหมดที่สร้างมาจากโหนดที่กำหนด แล้วเก็บตำแหน่งสิ้นสุดคำโดด และหมายเลขโหนดก่อนหน้าในอะเรย์

r แทนหมายเลขโหนดปัจจุบัน

t แทนหมายเลขโหนดที่สร้างจากโหนด r

h แทนตำแหน่งของคำศัพท์ที่ใช้สร้างโหนด

ขั้นตอนการทำงานของฟังก์ชันมีดังนี้

1. กำหนดหมายเลขโหนดเป็น r และ h เป็นตำแหน่งของคำศัพท์
2. ตรวจสอบว่าโหนด r เป็นโหนดประเภทใด โดยพิจารณาว่าค่าเบสอะเรย์ที่ r เป็นค่าใดดังต่อไปนี้

   ค่าบวก แสดงว่าโหนด r เป็นมัลติโหนด

   ค่าลบ แสดงว่าเป็นตำแหน่งของชิงเกิลสตริงในเทล ทำงานในข้อ 6 ต่อ

   ค่าเท่ากับ 0 แสดงว่าโหนดนี้ยังไม่ได้ใช้งาน ออกจากฟังก์ชัน

   และตรวจสอบว่าคำศัพท์ที่ใช้หมดหรือยัง ถ้าหมดแล้วออกจากฟังก์ชัน
3. สร้างโหนดถัดไปจากโหนด r และตัวพยัญชนะตัวที่ h ของคำศัพท์โดยมีขั้นตอนดังนี้

   3.1 คำนวณค่าของโหนดถัดไปจากสมการ

   โหนดถัดไป = ค่าในเบสอะเรย์ที่ r + ค่าของพยัญชนะตำแหน่ง h

   โดยที่ค่าของพยัญชนะหามาจากฟังก์ชัน WordToInt

   3.2 ตรวจสอบว่าหมายเลขโหนดถัดไปมีค่าเกินกว่าขนาดอะเรย์หรือไม่ ถ้าเกินขนาดอะเรย์ออกจากฟังก์ชัน

3.3 ตรวจสอบว่าโหนดที่สร้างนี้เป็นโหนดที่สร้างมาจากโหนด r หรือไม่ โดยพิจารณาจากค่าเช็คอะเรย์เดียวกันกับหมายเลขโหนดใหม่ว่าเท่ากับ r หรือไม่ ถ้าไม่เท่าออกจากฟังก์ชัน

4. พิจารณาว่าโหนด r เป็นตำแหน่งสิ้นสุดคำโดดได้หรือไม่ โดยมีขั้นตอนดังนี้

4.1 คำนวณค่าของโหนดถัดไปจากสมการ

โหนดถัดไป = ค่าในเบสอะเรย์ที่ r +
ค่าของ '#'

4.2 ตรวจสอบว่าหมายเลขโหนดถัดไปมีค่าเกินกว่าขนาดอะเรย์หรือไม่ ถ้าเกินขนาดอะเรย์แสดงว่าไม่ใช่ตำแหน่งสิ้นสุดคำโดด

4.3 ตรวจสอบว่าโหนดที่สร้างนี้เป็นโหนดที่สร้างมาจากโหนด r หรือไม่ โดยพิจารณาจากค่าเช็คอะเรย์เดียวกันกับหมายเลขโหนดใหม่ว่าเท่ากับ r หรือไม่ ถ้าไม่เท่าแสดงว่าไม่ใช่ตำแหน่งสิ้นสุดคำโดด

ถ้าเป็นตำแหน่งสิ้นสุดคำโดดเก็บตำแหน่ง และหมายเลขโหนด r ในอะเรย์

5. สร้างโหนดถัดไปจากโหนด t โดยกำหนดโหนด r เป็น t และพยัญชนะตัวถัดไปคือ h แล้วหาตำแหน่งสิ้นสุดคำโดดตามข้อ 2 - 5 จนกระทั่งคำศัพท์หมดหรือโหนดที่พิจารณาไม่ใช่มัลติโหนด

6. ดึงชิงเกิลสตริงจากอะเรย์เทลที่ตำแหน่งตามค่าในเบส

7. เปรียบเทียบคำศัพท์ส่วนที่เหลือกับชิงเกิลสตริงว่าเหมือนกันหรือไม่ ถ้าเหมือนกันเก็บค่าตำแหน่งสิ้นสุดคำโดดและหมายเลขโหนดในอะเรย์

