

การสร้างต้นไม้ตัดสินใจเพื่อค้นหาชื่อจตุลินทรีย์



กิริติ เต็มชำนาญ

งานค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีคอมพิวเตอร์และการสื่อสาร คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

พ.ศ. 2555

Decision Tree Construction for Microorganism Names Search



Keerati Temchamnan

**An Independent Study Submitted in Partial Fulfillment of the Requirements for
the Degree of Master of Science (Computer and Communication Technology)**

Department of Computer and Communication Technology

Faculty of Engineering, Dhurakij Pundit University

2012

กิตติกรรมประกาศ

งานค้นคว้าอิสระเล่มนี้เสร็จสมบูรณ์ ล่วงได้ด้วยดี ด้วยความกรุณาและความเอาใจใส่ เป็นอย่างดี จากอาจารย์ที่ปรึกษา อาจารย์ ดร.เนืองวงศ์ ทวยเจริญ ที่ได้สละเวลาให้คำปรึกษา คำแนะนำ ตลอดจนการตรวจแก้ไขงานค้นคว้าอิสระเล่มนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณเป็นอย่าง ยิ่ง

ขอกราบขอบพระคุณ อาจารย์ ดร.วรพล พงษ์เพชร ผู้อำนวยการหลักสูตรเทคโนโลยี คอมพิวเตอร์และการสื่อสาร ที่กรุณาให้คำแนะนำเพิ่มเติมในการทำงานค้นคว้าอิสระเล่มนี้

ขอกราบขอบพระคุณ อาจารย์ทุกท่าน ที่ได้ประสิทธิประสาทวิชาความรู้ต่างๆ

ขอกราบขอบพระคุณ มหาวิทยาลัยธุรกิจบัณฑิตย์ ที่ให้ทุนการศึกษาในการเรียนระดับปริญญาโทครั้งนี้

ขอกราบขอบพระคุณ ครอบครัว ที่เป็นกำลังใจและช่วยแนะนำสิ่งต่างๆ ตลอดมา

สุดท้ายนี้ ผู้วิจัยขออุทิศคุณงามความดีให้แก่ทุกท่านที่กล่าวมาข้างต้นที่มีส่วนในการทำงานค้นคว้าอิสระครั้งนี้หากมีความผิดพลาดหรือข้อบกพร่องประการใด ผู้วิจัยขออภัยเป็นอย่าง สูง มา ณ ที่นี้

กิริติ เต็มชำนานู

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ฉ
บทคัดย่อภาษาอังกฤษ	ง
กิตติกรรมประกาศ.....	จ
สารบัญตาราง	ช
สารบัญภาพ	ฉ
บทที่	
1. บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของการวิจัย.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
2. แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ	4
2.2 DOM Tree (Document Object Model Tree)	9
2.3 XML (Extensible Markup Language).....	10
2.4 AJAX (Asynchronous JavaScript And XML)	11
2.5 การแะผ่านต้นไม้ (Tree Traversal).....	13
2.6 การค้นหาในแนวลึก (Depth-first Search).....	23
2.7 การค้นหาในแนวกว้าง (Breadth-first Search).....	25
2.8 ต้นไม้ตัดสินใจ (Decision Tree)	26
2.9 การตั้งชื่อจูลินทรีย์.....	28
2.10 การค้นหาแบบต่างๆ	28
2.11 PHP (PHP:Hypertext Preprocessor).....	29
2.12 งานวิจัยที่เกี่ยวข้อง.....	30
3. ระเบียบวิธีวิจัย	33
3.1 ขั้นตอนการดำเนินการวิจัย	33
3.2 อุปกรณ์และเครื่องมือที่ใช้ในการวิจัย	33

สารบัญ (ต่อ)

บทที่	หน้า
3.3 ระยะเวลาในการดำเนินการวิจัย.....	34
3.4 สรุป.....	34
4. ผลการวิเคราะห์และออกแบบระบบ	36
4.1 ภาพรวมของระบบ	36
4.2 แนวคิดในการค้นหาข้อจุกจิก.....	37
4.3 การออกแบบระบบ	41
5. ผลการจัดทำและการทดสอบระบบ	45
5.1 ตารางข้อจุกจิก.....	45
5.2 ผลการทดสอบระบบ	47
5.3 วิเคราะห์ผลการทดสอบ	47
5.4 การเปรียบเทียบประสิทธิภาพของต้นไม้ตัดสินใจ.....	56
6. สรุปผลการวิจัยและข้อเสนอแนะ	58
6.1 สรุปผลการวิจัย.....	58
6.2 ข้อเสนอแนะ.....	58
บรรณานุกรม	59
ประวัติผู้เขียน	63

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงข้อมูลผู้ขออนุมัติสินเชื่อ.....	26
3.1 ระยะเวลาในการดำเนินการวิจัย.....	34
5.1 ชื่อจุลินทรีย์ที่ใช้ในการทดสอบ.....	45
5.2 ตารางเปรียบเทียบกับการค้นหาประสิทธิภาพแบบอื่น.....	57

สารบัญภาพ

ภาพที่	หน้า
2.1 เว็บไซต์เพื่อการสั่งซื้อจูลินทรีย์.....	5
2.2 หน้าจอการค้นหาจูลินทรีย์.....	6
2.3 หน้าจอแสดงการพิมพ์ชื่อจูลินทรีย์ที่ต้องการค้นหา.....	7
2.4 หน้าจอแสดงผลการค้นหาจูลินทรีย์.....	8
2.5 โครงสร้างต้นไม้สำหรับเอกสาร article.xml	9
2.6 เว็บแอปพลิเคชันซึ่งเปิดใช้งาน AJAX ได้ต่อกับ Server	12
2.7 ต้นไม้ที่ใช้ในการแฉผ่านแบบ Preorder.....	13
2.8 โหนดทางด้านซ้ายของรูทโหนด P (แบบ Preorder).....	14
2.9 ลำดับชั้นของโหนด S (แบบ Preorder).....	14
2.10 ลำดับชั้นของโหนด T (แบบ Preorder).....	15
2.11 โหนดทางด้านขวาของรูทโหนด P (แบบ Preorder).....	15
2.12 ลำดับชั้นของโหนด U (แบบ Preorder)	16
2.13 ลำดับชั้นของโหนด W (แบบ Preorder)	16
2.14 ลำดับชั้นของโหนด V (แบบ Preorder)	16
2.15 ต้นไม้ที่ใช้ในการแฉผ่านแบบ Inorder.....	17
2.16 โหนดด้านซ้ายมือของรูทโหนด P (แบบ Inorder).....	17
2.17 ลำดับชั้นของโหนด S (แบบ Inorder)	18
2.18 ลำดับชั้นของโหนด T (แบบ Inorder).....	18
2.19 โหนดด้านขวาของรูทโหนด P (แบบ Inorder)	19
2.20 ลำดับชั้นของโหนด U (แบบ Inorder)	19
2.21 ลำดับชั้นของโหนด W (แบบ Inorder)	19
2.22 ลำดับชั้นของโหนด V (แบบ Inorder)	20
2.23 ต้นไม้ที่ใช้ในการแฉผ่านแบบ Postorder	20
2.24 โหนดทางด้านซ้ายของรูทโหนด P (แบบ Postorder).....	21
2.25 ลำดับชั้นของโหนด S (แบบ Postorder)	21
2.26 ลำดับชั้นของโหนด T (แบบ Postorder).....	22
2.27 โหนดด้านขวาของรูทโหนด P (แบบ Postorder).....	22

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2.28 ลำดับชั้นของโหนด U (แบบ Postorder).....	22
2.29 ลำดับชั้นของโหนด W (แบบ Postorder).....	23
2.30 ลำดับชั้นของโหนด V (แบบ Postorder).....	23
2.31 รูปแบบการค้นหาในแนวลึก	24
2.32 รูปแบบการค้นหาในแนวกว้าง	25
2.33 ต้นไม้ตัดสินใจสำหรับข้อมูลในตารางที่ 2.1	27
2.34 กระบวนการการทำงานระหว่าง ลูกค้า เซิร์ฟเวอร์และ โมดูล PHP	30
4.1 ภาพรวมของเว็บไซต์ค้นหาเชื้อจุลินทรีย์	36
4.2 Dom Tree ของจุลินทรีย์ <i>Candida sorbosivorans</i>	37
4.3 Dom Tree ระดับที่ 1 ของจุลินทรีย์ <i>Candida sorbosivorans</i>	38
4.4 Dom Tree ระดับที่ 2 ของจุลินทรีย์ <i>Candida sorbosivorans</i>	38
4.5 Dom Tree ระดับที่ 3 ของจุลินทรีย์ <i>Candida sorbosivorans</i>	38
4.6 Dom Tree ระดับที่ 4 ของจุลินทรีย์ <i>Candida sorbosivorans</i>	39
4.7 Dom Tree ระดับที่ 5 ของจุลินทรีย์ <i>Candida sorbosivorans</i>	39
4.8 Dom Tree ระดับที่ 6 ของจุลินทรีย์ <i>Candida sorbosivorans</i>	39
4.9 Dom Tree ระดับที่ 7 ของจุลินทรีย์ <i>Candida sorbosivorans</i>	40
4.10 Dom Tree ระดับที่ 8 ของจุลินทรีย์ <i>Candida sorbosivorans</i>	40
4.11 Flow chart แสดงขั้นตอนการทำงานของโปรแกรม	41
4.12 หน้าจอการหาความแตกต่างของจำนวนตัวอักษรของสกุลจุลินทรีย์	42
4.13 หน้าจอแสดงผลการหาความแตกต่างของจำนวนตัวอักษรของสกุลจุลินทรีย์	42
4.14 หน้าจอการหาความแตกต่างของจำนวนตัวอักษรของชนิดในสกุลจุลินทรีย์.....	43
4.15 หน้าจอแสดงผลการหาความแตกต่างของจำนวนตัวอักษรของชนิดในสกุลจุลินทรีย์... ..	43
5.1 ผลการทดสอบการหาความลึกของต้นไม้.....	47
5.2 Dom Tree ของการค้นหาจุลินทรีย์ <i>Micrococcus luteus</i>	48
5.3 Dom Tree ระดับที่ 1 ของจุลินทรีย์ <i>Micrococcus luteus</i>	48
5.4 Dom Tree ระดับที่ 2 ของจุลินทรีย์ <i>Micrococcus luteus</i>	49
5.5 Dom Tree ระดับที่ 3 ของจุลินทรีย์ <i>Micrococcus luteus</i>	49

