

## บทที่ 5

### สรุป

#### 5.1 สรุปผล

จากการจัดเก็บพจนานุกรมทั้ง 2 แบบ สามารถเปรียบเทียบประสิทธิภาพในอัลกอริทึมการตัดคำตามตารางที่ 5.1

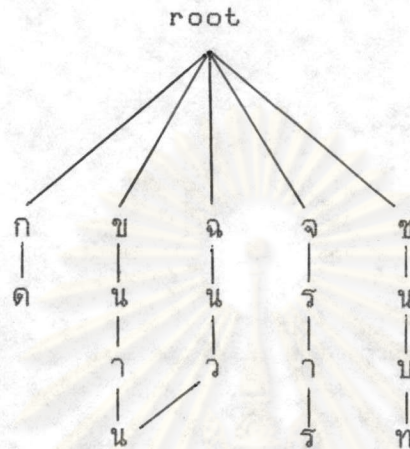
	Breath First Order	Depth First Order
จำนวนครั้งในการเข้าถึงพจนานุกรม	6198	5971
จำนวนครั้งในการเกิดการขาดเพจ(page fault)	2175	1690
เวลาที่ใช้ทั้งหมดในการตัดคำ(วินาที)	1031	628

ตารางที่ 5.1 แสดงการเปรียบเทียบประสิทธิภาพการจัดเก็บพจนานุกรม ตามหลักการเข้าถึงในทางกว้าง และการเข้าถึงในทางลึก

ผลที่ได้จากตารางที่ 5.1 เกิดจากการนำบทความจำนวน 4 หน้าเป็นข้อมูลสำหรับอัลกอริทึมการตัดคำ ซึ่งใช้พจนานุกรมขนาด 17000 คำ จากผลการเปรียบเทียบจะเห็นได้ว่าการจัดเก็บพจนานุกรมตามลำดับการเชื่อมโยงโหนดในทางลึก จำนวนครั้งของการเกิดการขาดเพจ(page fault) น้อยกว่า การจัดเก็บพจนานุกรมตามลำดับการเชื่อมโยงโหนดตามลำดับชั้น

ซึ่งสามารถอธิบายสาเหตุ ได้โดยอาศัยตัวอย่างดังต่อไปนี้

สมมติว่าพจนานุกรม ประกอบด้วยคำ 5 คำ ได้แก่ กต ขนาน จราจร ฉนวน ชนบท สามารถนำมาสร้างเป็นโครงสร้างแบบทราาย ได้ดังนี้



ในการจัดเก็บพจนานุกรมจะเก็บเรียงตามตัวอักษรดังนี้

การจัดเก็บพจนานุกรมตามลำดับการเยี่ยมชมโหนดตามลำดับชั้น(Breath First Order)

ก(1) ข(2) ฉ(3) จ(4) ช(5) ด(1) น(2) น(3) ร(4) น(5) ำ(2) ว(3)  
 ำ(4) บ(5) น(2,3) จ(4) ท(5) ร(4)

ตัวเลขที่อยู่ข้างหลังตัวอักษรเป็นตัวเลขที่บอกว่าตัวอักษรตัวดังกล่าวเป็นตัวอักษรของคำที่เท่าไร

การจัดเก็บพจนานุกรมตามลำดับการเยี่ยมชมโหนดในทางลึก(Depth First Order)

ก(1) ข(2) ฉ(3) จ(4) ช(5) ด(1) น(2,3) ำ(2) น(2) ว(3) น(3) ร(4)  
 จ(4) ำ(4) ร(4) ท(5) บ(5) น(5)

จะเห็นได้ว่าการจัดเก็บพจนานุกรมตามลำดับการเย็บมโหนดในทางลิกการกระจายของตัวอักษรของคำเดียวกัน จะกระจายน้อยกว่า การจัดเก็บพจนานุกรมตามลำดับการเย็บมโหนดตามลำดับชั้น ดังนั้นการจัดเก็บพจนานุกรมตามลำดับการเย็บมโหนดในทางลิกทำให้โอกาสที่คำๆเดียวกันจะถูกจัดอยู่ในเพจเดียวกัน มีมากกว่าการจัดเก็บพจนานุกรมตามลำดับการเย็บมโหนดตามลำดับชั้น ทำให้ในระหว่างการประมวลผลการตัดคำเกิดการขาดเพจ (page fault) น้อยกว่า

และจากการทดสอบอัลกอริทึมตัดคำทั้ง 2 แบบนี้มีข้อผิดพลาดในการตัดคำที่มีเกิดจาก

1. การที่ข้อความที่นำมาตัดคำเขียนผิด
2. คำที่มีอยู่ในข้อความที่ต้องการตัดคำ ไม่มีอยู่ในพจนานุกรม ซึ่งอาจแก้ไขได้โดยสร้างพจนานุกรมที่สามารถเพิ่มคำในระหว่างการทำงานได้

## 5.2 ข้อเสนอแนะ

5.2.1 อัลกอริทึมที่อยู่ในวิทยานิพนธ์นี้จะทำงานกับพจนานุกรมซึ่งเป็นทราาย 1 ต้น หมายความว่า จะต้องสร้างพจนานุกรม จากคำศัพท์ 1 ชุด เป็นทราาย 1 ต้นสำหรับให้อัลกอริทึมใช้การทำงานเช่นนี้ไม่ค่อยสะดวกนักเนื่องจากการเพิ่มเติมคำศัพท์จะต้อง Compile พจนานุกรมใหม่ ทางแก้ไขทำได้โดยให้อัลกอริทึมแบ่งคำทำงานกับทราาย 2 ต้น แต่มองเป็นพจนานุกรมเพียงอันเดียว โดย

ทราายต้นที่ 1 เป็นพจนานุกรมที่มีการ ทำ Suffix compress

ทราายต้นที่ 2 เป็นพจนานุกรมที่ไม่มีการ ทำ Suffix compress อนุญาตให้เพิ่มคำศัพท์ เข้าไปได้ในระหว่างทำงาน

การเพิ่มทราายที่ 2 นี้คงจะมีผลต่อความเร็วอยู่บ้าง

5.2.2 เพื่อให้การค้นหาคำศัพท์ทำได้เร็วขึ้นอาจทำได้โดยการนำคำศัพท์ที่ใช้บ่อยมาสร้างเป็นตารางแสดงความสัมพันธ์ระหว่างคำศัพท์และตำแหน่งของคำศัพท์นั้นๆ โดยอาจใช้หลักการของ Cache เข้ามาช่วย การค้นหาทำโดยเริ่มต้นที่ตารางก่อน