1.3 ฟังก์ชัน WordToInt

เป็นฟังก์ชันที่ใช้สำหรับหาค่าของคำโดดหรือค่าของพยัญชนะ ซึ่งมีพารามิเตอร์ cnvflag กำหนดว่าต้องการหาค่าของพยัญชนะหรือคำศัพท์ สำหรับขั้นตอนการหาค่าของคำศัพท์มีดังนี้

1. ถ้า cnvflag = 0 แสดงว่าต้องการหาค่าของพยัญชนะ ที่ส่งมา หาค่าพยัญชนะโดยใช้ฟังก์ชัน CharToInt ส่วนตำแหน่งพยัญชนะคือพยัญชนะตัวถัดไปของคำศัพท์
2. ถ้า cnvflag = 9 แสดงว่าต้องการหาค่าของคำโดด โดยใช้ใช้ฟังก์ชัน RetrieveWord ในการหาค่าคำโดด และตำแหน่งสิ้นสุดคำโดด

2. ตัวอย่างการสืบค้นคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

สมมติว่าพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยจัดเก็บคำศัพท์ชุดหนึ่ง โดยใช้โครงสร้างข้อมูล ตามที่นำเสนอไว้ในบทที่ 4 นี้ และต้องการสืบค้นคำว่า "ตากลม" ว่ามีในพจนานุกรมหรือไม่ สำหรับขั้นตอนการสืบค้นคำศัพท์ แสดงตามตาราง 4.1

	อินพุท	การปฏิบัติการ	เอาท์พุท
1.	"ตากลม#"	หาคำศัพท์ที่อยู่ในพจนานุกรมและเก็บในอะเรย์	{ตา,ตาก} {52,2} {101,104}
2.	ตากลม# ตา	ดึงคำศัพท์ชุดแรกจากอะเรย์มาพิจารณา โดยตัดออกจากคำศัพท์ที่เป็นอินพุท	กลม#
3.	กลม#	หาคำศัพท์ที่อยู่ในพจนานุกรมและเก็บในอะเรย์	{กลม} {0} {102}
4.	ตา กลม	ตรวจสอบว่าสามารถสร้างโหนดใหม่จาก โหนด 52 และ "กลม" ได้หรือไม่ BASE[52] + 102 = 103 CHECK[103] = 52	สร้างโหนด ใหม่ได้
5.	กลม# กลม	ดึงคำว่า "กลม" ออกจากคำศัพท์ที่เป็นอินพุท ส่วนที่เหลือ	#
6.	#	หาคำศัพท์ที่อยู่ในพจนานุกรมและเก็บในอะเรย์ ปรากฏว่าสิ้นสุดคำศัพท์แล้ว จึงตรวจสอบต่อว่า สามารถสร้างโหนดใหม่จากโหนด 103 ได้ หรือไม่ BASE[103] = -8 ปรากฏว่าค่า BASE เป็นลบแสดงว่าต้อง ไปดึงซิงเกิลสตริงในอะเรย์เทล	ซิงเกิลสตริง "#"
7.	คำศัพท์ "#" ซิงเกิลสตริง "#"	เปรียบเทียบคำศัพท์ที่เป็นอินพุทส่วนที่เหลือกับ ซิงเกิลสตริง ปรากฏว่าเหมือนกัน	-1