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
5.6 Dom Tree ของการค้นหาจุลินทรีย์ Trichosporiella sp.	50
5.7 Dom Tree ระดับที่ 1 ของจุลินทรีย์ Trichosporiella sp.	50
5.8 Dom Tree ระดับที่ 2 ของจุลินทรีย์ Trichosporiella sp.	51
5.9 Dom Tree ระดับที่ 3 ของจุลินทรีย์ Trichosporiella sp.	51
5.10 Dom Tree ระดับที่ 4 ของจุลินทรีย์ Trichosporiella sp.	52
5.11 Dom Tree ของจุลินทรีย์ Candida sorbosivorans.....	53
5.12 Dom Tree ระดับที่ 1 ของจุลินทรีย์ Candida sorbosivorans.....	54
5.13 Dom Tree ระดับที่ 2 ของจุลินทรีย์ Candida sorbosivorans.....	54
5.14 Dom Tree ระดับที่ 3 ของจุลินทรีย์ Candida sorbosivorans.....	54
5.15 Dom Tree ระดับที่ 4 ของจุลินทรีย์ Candida sorbosivorans.....	55
5.16 Dom Tree ระดับที่ 5 ของจุลินทรีย์ Candida sorbosivorans.....	55
5.17 Dom Tree ระดับที่ 6 ของจุลินทรีย์ Candida sorbosivorans.....	55
5.18 Dom Tree ระดับที่ 7 ของจุลินทรีย์ Candida sorbosivorans.....	56
5.19 Dom Tree ระดับที่ 8 ของจุลินทรีย์ Candida sorbosivorans.....	56

หัวข้องานค้นคว้าอิสระ
ชื่อผู้เขียน
อาจารย์ที่ปรึกษา
สาขาวิชา
ปีการศึกษา

การสร้างต้นไม้มัดคตินใจเพื่อค้นหาชื่อจุลินทรีย์
กิริติ เต็มชำนาญ
อาจารย์ ดร. เนื่องวงศ์ ทวยเจริญ
เทคโนโลยีคอมพิวเตอร์และการสื่อสาร
2554

บทคัดย่อ

จุลินทรีย์เป็นองค์ประกอบที่สำคัญในการนำมาใช้ในการวิจัย และจากการที่จุลินทรีย์มีหลากหลายสายพันธุ์และมีเป็นจำนวนมากในประเทศไทย ศูนย์พันธุวิศวกรรมและเทคโนโลยีแห่งชาติ (ไบโอเทค) จึงได้มีการจัดตั้ง ธนาคารจุลินทรีย์ เพื่อทำการเก็บรวบรวมจุลินทรีย์ตามสกุลและชนิดต่างๆ รวมทั้งนำไปทำการวิจัย และมีการจำหน่ายให้กับองค์กรหรือบุคคลภายนอกที่จะนำจุลินทรีย์ไปใช้ประโยชน์ ไบโอเทคจึงมีการจัดทำเว็บไซต์เพื่ออำนวยความสะดวกให้กับผู้ที่สนใจจะสั่งซื้อจุลินทรีย์เพื่อให้เกิดความสะดวกรวดเร็ว แต่อย่างไรก็ตาม การค้นหาจุลินทรีย์ยังเกิดความล่าช้า เนื่องจากการค้นหาที่ผู้ใช้ระบบต้องพิมพ์ชื่อเชื้อจุลินทรีย์เอง และหากมีการสะกดที่ผิดพลาดระบบจะไม่แสดงชื่อของจุลินทรีย์ ทำให้ผู้ใช้ต้องพิมพ์ใหม่อีกครั้ง

งานค้นคว้าอิสระเล่มนี้เสนอแนวคิดในการเพิ่มความรวดเร็วในการค้นหาจุลินทรีย์โดยใช้เทคนิคต้นไม้มัดคตินใจในการค้นหาชื่อจุลินทรีย์ ส่งผลให้การค้นหาเกิดความสะดวกรวดเร็ว รวมไปถึงลดความเสี่ยงในการค้นหาผิดพลาดได้ ทำให้การสั่งซื้อจุลินทรีย์นั้นเกิดความสะดวกและรวดเร็วเช่นกัน

Independent Study Title	Decision Tree Construction for Microorganism Names Search
Author	Keerati Temchamnan
Independent Study Advisor	Dr. Nuengwong Tuaycharoen
Department	Computer and Communication Technology
Academic Year	2011

ABSTRACT

Microorganism is an important component in various research fields in Thailand in which there are various genus and species of Microorganism. The National Center for Genetic Engineering and Biotechnology (BIOTEC) had established a microorganism bank for various genus and species collections and conducts a research. Additionally, BIOTEC sells microorganism to organizations and general public, who acquire microorganism. BIOTEC hosts a website to facilitate those who are interested to order microorganism more conveniently via the Internet. However, searching for a microorganism name still delays the ordering process because user must type the exact name of the microorganism by himself/herself. If the user misspells the name, the system would not return any results, and the user must retype the name again.

This independent study presents the concept to make a microorganism names searching process faster by using a decision tree. The user is not required to type all of the letters in a microorganism name when search. As a result, the proposed method can accelerate searching process and reduce risks in misspelling searches. So, the microorganism ordering process will be more convenient and faster.

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

จุลินทรีย์เป็นองค์ประกอบที่สำคัญและถูกนำมาใช้อย่างแพร่หลายในการวิจัย โดยการวิจัยที่มีจุลินทรีย์เข้ามาเกี่ยวข้อง เช่น ด้านอาหาร ด้านการเกษตร ด้านอุตสาหกรรม ฯลฯ ประโยชน์ของจุลินทรีย์ที่ถูกนำมาใช้ในการเกษตร เช่น ย่อยสลายสารอินทรีย์และเพิ่มความสมบูรณ์ให้แก่ดิน เป็นต้น ในด้านอุตสาหกรรม เช่น บำบัดน้ำเสีย เป็นต้น ในด้านอาหาร เช่น ผลิตโยเกิร์ต ไวน์ ข้าวสาลี เป็นต้น และเนื่องจากจุลินทรีย์ในประเทศไทยมีเป็นจำนวนมากหลากหลายสกุลและชนิด ทำให้ยากต่อการค้นหาและนำมาใช้ในการวิจัย

ศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ (ไบโอเทค) จัดตั้งขึ้นเมื่อปี พ.ศ. 2526 เป็นหน่วยงานในการกำกับดูแลของสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.) ไบโอเทคเป็นหน่วยงานที่เป็นแกนหลักในการสนับสนุนและนำวิทยาศาสตร์และเทคโนโลยีมาใช้ในการพัฒนาทางด้านอุตสาหกรรม เกษตรกรรม และสิ่งแวดล้อม ไบโอเทคมีหน่วยวิจัยซึ่งแต่ละหน่วยวิจัยแบ่งเป็นห้องปฏิบัติการต่างๆ เพื่อทำการวิจัยให้กับภาครัฐบาล ภาคเอกชนและหน่วยงานภายนอก ที่สนใจทำการวิจัยร่วมกับทางไบโอเทค

จากสภาพภูมิศาสตร์ที่หลากหลายของประเทศไทย ทำให้เป็นแหล่งรวมของจุลินทรีย์ประเภทต่างๆ มากกว่า 200,000 ชนิด และจากการวิจัยพบว่าจุลินทรีย์สามารถนำไปใช้ประโยชน์ได้หลากหลายด้าน ไบโอเทคจึงเล็งเห็นถึงความสำคัญของจุลินทรีย์ว่าควรเก็บรักษาและใช้ประโยชน์ในอนาคตได้ต่อไป

ห้องปฏิบัติการเก็บรักษาสายพันธุ์จุลินทรีย์ สังกัดหน่วยวิจัยเทคโนโลยีชีวภาพ ซึ่งเป็นหน่วยวิจัยของไบโอเทคจึงมีการจัดตั้งธนาคารจุลินทรีย์ขึ้นในปีพ.ศ. 2540 เพื่อเป็นสถานที่เก็บรวบรวมจุลินทรีย์ตามสกุลและชนิดต่างๆ รวมไปถึงเป็นประโยชน์ในการทดลองสำหรับนักวิจัยภายในหน่วยงาน หลังจากนั้นจึงได้ขยายออกเป็นวงกว้างด้วยการรับฝากจุลินทรีย์จากภายนอกเพื่อนำมาเก็บรักษาไว้ในธนาคารจุลินทรีย์และจำหน่ายให้กับบุคคลภายนอกที่สนใจ เช่น บริษัทเอกชน นักศึกษา นักวิจัยภายนอกหน่วยงาน หน่วยงานภาครัฐบาลอื่นๆ เป็นต้น