ตารางที่ 4.1 แสดงขั้นตอนวิธีการสืบค้นคำศัพท์

	อินพุท	การปฏิบัติการ	เอาท์พุท
8.		พบว่ามีคำว่า "ตา กลม" ในพจนานุกรม	
9.	ตากลม# ตาก	ดึงคำศัพท์ชุดที่ 2 จากอะเรย์มาพิจารณา โดยตัดออกจากคำศัพท์ที่เป็นอินพุท	ลม#
10.	ลม#	หาคำศัพท์ที่อยู่ในพจนานุกรมและเก็บในอะเรย์	{ลม} {0} {104}
11.	ตาก ลม	ตรวจสอบว่าสามารถสร้างโหนดใหม่จาก โหนด 2 และ "ลม" ได้หรือไม่ BASE[2] + 104 = 107 CHECK[107] = 2	สร้างโหนด ใหม่ได้
12.	ลม# ลม	ดึงคำว่า "ลม" ออกจากคำศัพท์ที่เป็นอินพุท ส่วนที่เหลือ	#
13.	#	หาคำศัพท์ที่อยู่ในพจนานุกรมและเก็บในอะเรย์ ปรากฏว่าสิ้นสุดคำศัพท์แล้ว จึงตรวจสอบต่อว่า สามารถสร้างโหนดใหม่จากโหนด 107 ได้ หรือไม่ BASE[107] = -15 ปรากฏว่าค่า BASE เป็นลบแสดงว่าต้อง ไปดึงซิงเกิลสตริงในอะเรย์เทล	ซิงเกิลสตริง "#"
14.	คำศัพท์ "#" ซิงเกิลสตริง "#"	เปรียบเทียบคำศัพท์ที่เป็นอินพุทส่วนที่เหลือกับ ซิงเกิลสตริง ปรากฏว่าเหมือนกัน	-1
15.		พบว่ามีคำว่า "ตาก ลม" ในพจนานุกรม เพราะฉะนั้นจากการสืบค้นคำ "ตากลม" ครั้งนี้จะได้คำศัพท์ออกมาเป็น 2 คำก็คือ "ตา กลม" และ "ตาก ลม"	

ตารางที่ 4.1 (ต่อ) แสดงขั้นตอนวิธีการสืบค้นคำศัพท์

อัลกอริทึมการเพิ่มคำศัพท์

การเพิ่มคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยประกอบด้วยขั้นตอนต่าง ๆ ดังนี้

1. การแยกคำศัพท์ออกเป็นคำโดดโดยใช้เว้นวรรค
2. เพิ่มคำโดดทั้งหมดในพจนานุกรม
3. เพิ่มคำศัพท์ที่แยกคำโดดแล้วในพจนานุกรม

การแยกคำศัพท์ออกเป็นคำโดดนั้นจะใช้คนเป็นผู้พิจารณา ส่วนการเพิ่มคำโดดและคำศัพท์จะใช้อัลกอริทึมการเพิ่มคำศัพท์ซึ่งมีรายละเอียดการทำงานดังนี้

1. อัลกอริทึมการเพิ่มคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

อัลกอริทึมการเพิ่มคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยประกอบด้วยฟังก์ชั่นต่าง ๆ ดังต่อไปนี้

1.1 ฟังก์ชัน InsertDict

เป็นฟังก์ชันที่ใช้สำหรับเพิ่มคำศัพท์ในพจนานุกรม ขั้นตอนการทำงานของฟังก์ชันนี้มีดังนี้

1. เพิ่มคำโดดแต่ละคำในพจนานุกรมโดยใช้ฟังก์ชัน AddDict
2. นำคำศัพท์ทั้งคำเพิ่มในพจนานุกรมโดยใช้ฟังก์ชัน AddDict

1.2 ฟังก์ชัน AddDict

เป็นฟังก์ชันที่ใช้เพิ่มคำศัพท์ในพจนานุกรม ทั้งคำโดด และคำศัพท์ทั้งคำ โดยฟังก์ชันนี้จะเก็บคำโดดและค่าตัวเลขที่ใช้แทนคำโดดคำนั้นในพจนานุกรม สำหรับค่าของคำโดดจะใช้ในขณะสร้างโหนดต่าง ๆ ของฟังก์ชันการเพิ่มและค้นหาคำศัพท์ ขั้นตอนการทำงานมีดังนี้

1. สร้างมัลติโหนดโดยใช้โหนดที่ 1 และคำศัพท์โดยเริ่มต้นที่ตำแหน่งแรก โดยมีขั้นตอนดังนี้

1.1 ตรวจสอบค่าเบสอะเรย์ที่ 1
น้อยกว่า 0 ทำงานตามข้อ 1.4
มากกว่าหรือเท่ากับ 0 ทำงานในข้อ 1.2

1.2 คำนวณหมายเลขโหนดถัดไปจากสมการ
หมายเลขโหนดถัดไป = ค่าเบสอะเรย์ที่ 1
+ ค่าของพยัญชนะตัวแรกของคำศัพท์

1.3 ตรวจสอบว่าเป็นโหนดที่สร้างจากโหนดที่ 1 หรือไม่
ถ้าใช่ ให้สร้างโหนดต่อไปตามขั้นตอนที่ 2
ถ้าไม่ใช่ สร้างมัลติโหนดและซิงเกิลสตริงโดยใช้
ฟังก์ชัน AInsert(1,คำศัพท์ส่วนที่เหลือ
หลังจากตัดพยัญชนะตัวแรกออก)
ออกจากฟังก์ชัน

1.4 ดึงซิงเกิลสตริงจากอะเรย์เทลโดยใช้ฟังก์ชัน FetchStr
ตำแหน่งใช้ค่าของเบสอะเรย์ที่ 1

1.5 เปรียบเทียบคำศัพท์ส่วนที่เหลือหลังจากดึงพยัญชนะที่ที่
ใช้สร้างโหนดออกกับซิงเกิลสตริง
เหมือนกัน แสดงว่าคำศัพท์มีในพจนานุกรมแล้วออกจาก
ฟังก์ชันพร้อมกับระบุว่าไม่เพิ่มคำศัพท์
เนื่องจากมีคำศัพท์แล้ว
ไม่เหมือนกัน สร้างมัลติโหนดและซิงเกิลสตริงโดยใช้
ฟังก์ชัน BInsert(1,คำศัพท์ที่ซ้ำกัน,
ซิงเกิลสตริงหลังจากตัดส่วนที่ซ้ำ,
คำศัพท์ส่วนที่เหลือหลังจาก
ตัดส่วนที่ซ้ำ)
ออกจากฟังก์ชัน

2. กำหนดให้โหนดใหม่ที่สร้างได้เป็นโหนด t
สร้างมัลติโหนดโดยใช้โหนด t และคำศัพท์ส่วนที่เหลือ
โดยมีขั้นตอนดังนี้

2.1 ตรวจสอบว่าคำศัพท์ที่ใช้หมดหรือยัง ถ้าหมดแล้ว
แสดงว่าคำศัพท์มีในพจนานุกรมแล้ว ออกจากฟังก์ชัน
พร้อมกับระบุว่าไม่เพิ่มคำศัพท์เนื่องจากมีคำศัพท์แล้ว

2.2 ตรวจสอบค่าเบสอะเรย์ที่ t
น้อยกว่า 0 ทำงานตามข้อ 2.4
มากกว่าหรือเท่ากับ 0 ทำงานในข้อ 2.2

2.2 คำนวณหมายเลขโหนดถัดไปจากสมการ
หมายเลขโหนดถัดไป = ค่าเบสอะเรย์ที่ t
+ ค่าของคำโดดหรือพยัญชนะตัวแรกของคำศัพท์ที่เหลือ

2.3 ตรวจสอบว่าเป็นโหนดที่สร้างจากโหนดที่ t หรือไม่
ถ้าใช่ ให้สร้างโหนดต่อไปตามขั้นตอนที่ 3
ถ้าไม่ใช่ สร้างมัลติโหนดและชิงเกิลสตริงโดยใช้
ฟังก์ชัน AInsert(t,คำศัพท์ส่วนที่เหลือ
หลังจากตัดคำโดดหรือพยัญชนะตัวแรกออก)
ออกจากฟังก์ชัน

2.4 ดึงชิงเกิลสตริงจากอะเรย์เทลโดยใช้ฟังก์ชัน FetchStr
ตำแหน่งใช้ค่าของเบสอะเรย์ที่ t

2.5 เปรียบเทียบคำศัพท์ส่วนที่เหลือหลังจากดึงพยัญชนะที่ที่
ใช้สร้างโหนดออกกับชิงเกิลสตริง
เหมือนกัน แสดงว่าคำศัพท์มีในพจนานุกรมแล้วออกจาก
ฟังก์ชันพร้อมกับระบุว่าไม่เพิ่มคำศัพท์
เนื่องจากมีคำศัพท์แล้ว
ไม่เหมือนกัน สร้างมัลติโหนดและชิงเกิลสตริงโดยใช้
ฟังก์ชัน BInsert(t,คำศัพท์ที่ซ้ำกัน,
ชิงเกิลสตริงหลังจากตัดส่วนที่ซ้ำ,
คำศัพท์ส่วนที่เหลือหลังจาก
ตัดส่วนที่ซ้ำ)
ออกจากฟังก์ชัน