สำหรับปัญหาที่พบในช่วงแรกของการจำหน่ายจูลินทรีย์ให้กับบุคคลภายนอก เช่น ผู้ซื้อไม่ทราบสกุลและชนิดของจูลินทรีย์ที่ต้องการซื้อ ทำให้ยากต่อการค้นหา ผู้ซื้อไม่ทราบว่าไปโอเทคมีจูลินทรีย์ประเภทใดจำหน่ายบ้าง ทำให้เกิดความล่าช้าขึ้นได้ เป็นต้น ปัญหาเหล่านี้ส่งผลให้เกิดความล่าช้าในการสั่งซื้อจูลินทรีย์ด้วยเช่นกัน

ไปโอเทคจึงมีการจัดทำเว็บไซต์เพื่ออำนวยความสะดวกให้กับบุคคลภายนอก เพื่อให้สามารถสั่งซื้อจูลินทรีย์แบบออนไลน์ได้ทันที โดยผู้ใช้งานสามารถค้นหาจูลินทรีย์ที่ตนเองต้องการว่ามีจำหน่ายที่ไปโอเทคหรือไม่ รวมไปถึงเป็นจูลินทรีย์ประเภทใดชนิดใด หลังจากการค้นหาผู้ใช้งานสามารถเลือกจำนวนของจูลินทรีย์ที่ต้องการและสั่งซื้อจูลินทรีย์ได้อย่างสะดวกและรวดเร็ว

อย่างไรก็ตาม จูลินทรีย์ประเภทต่างๆ มีหลากหลายสกุลและชนิด รวมทั้งแต่ละชนิดและสกุลนั้น มีชื่อทางวิทยาศาสตร์ที่เฉพาะและมีความยาวของตัวอักษรอยู่มาก จึงเกิดปัญหาในการค้นหา ดังนี้

1. หากผู้ใช้งานพิมพ์ชื่อจูลินทรีย์หรือตัวอักษรผิดพลาด ระบบจะไม่แสดงข้อมูลใดๆ ที่แสดงให้เห็นว่าผู้ใช้งานพิมพ์ตัวอักษรผิด

2. เกิดความล่าช้า เนื่องจากผู้ใช้งานต้องพิมพ์ชื่อจูลินทรีย์ให้ครบถ้วนก่อนทำการค้นหา ซึ่งจะทำให้เสียเวลาและล่าช้าต่อไป

จากสองปัญหานี้ทำให้ผู้ใช้งานต้องทำการค้นหาใหม่อีกครั้ง ส่งผลให้เกิดความล่าช้าในการสั่งซื้อจูลินทรีย์ตามมาเช่นกัน

ผู้วิจัยจึงสนใจในการพัฒนาแนวคิดการค้นหาชื่อจูลินทรีย์ โดยใช้ต้นไม้ตัดสินใจในการค้นหา ซึ่งต้นไม้ตัดสินใจทำให้เข้าใจได้ง่าย รวมทั้งเป็นเทคนิคที่ให้ผลเร็วและให้ความแม่นยำสูง โดยการค้นหาโดยต้นไม้ตัดสินใจนี้จะช่วยอำนวยความสะดวกและรวดเร็วให้กับผู้ใช้งานได้

1.2 วัตถุประสงค์ของการวิจัย

วัตถุประสงค์ของการวิจัย มีดังต่อไปนี้

1. เพื่อนำเสนอแนวคิดในการใช้ต้นไม้ตัดสินใจที่สามารถช่วยให้การค้นหาชื่อจูลินทรีย์ทำได้อย่างรวดเร็ว

2. เพื่อค้นหาขอบเขตความลึกของต้นไม้ตัดสินใจสำหรับการค้นหาชื่อจูลินทรีย์ ซึ่งเป็นตัวบ่งชี้ประสิทธิภาพทางเวลาในการค้นหาชื่อจูลินทรีย์

1.3 ขอบเขตของการวิจัย

ขอบเขตของการวิจัย มีดังต่อไปนี้

1. ศึกษาจำนวนสกุลและชนิดของจุลินทรีย์ในศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ ที่ใช้จำหน่ายให้กับบุคคลภายนอก
2. ค้นหาความแตกต่างของจำนวนตัวอักษรของสกุลและชนิดของชื่อจุลินทรีย์
3. ออกแบบต้นไม้ตัดสินใจในการค้นหาชื่อจุลินทรีย์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้ทราบถึงความสูงของต้นไม้ตัดสินใจในการค้นหาชื่อจุลินทรีย์
2. สามารถกำหนดขอบเขตประสิทธิภาพทางเวลาในการค้นหาชื่อจุลินทรีย์ได้จากความสูงของต้นไม้ตัดสินใจ
3. การค้นหาชื่อจุลินทรีย์ มีความสะดวกและรวดเร็วมากขึ้น

บทที่ 2

แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

2.1 ศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ

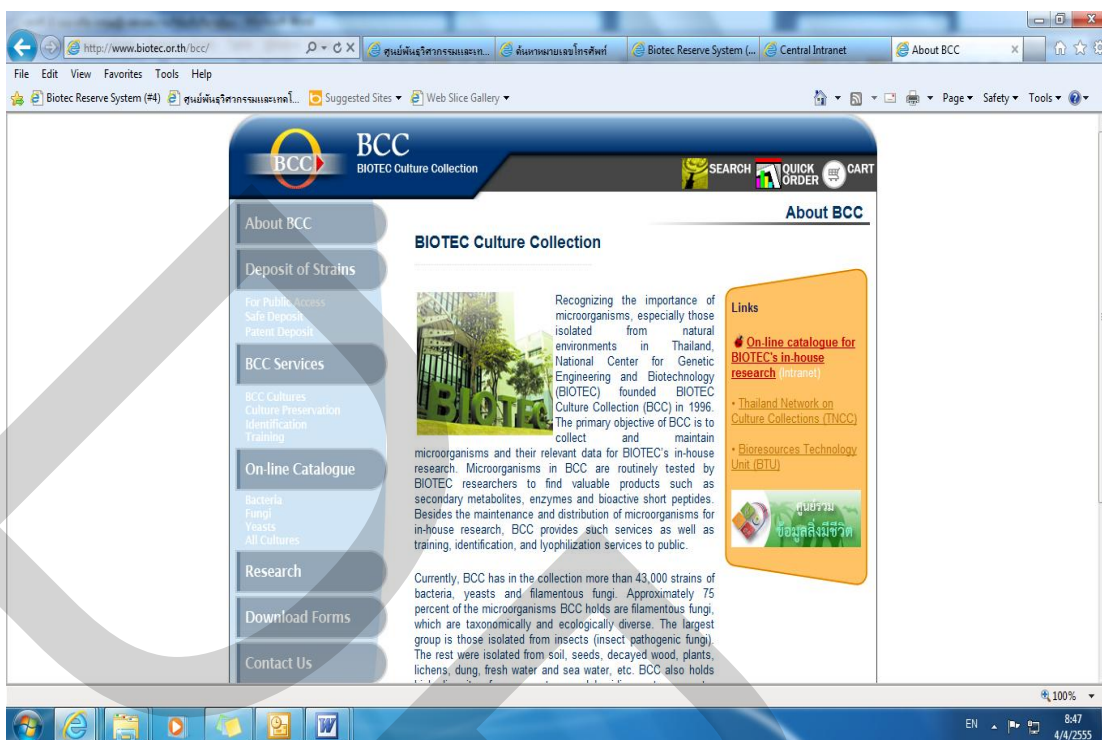
ศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ หรือ ไบโอเทค เป็นหน่วยงานในการกำกับดูแลของสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ หรือ สวทช. โดยไบโอเทคเป็นหน่วยงานที่เป็นแกนหลักในการสนับสนุนและนำวิทยาศาสตร์และเทคโนโลยีมาใช้ในการพัฒนาทางด้านอุตสาหกรรม เกษตรกรรมและสิ่งแวดล้อม รวมถึงไบโอเทคมีหน่วยวิจัยภายในและหน่วยวิจัยเครือข่ายซึ่งตั้งอยู่ตามมหาวิทยาลัยต่างๆ เพื่อสร้างความเข้มแข็งด้านการวิจัยอีกด้วย

หน่วยวิจัยของไบโอเทคนั้น จะแบ่งเป็นห้องปฏิบัติการในด้านต่างๆ เพื่อทำการวิจัยให้กับภาครัฐบาล ภาคเอกชน และหน่วยงานภายนอก ที่สนใจจะทำการวิจัยร่วมกันกับทางไบโอเทค เช่น หน่วยวิจัยชีวภาพทางการเกษตร แบ่งเป็น ห้องปฏิบัติการไมโครอะเรย์ครบวงจร ห้องปฏิบัติการวิจัยด้านพืช เป็นต้น หรือ หน่วยวิจัยเทคโนโลยีชีวภาพ แบ่งเป็น ห้องปฏิบัติการนิเวศวิทยา ห้องปฏิบัติการเทคโนโลยีเอนไซม์ ห้องปฏิบัติการเก็บรักษาสายพันธุ์จุลินทรีย์ เป็นต้น

2.1.1 ธนาคารจุลินทรีย์

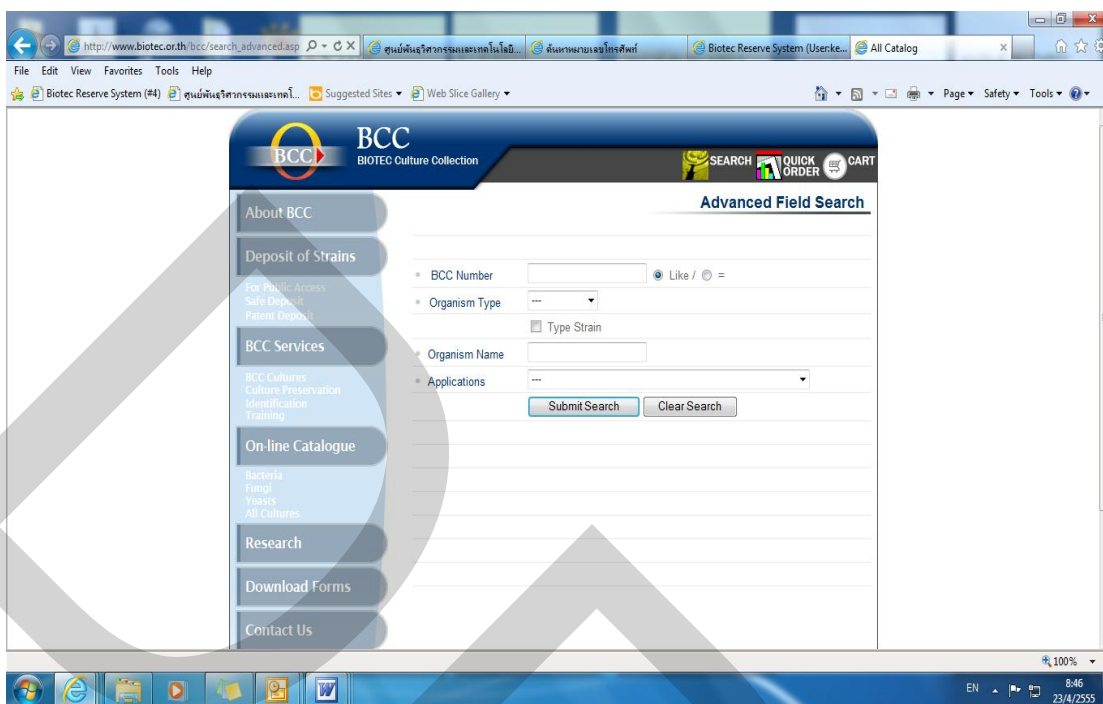
ห้องปฏิบัติการเก็บรักษาสายพันธุ์จุลินทรีย์ มีธนาคารจุลินทรีย์ เป็นแหล่งเก็บรวบรวมจุลินทรีย์จำพวก แบคทีเรีย รา และยีสต์ ที่มีในประเทศไทย เพื่อให้นักวิจัยภายในหน่วยงานได้นำไปทดลองทำการวิจัย รวมถึงจำหน่ายให้กับบุคคลภายนอกที่จะนำจุลินทรีย์ไปใช้ประโยชน์เช่นกัน

ไบโอเทคมีการจัดทำเว็บไซต์เพื่อให้บุคคลภายนอกทำการสั่งซื้อจุลินทรีย์ได้ทันที เพื่อให้เกิดความสะดวกและรวดเร็วในการสั่งซื้อจุลินทรีย์ แต่อย่างไรก็ตาม ในด้านการค้นหาชื่อจุลินทรีย์เพื่อทำการสั่งซื้อจากทางเว็บไซต์นั้น ผู้ใช้งานต้องพิมพ์ชื่อจุลินทรีย์ที่ต้องการค้นหาเมื่อผู้ใช้งานสะกดตัวอักษรผิด หรือสะกดไม่ครบ ระบบจะไม่แสดงชื่อจุลินทรีย์ และผู้ใช้งานต้องพิมพ์ใหม่อีกครั้ง ส่งผลให้เกิดความล่าช้าในการค้นหาและการสั่งซื้อจุลินทรีย์กับผู้ใช้งานได้



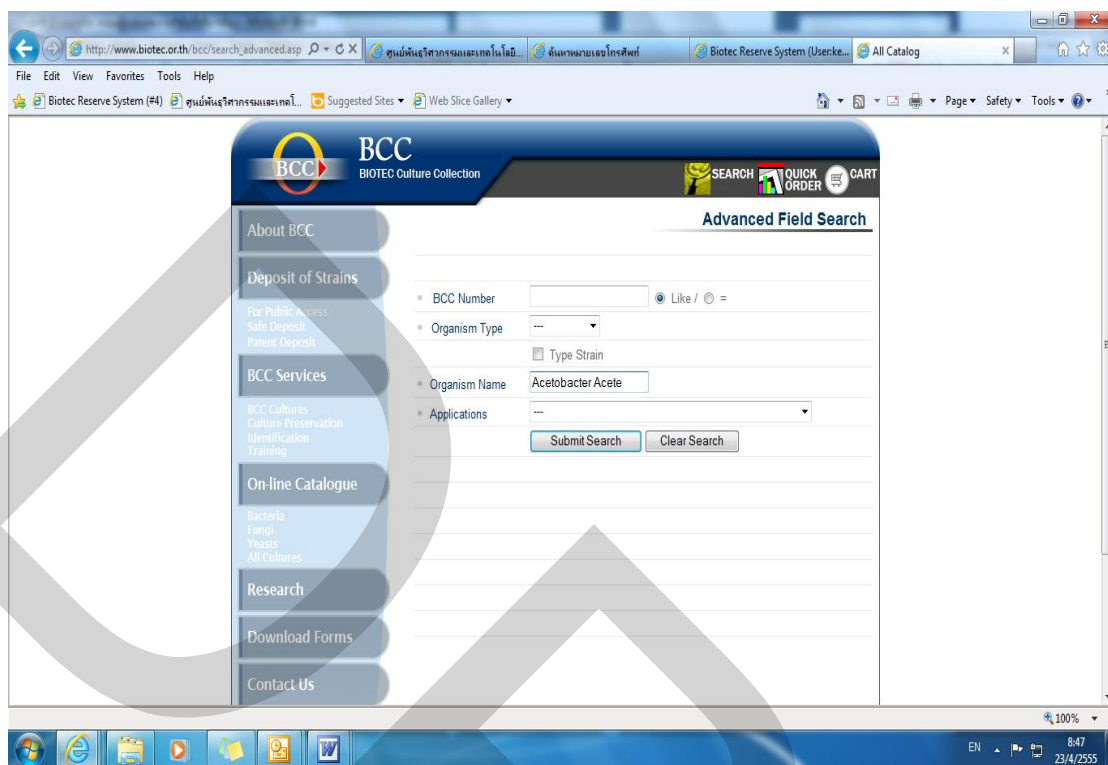
ภาพที่ 2.1 เว็บไซต์เพื่อการสั่งซื้อจุลินทรีย์

จากภาพที่ 2.1 คือหน้าจอของเว็บไซต์เพื่อการสั่งซื้อจุลินทรีย์ โดยผู้ใช้งานสามารถค้นหาได้จากสองทาง คือ เมนูด้านบนสุด คลิกที่คำว่า SEARCH หรือ เมนูด้านซ้าย ในส่วนของ Online Catalogue ซึ่งในส่วนนี้สามารถเลือกได้ว่าจะค้นหาจุลินทรีย์ประเภทใด แบ่งเป็น แบคทีเรีย และ ยีสต์



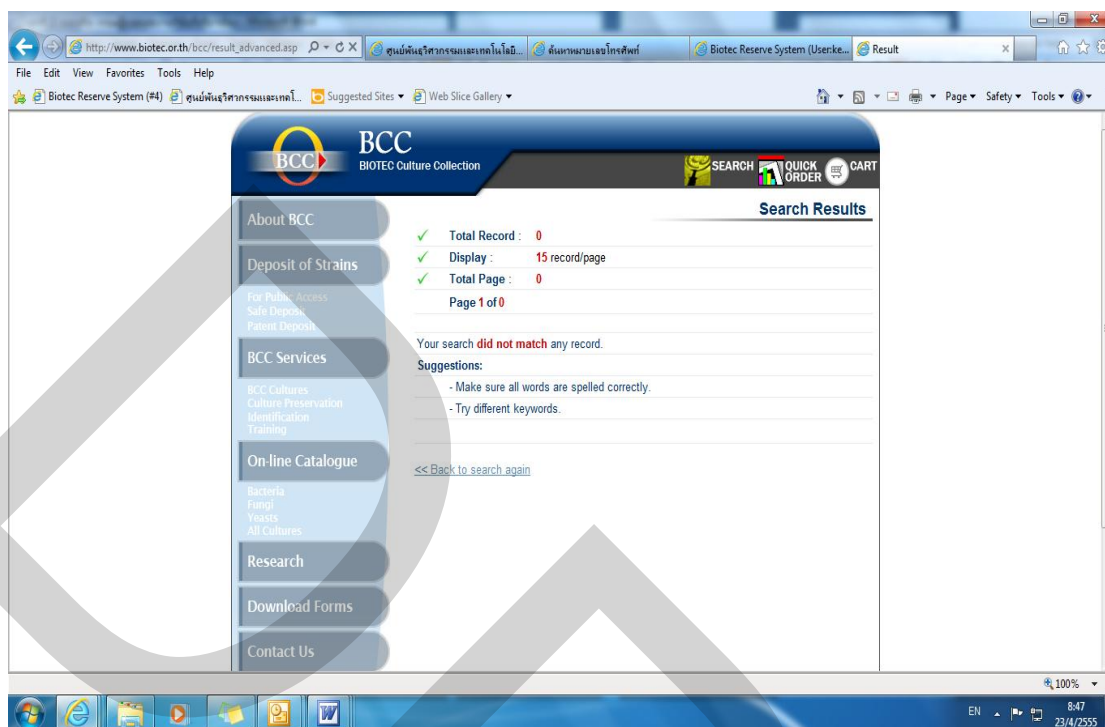
ภาพที่ 2.2 หน้าจอการค้นหาจุลินทรีย์

จากภาพที่ 2.2 หน้าจอการค้นหาจุลินทรีย์ ผู้ใช้งานสามารถค้นหาได้จากหมายเลขของจุลินทรีย์ หรือ ชื่อของจุลินทรีย์ หรือ การใช้งาน (Application) ของจุลินทรีย์ที่จะนำไปใช้งาน รวมไปถึงสามารถเลือกได้จากประเภทของจุลินทรีย์ได้เช่นกัน