3. ให้ทำงานตามข้อ 2 จนกว่าคำศัพท์จะหมดหรือเพิ่ม
คำศัพท์ได้สำเร็จ

1.3 <u>ฟังก์ชัน SetList</u>

เป็นฟังก์ชันที่ใช้สำหรับค้นหาชุดของพยัญชนะหรือคำโดดที่สร้างจากโหนด r โดยเก็บค่าของพยัญชนะหรือคำโดดไว้ในอะเรย์

2. ตัวอย่างการเพิ่มคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

ถ้าต้องการเพิ่มคำว่า "ตา กลม" ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สืบค้นว่ามีคำศัพท์ "ตา" หรือไม่ถ้าไม่มี เพิ่มคำศัพท์ "ตา"
ขั้นตอนที่ 2 สืบค้นว่ามีคำศัพท์ "กลม" หรือไม่ถ้าไม่มี เพิ่มคำศัพท์ "กลม"
ขั้นตอนที่ 3 เพิ่มคำศัพท์ "ตา กลม"ในพจนานุกรมฯ

รายละเอียดวิธีการเพิ่มคำศัพท์แสดงตามตาราง 4.2 รูปที่ 4.4 และ 4.5 แสดงรายละเอียดการเก็บคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยตามโครงสร้างข้อมูลที่พัฒนาขึ้น

	อินพุท	การปฏิบัติการ	เอาท์พุท
1.	ตา BASE[1]	สร้างโหนดใหม่จากโหนดรากกับ "ต" โดยใช้สมการ BASE[1] + ต = 22	โหนดใหม่ 22
2.		ตรวจสอบว่าโหนดถัดไปถูกใช้งานหรือยัง CHECK[22] = 0 ปรากฏว่ามีค่าเป็น 0 แสดงว่ายังไม่ได้ใช้งาน	
3.		เก็บชิงเกิลสตริงในอะเรย์เทล พร้อมกับ ตำแหน่งของชิงเกิลสตริงในอะเรย์ BASE INSSTR(1,า#)	TAIL[1] = า TAIL[2] = # TAIL[3] = $ BASE[22]= -1 POS = 4
4.	กลม BASE[1]	สร้างโหนดใหม่จากโหนดรากกับ "ก" โดยใช้สมการ BASE[1] + ก = 2	โหนดใหม่ 2
5.		ตรวจสอบว่าโหนดถัดไปถูกใช้งานหรือยัง CHECK[2] = 0 ปรากฏว่ามีค่าเป็น 0 แสดงว่ายังไม่ได้ใช้งาน	
6.		เก็บชิงเกิลสตริงในอะเรย์เทล พร้อมกับ ตำแหน่งของชิงเกิลสตริงในอะเรย์ BASE INSSTR(1,ลม#$)	TAIL[4] = ล TAIL[5] = ม TAIL[6] = # TAIL[7] = $ BASE[2]= -4 POS = 8
7.	ตา กลม BASE[1]	สร้างโหนดใหม่จากโหนดรากกับ "ต" โดยใช้สมการ BASE[1] + ต = 22	โหนดใหม่ 22