ภาพที่ 2.3 หน้าจอแสดงการพิมพ์ชื่อจุลินทรีย์ที่ต้องการค้นหา

จากภาพที่ 2.3 ผู้ใช้งานพิมพ์ชื่อจุลินทรีย์ที่ต้องการค้นหา ลงในช่อง Organism Name หลังจากนั้นกดปุ่ม Submit Search เพื่อแสดงผลการค้นหาจุลินทรีย์

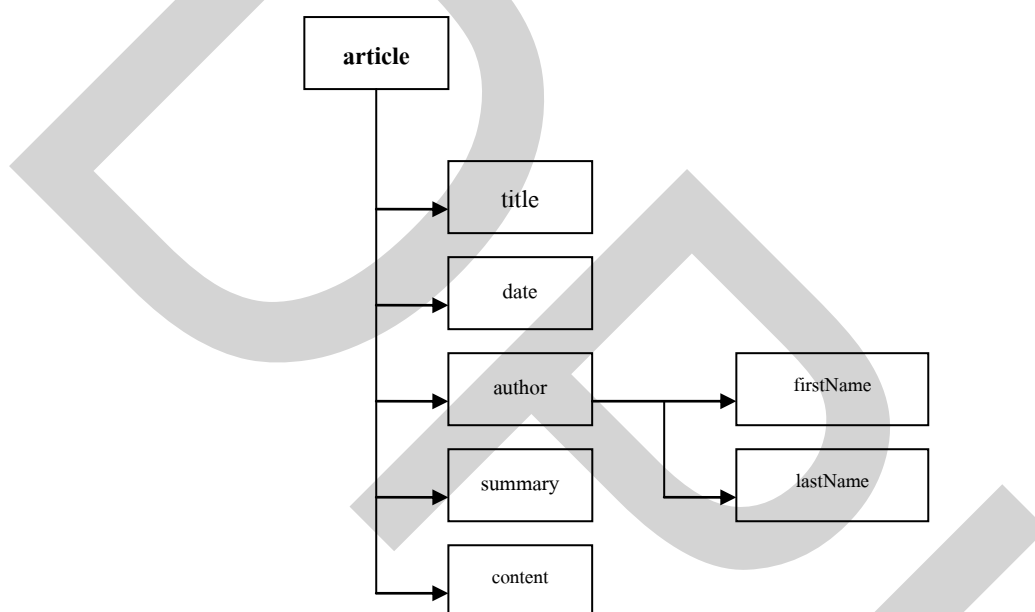


ภาพที่ 2.4 หน้าจอแสดงผลการค้นหาจุลินทรีย์

จากภาพที่ 2.4 หลังจากกดปุ่ม Submit Search ระบบจะแสดงผลการค้นหา ในกรณีนี้ระบบไม่พบชื่อจุลินทรีย์ที่ต้องการค้นหา เนื่องจาก อาจจะมีชื่อจุลินทรีย์ผิดพลาด หรือ ไม่มีชื่อจุลินทรีย์ที่ต้องการ

2.2 DOM Tree (Document Object Model Tree)

DOM Tree เป็นโครงสร้างลำดับชั้นของต้นไม้ จะมีรูทโหนดเพียงโหนดเดียวที่ประกอบด้วยโหนดอื่นๆ ในเอกสาร โดยแต่ละชื่อขององค์ประกอบจะแทนด้วยโหนด โหนดที่ประกอบด้วยโหนดอื่นๆ (โหนดลูก) เรียกว่า โหนดพ่อแม่ โหนดพ่อแม่สามารถมีโหนดลูกได้หลากหลาย แต่โหนดลูกสามารถมีโหนดพ่อแม่ได้เพียงโหนดเดียว ดังภาพ



ภาพที่ 2.5 โครงสร้างต้นไม้สำหรับเอกสาร article.xml

ที่มา : AJAX, Rich Internet Applications and Web Development for programmers books

จากภาพที่ 2.5 จะแสดงให้เห็นถึงโครงสร้างต้นไม้สำหรับองค์ประกอบรากของเอกสารชื่อ article.xml โครงสร้างลำดับชั้นของต้นไม้ นี้ เรียกว่า DOM Tree โดย DOM Tree ของเอกสาร article.xml มีรูทโหนด คือ article และมีโหนดลูก ได้แก่ title, date, author, summary และ content และจากภาพจะเห็นได้ว่าโหนด author เป็นโหนดพ่อแม่ มีโหนดลูกคือ firstName และ lastName ซึ่งโหนดพ่อแม่สามารถมีโหนดลูกได้มากมาย แต่โหนดลูกนั้น สามารถมีโหนดพ่อแม่ได้เพียงโหนดเดียว

ดังนั้น DOM Tree จะอธิบายความสัมพันธ์ระหว่างแต่ละองค์ประกอบในเอกสาร โดยจะผ่านความสัมพันธ์ระหว่าง พ่อแม่-ลูก ของโหนดที่มีความเกี่ยวข้องกัน

2.3 XML (Extensible Markup Language)

XML ถูกพัฒนาขึ้นในปี 1996 โดย W3C (The World Wide Web Consortium) ซึ่ง XML เป็นภาษามาร์กอัป XML อธิบายข้อมูลในแง่ของความหมายที่มนุษย์และคอมพิวเตอร์เข้าใจได้

เอกสาร XML คือ แฟ้มข้อความซึ่งมีนามสกุลลงท้ายด้วย .xml เอกสารนี้สามารถอ้างอิง DTD หรือ Schema ซึ่งเป็นส่วนกำหนดโครงสร้างของเอกสารที่อนุญาตให้ข้อมูลในเอกสารตรวจสอบได้โดย parser หมายถึง จะมีการตรวจสอบให้แน่ใจว่าไม่มีองค์ประกอบใดที่ขาดหายไป และองค์ประกอบนั้นเกิดขึ้นเหมาะสม ในส่วนนี้ทำให้ข้อมูล XML มีความน่าเชื่อถือมากกว่าข้อมูลที่จัดทำขึ้นด้วยข้อมูลอื่น XML สามารถใช้ในการสร้างภาษามาร์กอัปที่กำหนดเองได้เช่นกัน

ตัวอย่าง XML

```
<?xml version= "1.0"?>
<!DOCTYPE telegram SYSTEM "/xml-resources/dtds/telegram.dtd">
<telegram pri= "important">
  <to>Sarah Bellum</to>
  <from>Colonel Timeslip</from>
  <subject>Robot-sitting instructions</subject>
  <graphic fileref= "figs/me.eps"/>
  <message>Thanks for watching my robot pal
    <name>Zonky</name> while I'm away.
    He needs to be recharged <emphasis>twice a
    day</emphasis> and if he starts to get cranky,
    give him a quart of oil. I'll be back soon,
    after I've tracked down that evil
    mastermind <villain>Dr. Indigo Riceway</villain.>
  </message>
</telegram>
```

ที่มา : Learning XML books

สัญลักษณ์มาร์กอัพจะถูกอธิบายโดยเครื่องหมาย < > ซึ่ง <to> และ </villain> สัญลักษณ์ทั้งสองนี้ จะเรียกว่า แท็ก ข้อมูลหรือเนื้อหาต่างๆ จะอยู่ระหว่างแท็กเหล่านี้

ส่วนบนสุดของเอกสาร คือ การประกาศ XML จะช่วยในการระบุเวอร์ชันของ XML และชนิดของตัวอักษรที่เข้ารหัสได้ รวมถึงช่วยให้การประมวลผล XML เริ่มต้นในเอกสารอีกด้วย

หลังจากที่มีการประกาศประเภทของเอกสารที่มีการอ้างอิงเอกสารอธิบายไวยากรณ์ โดยตั้งอยู่บนระบบในไฟล์ /xml-resources/dtds/telegram.dtd. ซึ่งจะเรียกว่า การกำหนดประเภทเอกสาร (Document type definition (DTD)).

<!DOCTYPE...> เป็นตัวอย่างของประเภทของมาร์กอัพ ที่เรียกว่า การประกาศ โดยการประกาศจะใช้ในการจำกัดไวยากรณ์และประกาศส่วนของข้อความหรือแหล่งข้อมูลที่อยู่ในเอกสาร

<telegram> แท็กนี้เป็นจุดเริ่มต้นขององค์ประกอบ(element) สำหรับจุดสิ้นสุดขององค์ประกอบนี้จะอยู่ด้านล่างและจะแสดงโดยแท็ก </telegram> องค์ประกอบนี้จะประกอบด้วยเนื้อหาทั้งหมดของเอกสาร ดังนั้นจึงเรียกว่า องค์ประกอบเอกสาร ภายในองค์ประกอบเอกสารจะประกอบด้วยองค์ประกอบอื่นๆ ที่มีแท็กเริ่มต้นและแท็กสิ้นสุดตามรูปแบบคล้ายกัน

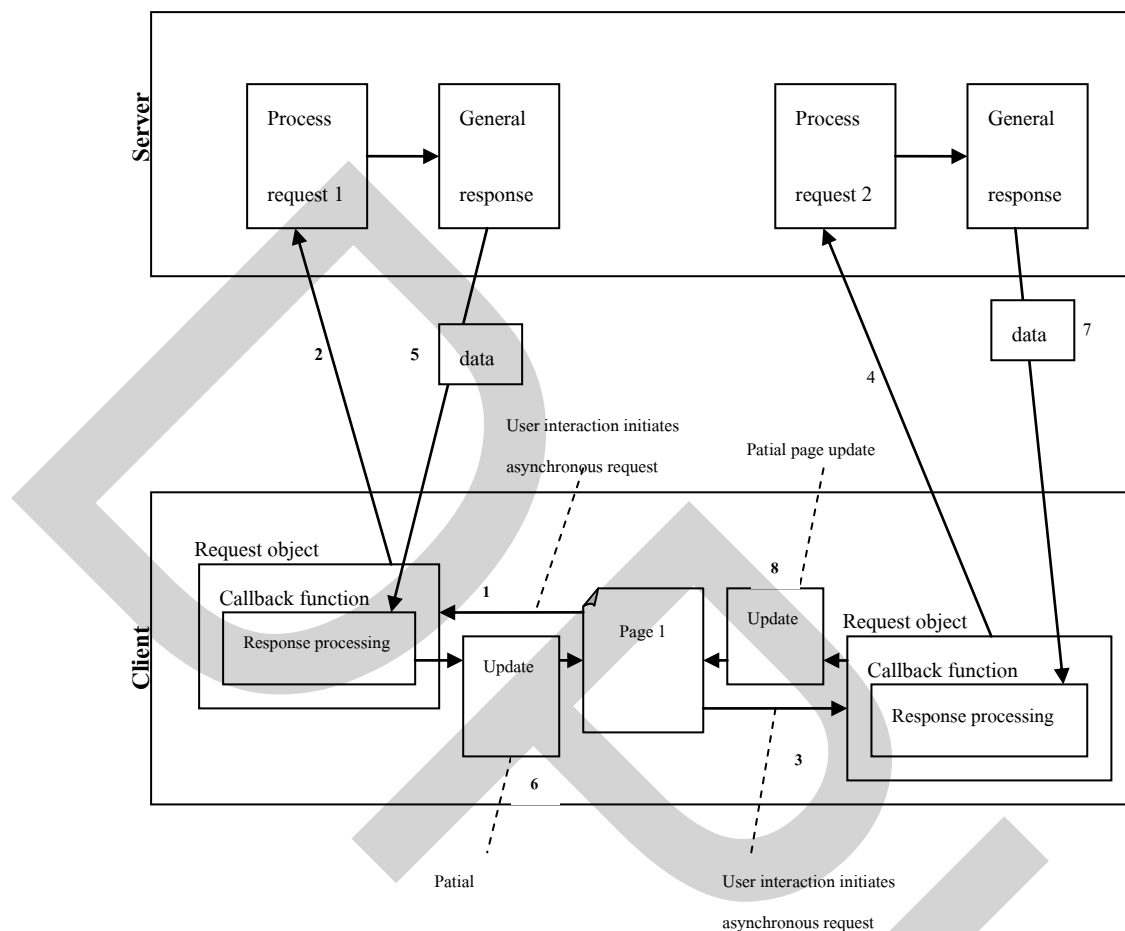
ชื่อแท็กสามารถบ่งบอกถึงวัตถุประสงค์ที่ชัดเจนในเนื้อหาขององค์ประกอบ เช่น แท็ก <name> และ </name>

แท็ก <graphic fileref= "figs/me.eps"/> สร้างความสัมพันธ์ (ลิงค์) ระหว่างส่วนของ XML และเพิ่มชื่อ me.eps. เพื่อนำเข้าข้อมูลภาพจากแฟ้มและแสดงในส่วน XML นี้

ในเอกสาร XML นั้น DOM เป็นกระบวนการหนึ่งของ XML โดย DOM สามารถที่จะทำการ อ่านและเขียน ในเอกสาร XML และผู้ใช้งานสามารถท่องไปยังแผนภาพต้นไม้ของเอกสารเพื่อเข้าไปยังโหนดต่างๆ ในขณะที่ไม่มีเอกสารก็ได้

2.4 AJAX (Asynchronous JavaScript And XML)

เป็นการใช้สคริปต์ฝั่งผู้ใช้งานเพื่อให้การใช้งานเว็บไซต์มีการตอบสนองได้มากขึ้น โดย AJAX จะแยกส่วนระหว่างผู้ใช้งานและเซิร์ฟเวอร์ โดยจะเรียกใช้งานแบบคู่ขนาน ทำให้ลดการประมวลผลล่าช้าของฝั่งเซิร์ฟเวอร์ซึ่งมีผลต่อประสบการณ์ของผู้ใช้งานได้



ภาพที่ 2.6 เว็บแอปพลิเคชันซึ่งเปิดใช้งาน AJAX ได้ต่อกับ Server

ที่มา : AJAX, Rich Internet Applications and Web Development for programmers books

จากภาพที่ 2.6 แสดงให้เห็นว่า AJAX ทำการเพิ่มชั้นระหว่างผู้ใช้งานและเซิร์ฟเวอร์เพื่อจัดการการสื่อสารระหว่างสองฝั่ง สามารถอธิบายได้ ดังต่อไปนี้

1. เมื่อผู้ใช้ได้ต่อกับ page ฝั่งผู้ใช้งานจะสร้าง XMLHttpRequest เพื่อจัดการคำขอ
2. XMLHttpRequest ได้ทำการส่งคำขอ ไปยังฝั่งเซิร์ฟเวอร์และรอการตอบกลับ
3. คำขอถูกส่งไม่พร้อมกัน ดังนั้นผู้ใช้สามารถที่จะได้ต่อกับโปรแกรมอื่นๆ บนฝั่งผู้ใช้งาน ในขณะที่เซิร์ฟเวอร์ทำการประมวลผลคำขอที่ส่งไปก่อนหน้านี้ควบคู่กัน
4. ผู้ใช้สามารถส่งคำขอเพิ่มเติมจากส่วนอื่นในหน้าเว็บเพจไปยังเซิร์ฟเวอร์ได้เช่นกัน
5. เซิร์ฟเวอร์ตอบกลับไปยังคำขอแรกที่ถูกส่งไป

6. XMLHttpRequest ที่ส่งค่าออกไป จะเรียกว่า ฟังก์ชันผู้ใช้งาน ซึ่งจะเป็นการประมวลผลข้อมูลกลับไปยังเซิร์ฟเวอร์ ฟังก์ชันนี้คือ ฟังก์ชันเรียกกลับ (callback function) ซึ่งจะใช้ในการปรับ page บางส่วนเพื่อแสดงข้อมูลบน webpage ที่ยังมีอยู่ โดยไม่ต้องโหลดทั้งหน้าของ page นั้น

7. เซิร์ฟเวอร์มีการตอบกลับไปยังค่าขอที่สอง

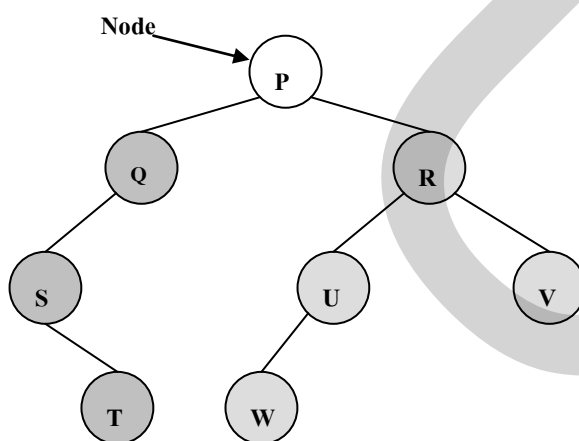
8. ผู้ใช้งานอาจจะเริ่มการปรับ page บางส่วน โดยฟังก์ชันเรียกกลับจะทำการปรับเฉพาะส่วนของ page ที่กำหนดไว้เท่านั้น

การปรับปรุง page บางส่วน ช่วยให้การใช้งานเว็บไซต์มีการตอบสนองมากขึ้นและเว็บไซต์ไม่ต้องโหลด page ใหม่ในขณะที่ผู้ใช้งานโต้ตอบด้วย

2.5 การแหวะผ่านต้นไม้ (Tree Traversal)

คือวิธีการเข้าถึงโหนดทั้งหมดที่มีอยู่ใน Tree โดย แบ่งเป็น 3 แบบ คือ

2.5.1 Preorder Traversal เป็นการแหวะผ่านต้นไม้ โดยจะเริ่มจากทุกโหนดทางด้านซ้ายก่อน จนกระทั่งไม่มีโหนดใดๆ แล้ว จึงย้ายไปยังโหนดด้านขวา หรือ หากไม่มีโหนดทางด้านขวาให้ดึงโหนดกลับ จนกระทั่งสามารถเคลื่อนย้ายไปยังโหนดทางด้านขวาได้ใหม่ และแหวะผ่านไปยังโหนดอื่นๆ จนถึงสิ้นสุดโหนดของต้นไม้

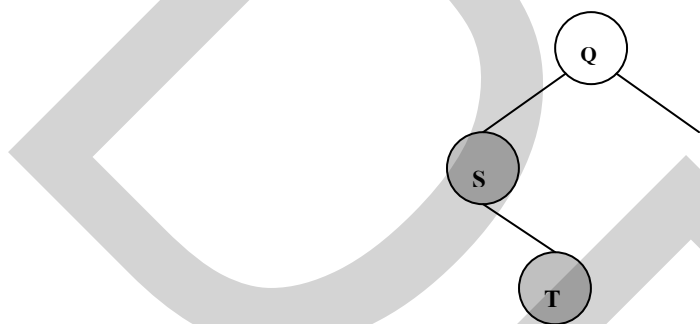


ภาพที่ 2.7 ต้นไม้ที่ใช้ในการแหวะผ่านแบบ Preorder

ที่มา : Data Structures and algorithms Concepts, Techniques and Applications books