ตารางที่ 4.2 แสดงขั้นตอนการเพิ่มคำศัพท์

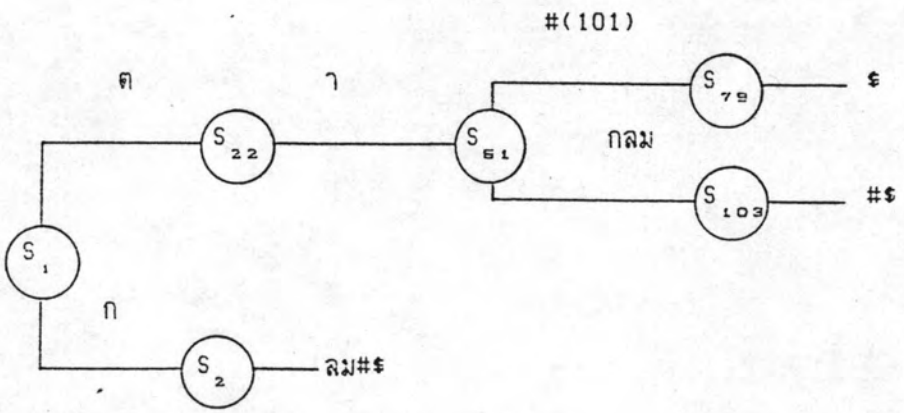
	อินพุท	การปฏิบัติการ	เอาท์พุท
8.		ตรวจสอบว่าโหนดถัดไปถูกใช้งานหรือยัง CHECK[22] = 1 ปรากฏว่ามีค่าเป็น 1 แสดงว่าเป็นเส้นทางเดินจากโหนดรากที่สร้างด้วย "ต"	
9.		ตรวจสอบค่าที่เก็บใน BASE[22] ปรากฏว่าเป็นลบต้องดึงชิงเกิลสตริงจากอะเรย์เทล	ชิงเกิลสตริง "า#"
10.	า กลม# า#	ตรวจสอบว่าคำศัพท์อินพุทส่วนที่เหลือว่าเหมือนกับชิงเกิลสตริงหรือไม่ ปรากฏว่าเหมือนกัน 1 ตัว	1
14.		สร้างมัลติโหนดโดยใช้ส่วนของสตริงที่ซ้ำกันตามนี้ BASE[22] + า = 51	CHECK[51]=22
15.	า#	พิจารณาชิงเกิลสตริง โดยแทนพยัญชนะส่วนที่ซ้ำด้วย ?	?#
16.	?#	บันทึกคำศัพท์ในอะเรย์เทล โดยสร้างโหนดสำหรับเก็บตำแหน่งคำศัพท์ในจากสมการ BASE[22] + # = 79	BASE[79]=-1 CHECK[79]=51 TAIL[1] = ? TAIL[2] = # TAIL[3] = $
17.	"า กลม#"	พิจารณาชิงเกิลสตริง โดยแทนพยัญชนะส่วนที่ซ้ำด้วย ?	? กลม#
18.	? กลม	บันทึกคำศัพท์ในอะเรย์เทล โดยสร้างโหนดสำหรับเก็บตำแหน่งคำศัพท์ในจากสมการ BASE[22] + กลม = 103 เนื่องจากพบช่องว่างทำให้ทราบว่าต้องหาค่าของคำศัพท์ทั้งคำ	BASE[103]=-8 CHECK[103]=51 TAIL[8] = # TAIL[9] =$ POS = 10

ตารางที่ 4.2 แสดงขั้นตอนการเพิ่มคำศัพท์

ดรรชนี	ค่าของอะเรย์เบส	ค่าของอะเรย์เช็ค	ค่าของอะเรย์เก็บค่า
1	1	103	0
2	-4	1	102
32	1	1	0
51	1	22	0
79	-1	51	101
103	-8	51	0

??$ลม#$#$

รูปที่ 4.4 แสดงความสัมพันธ์ของอะเรย์เบส อะเรย์เช็ค อะเรย์เก็บค่า และเทล



รูปที่ 4.5 รูปแสดงลักษณะโครงสร้างพจนานุกรมอิเล็กทรอนิกส์

การพัฒนาโครงสร้างข้อมูลสำหรับจัดเก็บพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

พจนานุกรมอิเล็กทรอนิกส์ภาษาไทยได้รับการพัฒนาด้วยภาษาซี [15] [16] [17] และเครื่องมือซอฟท์แวร์ต่างๆ ของไมโครซอฟท์ซี ภายใต้การดำเนินการของระบบดอส วิทยานิพนธ์ครั้งนี้ได้ทำการสร้างพจนานุกรมโดยใช้โครงสร้าง 2 รูปแบบ ดังนี้

1. พจนานุกรมอิเล็กทรอนิกส์ภาษาไทยตามโครงสร้างข้อมูลแบบดับเบิลอะเรย์

โปรแกรมชุดนี้มีขนาด 80 กิโลไบท์ ประกอบด้วยโมดูล (module) 2 ส่วน คือ

1. ส่วนของการเพิ่มคำศัพท์

และ 2. การสืบค้นคำศัพท์

รายละเอียดของโมดูลต่างๆ มี ดังนี้

1.1 โมดูลการเพิ่มคำศัพท์

ในส่วนนี้ได้พัฒนาโปรแกรมโดยใช้ภาษาซี และนำเอาอัลกอริทึมที่นิยามไว้ในบทที่ 3 มาใช้ ส่วนอะเรย์เบส อะเรย์เช็ค และอะเรย์เทลจัดเก็บไว้เป็นแฟ้มข้อมูล 3 แฟ้ม

1.2 โมดูลการสืบค้นคำศัพท์

ส่วนของการสืบค้นคำก็เช่นกันใช้อัลกอริทึมที่นิยามไว้ในบทที่ 3 มาใช้ ส่วนอินพุทที่ต้องการสืบค้นเก็บในแฟ้มข้อมูล แต่เพื่อความสะดวกจึงยินยอมให้ใส่ทางแป้นพิมพ์ได้ด้วย ผลลัพธ์ที่ได้จะเป็นจำนวนคำศัพท์ที่ค้นพบ และจำนวนคำศัพท์ที่สืบค้นไม่ได้ พร้อมทั้งเวลาที่ใช้ในการสืบค้นเพื่อใช้สำหรับการเปรียบเทียบประสิทธิภาพการทำงาน

นอกจากโปรแกรมที่พัฒนาขึ้นจะมี 2 ส่วนนี้แล้วยังจัดทำโมดูลเพิ่มอีก 2 โมดูล คือ

1. โมดูลสำหรับลบคำศัพท์
2. โมดูลสำหรับกำจัดการ์เบจในเทล

ซึ่งโมดูลทั้ง 2 เป็นโมดูลที่พัฒนาขึ้นเพื่อเพิ่มประสิทธิภาพให้พจนานุกรมอิเล็กทรอนิกส์ภาษาไทย โดยลบคำศัพท์ที่ไม่ใช้แล้ว และลดขนาดของแฟ้มข้อมูลเทลโดยกำจัดการ์เบจออกจากเทล ซึ่งเกิดขึ้นขณะทำการเพิ่มคำศัพท์

2. <u>พจนานุกรมอิเล็กทรอนิกส์ภาษาไทยตามโครงสร้างข้อมูลที่ออกแบบ</u>

โปรแกรมชุดนี้มีขนาด 80 กิโลไบท์ ประกอบด้วยโมดูล (module) 2 ส่วน คือ

1. ส่วนของการเพิ่มคำศัพท์
2. การสืบค้นคำศัพท์

รายละเอียดของโมดูลต่างๆ มี ดังนี้

2.1 โมดูลการเพิ่มคำศัพท์

ในส่วนนี้ได้พัฒนาโปรแกรมโดยใช้ภาษาซี โดยนำโครงสร้างข้อมูลและอัลกอริทึมที่ได้ออกแบบตามที่ได้อธิบายในบทที่ 4 มาใช้ ส่วนอะเรย์เบส อะเรย์เช็คเทล และอะเรย์เก็บค่าจัดเก็บไว้เป็นแฟ้มข้อมูล 4 แฟ้ม

2.2 โมดูลการสืบค้นคำศัพท์

ส่วนของการสืบค้นคำได้นำอัลกอริทึมที่ออกแบบไว้ในบทที่ 4 มาใช้ อินพุทที่ต้องการสืบค้นเก็บในแฟ้มข้อมูล ผลลัพธ์ที่ได้จะเป็นจำนวนคำศัพท์ที่ค้นพบ และจำนวนคำศัพท์ที่สืบค้นไม่ได้

นอกจากโปรแกรมที่พัฒนาขึ้นจะมี 2 ส่วนนี้แล้วยังจัดทำโมดูลเพิ่มอีก 2 โมดูล คือ

1. โมดูลสำหรับลบคำศัพท์
2. โมดูลสำหรับกำจัดการ์เบจในเทล

ซึ่งโมดูลทั้ง 2 เป็นโมดูลที่พัฒนาขึ้นเพื่อเพิ่มประสิทธิภาพให้พจนานุกรมอิเล็กทรอนิกส์ภาษาไทย โดยลบคำศัพท์ที่ไม่ใช้แล้ว และลดขนาดของแฟ้มข้อมูลเทลโดยกำจัดการ์เบจออกจากเทล ซึ่งเกิดขึ้นขณะทำการเพิ่มคำศัพท์