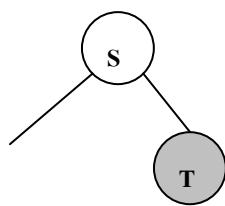
จากภาพที่ 2.7 มีโหนด P เป็นรูทโหนด และมีโหนดทางด้านซ้ายคือ โหนด Q, S และ T โหนดทางด้านขวา คือ โหนด R, U, W และ V สำหรับการแวะผ่านต้นไม้แบบ Preorder มีขั้นตอนดังนี้

1. จากรูทโหนด P จะแวะไปยังโหนดทางด้านซ้ายก่อน เริ่มจาก โหนด Q ซึ่งโหนด Q จะมีโหนดทางด้านซ้าย คือ โหนด S ส่วนทางด้านขวาของโหนด Q ไม่มีโหนดใดๆ ดังภาพที่ 2.8



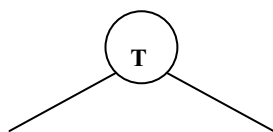
ภาพที่ 2.8 โหนดทางด้านซ้ายของรูทโหนด P (แบบ Preorder)

2. จากโหนด Q แวะผ่านต่อไปยังโหนด S พบว่า โหนด S ไม่มีโหนดใดๆ ทางด้านซ้าย แต่มีโหนดทางด้านขวา คือ โหนด T



ภาพที่ 2.9 ลำดับชั้นของโหนด S (แบบ Preorder)

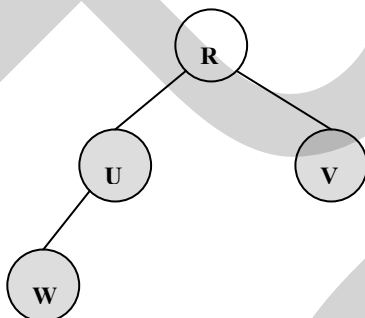
3. จากโหนด S แวะผ่านต่อไปยังโหนด T พบว่า โหนด T เป็นโหนดสุดท้ายทางด้านซ้ายมือของรูทโหนด P เนื่องจาก โหนด T ไม่มีโหนดใดๆ ต่อ ทั้งทางด้านซ้ายและด้านขวา



ภาพที่ 2.10 ลำดับชั้นของโหนด T (แบบ Preorder)

ดังนั้น การแหว่ผ่านต้นไม้แบบ Preorder ทางด้านซ้ายมือ จะได้ผลลัพธ์ คือ QST
เมื่อแหว่ผ่านโหนดทางด้านซ้าย จนกระทั่งไม่พบโหนดใดๆ แล้วนั้น จึงได้ย้ายไปยัง
โหนดขวามือของรูทโหนด P

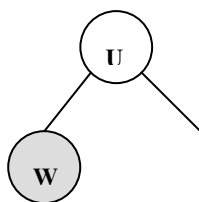
4. จากรูทโหนด P แหว่ผ่านไปยังโหนดทางด้านขวา เริ่มจากโหนด R โดยโหนด R จะมีโหนดทั้งทางด้านซ้ายและด้านขวา ซึ่งโหนดทางด้านซ้าย คือ โหนด U และ โหนดทางด้านขวาคือ โหนด V ดังภาพ



ภาพที่ 2.11 โหนดทางด้านขวาของรูทโหนด P (แบบ Preorder)

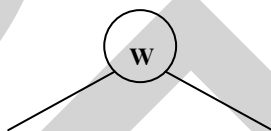
การแหว่ผ่านต้นไม้แบบ Preorder นั้น จะต้องแหว่ผ่านโหนดทางด้านซ้ายก่อน ดังนั้น โหนดถัดไปที่จะแหว่ผ่านคือ โหนด U

5. จากโหนด R แหว่ผ่านต่อไปยังโหนด U พบว่า โหนด U มีโหนดทางด้านซ้าย คือ โหนด W ส่วนทางด้านขวาไม่มีโหนดใดๆ



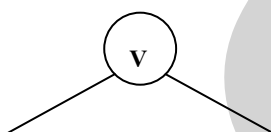
ภาพที่ 2.12 ลำดับชั้นของโหนด U (แบบ Preorder)

6. จากโหนด U แวะผ่านต่อไปยังโหนด W พบว่า โหนด W ไม่มีโหนดใดๆ ทางด้านซ้ายและด้านขวา โหนด W จึงเป็นโหนดสุดท้ายทางด้านซ้ายมือของโหนด R



ภาพที่ 2.13 ลำดับชั้นของโหนด W (แบบ Preorder)

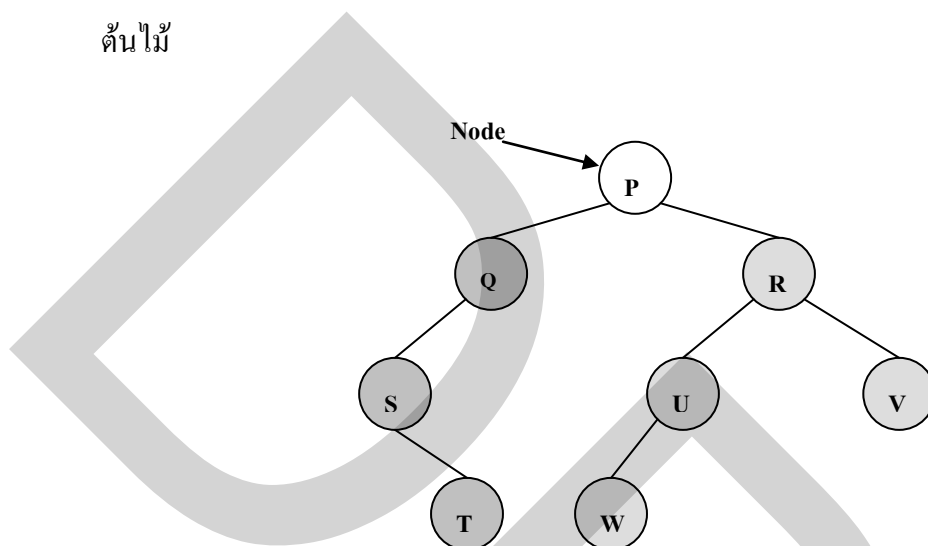
7. เมื่อแวะผ่านโหนดทั้งหมดทางด้านซ้ายของโหนด R แล้ว จึงได้ย้ายมายังโหนดด้านขวาของโหนด R ซึ่งก็คือ โหนด V พบว่า โหนด V ไม่มีโหนดใดๆ ทางด้านซ้ายและด้านขวา โหนด V จึงเป็นโหนดสุดท้ายของต้นไม้



ภาพที่ 2.14 ลำดับชั้นของโหนด V (แบบ Preorder)

ดังนั้น การแวะผ่านต้นไม้แบบ Preorder ทางด้านขวา จะได้ผลลัพธ์ คือ RUWV
 เพราะฉะนั้น ผลลัพธ์ของการแวะผ่านต้นไม้แบบ Preorder จากภาพที่ 2.7 คือ PQSTRUWV

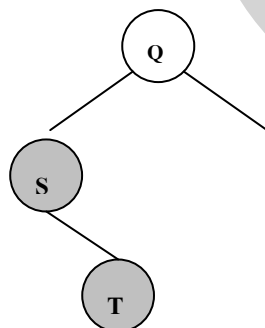
2.5.2 Inorder Traversal เป็นการแวะผ่านต้นไม้ โดยจะเริ่มจากโหนดด้านซ้ายก่อน จนกระทั่งไม่สามารถเคลื่อนย้ายได้อีก จากนั้นประมวลผลโหนดและย้ายไปโหนดด้านขวา ในกรณีที่ไม่มีโหนดใดๆ ทางด้านขวาแล้ว ให้ถอยโหนดกลับและแวะผ่านไปยังโหนดอื่นๆ จนสิ้นสุดโหนดของต้นไม้



ภาพที่ 2.15 ต้นไม้ที่ใช้ในการแวะผ่านแบบ Inorder

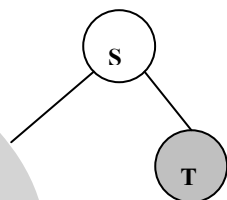
จากภาพที่ 2.15 มีโหนด P เป็นรากโหนด และมีโหนดทางด้านซ้ายคือ โหนด Q, S และ T โหนดทางด้านขวา คือ โหนด R, U, W และ V สำหรับการแวะผ่านต้นไม้แบบ Inorder มีขั้นตอน ดังนี้

1. จากรากโหนด P ไปยังโหนดด้านซ้ายมือ ดังภาพ ซึ่งการแวะผ่านต้นไม้แบบ Inorder นั้นจะผ่านโหนด Q ไปก่อน เพื่อไปยังโหนดสุดท้ายของโหนดด้านซ้ายมือจึงกลับมาแวะผ่านโหนด Q



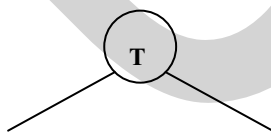
ภาพที่ 2.16 โหนดด้านซ้ายมือของรากโหนด P (แบบ Inorder)

2. เมื่อผ่านโหนด Q มาก่อนเพื่อไปยังโหนด S พบว่าโหนด S มีโหนด T ด้านขวาและไม่มีโหนดใดๆ ทางด้านซ้าย จึงแวะผ่านไปยังโหนด T ต่อไป



ภาพที่ 2.17 ลำดับชั้นของโหนด S (แบบ Inorder)

3. จากโหนด S แวะผ่านไปยังโหนด T พบว่า โหนด T ไม่มีโหนดใดๆ ทางด้านซ้ายและด้านขวา ถือเป็นโหนดสิ้นสุดทางด้านซ้าย เมื่อแวะผ่านโหนดสุดท้ายด้านซ้ายแล้วนั้น จะต้องย้อนกลับไปยังโหนด Q อีกครั้ง

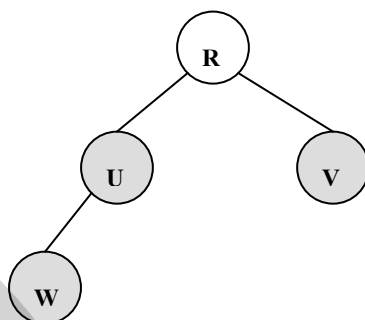


ภาพที่ 2.18 ลำดับชั้นของโหนด T (แบบ Inorder)

ดังนั้น การแวะผ่านต้นไม้แบบ Inorder ทางด้านซ้ายของรูทโหนด P จะได้ผลลัพธ์คือ STQ

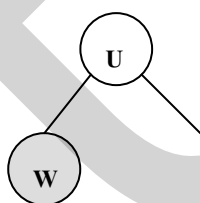
เมื่อแวะผ่านโหนดทางด้านซ้ายเรียบร้อยแล้วนั้น จะแวะผ่านรูทโหนด P ก่อน จึงย้ายไปยังโหนดด้านขวาของรูทโหนด P

4. จากรูทโหนด P จะไปยังโหนดด้านขวา จะผ่านโหนด R ไปก่อน เพื่อแวะผ่านไปยังโหนด U ซึ่งเป็นโหนดด้านซ้ายของโหนด R



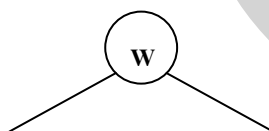
ภาพที่ 2.19 โหนดด้านขวาของรูทโหนด P (แบบ Inorder)

5. เมื่อไปยังโหนด U พบว่าโหนด U มีโหนดถัดไปทางด้านซ้ายคือ โหนด W ส่วนทางด้านขวาไม่มีโหนดใดๆ จึงผ่านโหนด U ไปก่อน เพื่อไปยังโหนด W



ภาพที่ 2.20 ลำดับชั้นของโหนด U (แบบ Inorder)

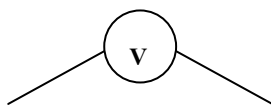
6. จากการผ่านโหนด U มาก่อน เพื่อที่จะมาถึงโหนด W พบว่า โหนด W ไม่มีโหนดใดๆ ทั้งทางด้านซ้ายและขวา ซึ่งถือว่าเป็นโหนดสุดท้ายของโหนดด้านซ้ายของโหนด R



ภาพที่ 2.21 ลำดับชั้นของโหนด W (แบบ Inorder)

เมื่อแวะผ่านไปยังโหนด W เรียบร้อยแล้วนั้น จะกลับมายังโหนด U และโหนด R เพื่อแวะผ่านต่อไปยังโหนด V ซึ่งอยู่ทางขวาของโหนด R

7. จากการแวะผ่านไปยังโหนด V พบว่าโหนด V ไม่มีโหนดใดๆ ทางด้านซ้ายและด้านขวาถือว่าเป็นโหนดสิ้นสุดของต้นไม้

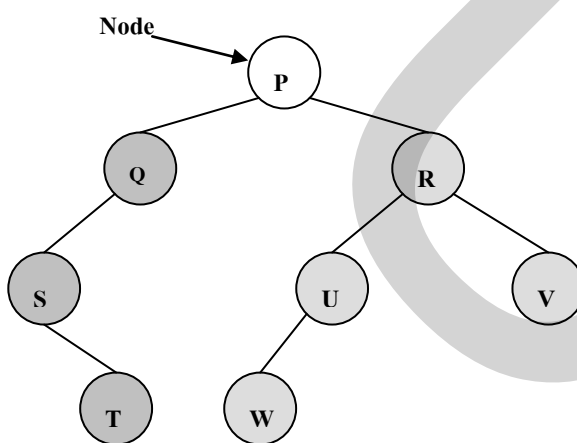


ภาพที่ 2.22 ลำดับชั้นของโหนด V (แบบ Inorder)

ดังนั้น การแวะผ่านต้นไม้แบบ Inorder ทางด้านขวา จะได้ผลลัพธ์ คือ WURV

เพราะฉะนั้น ผลลัพธ์ของการแวะผ่านต้นไม้แบบ Inorder จากภาพที่ 2.15 คือ STQPWURV

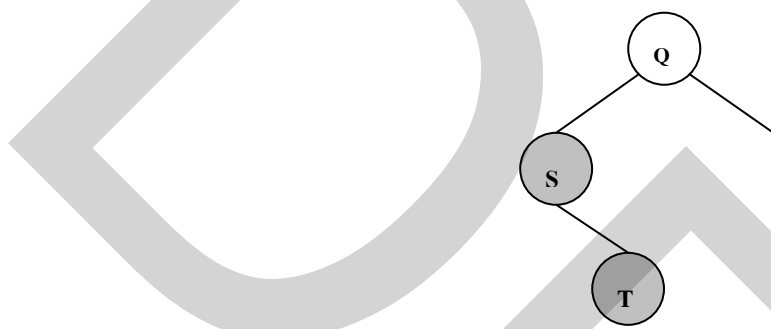
2.5.3 Postorder Traversal เป็นการแวะผ่านต้นไม้ โดยจะเริ่มจากโหนดด้านซ้าย จนกระทั่งไม่มีโหนดใดๆ ทางด้านซ้าย จึงเคลื่อนย้ายไปยังโหนดด้านขวา หรือ หากไม่มีโหนดด้านขวาประมวลผลโหนดและดึงโหนดมาตามทิศทางเดิม และแวะผ่านไปยังโหนดอื่นๆ จนถึงสิ้นสุดโหนดของต้นไม้



ภาพที่ 2.23 ต้นไม้ที่ใช้ในการแวะผ่านแบบ Postorder

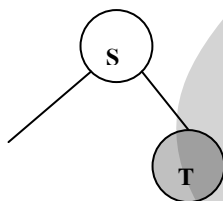
จากภาพที่ 2.23 มีโหนด P เป็นรากโหนด และมีโหนดทางด้านซ้ายคือ โหนด Q, S และ T โหนดทางด้านขวา คือ โหนด R, U, W และ V สำหรับการแวะผ่านต้นไม้แบบ Postorder มีขั้นตอน ดังนี้

1. จากรากโหนด P ไปยังโหนดด้านซ้ายมือ ซึ่งการแวะผ่านต้นไม้แบบ Postorder นั้นจะผ่านโหนด Q เพื่อไปยังโหนดถัดไปก่อน นั่นคือโหนด S



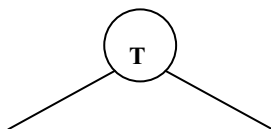
ภาพที่ 2.24 โหนดด้านซ้ายมือของรากโหนด P (แบบ Postorder)

2. เมื่อมายังโหนด S พบว่า โหนด S ยังมีโหนด T อยู่ถัดไป จึงผ่านโหนด S ไปก่อน เพื่อแวะผ่านไปยังโหนด T



ภาพที่ 2.25 ลำดับชั้นของโหนด S (แบบ Postorder)

3. เมื่อมาถึงโหนด T พบว่าโหนด T คือโหนดสุดท้ายของโหนดด้านซ้ายของรากโหนด P เนื่องจากไม่มีโหนดใดๆ ทางด้านซ้ายและด้านขวา จากนั้นจึงกลับไปยังโหนด S และ โหนด Q ตามลำดับ

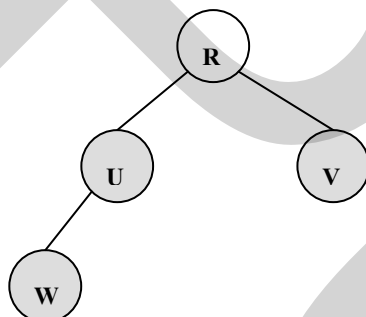


ภาพที่ 2.26 ลำดับชั้นของโหนด T (แบบ Postorder)

ดังนั้น การแหว่ผ่านต้นไม้แบบ Postorder ทางด้านซ้ายของรูทโหนด P จะได้ผลลัพธ์ คือ TSQ

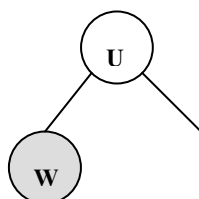
เมื่อแหว่ผ่านโหนดทางด้านซ้ายเรียบร้อยแล้วนั้น จะย้ายไปยังโหนดด้านขวาของรูทโหนด P ทันที โดยไม่ผ่านรูทโหนด P ก่อน

4. โหนดด้านขวาของรูทโหนด P การแหว่ผ่านต้นไม้ นั้น จะผ่านโหนด R ไปก่อน เพื่อแหว่ผ่านไปยังโหนด U ซึ่งเป็นโหนดด้านซ้ายของโหนด R



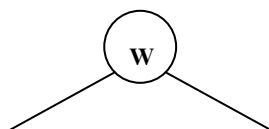
ภาพที่ 2.27 โหนดด้านขวาของรูทโหนด P (แบบ Postorder)

5. เมื่อมาถึงโหนด U พบว่าโหนด U มีโหนดถัดไปคือ โหนด W ส่วนทางด้านขวาไม่มีโหนดใดๆ จึงผ่านโหนด U ไปก่อนเพื่อแหว่ไปยังโหนด W



ภาพที่ 2.28 ลำดับชั้นของโหนด U (แบบ Postorder)

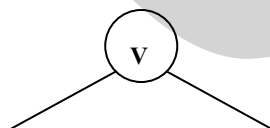
6. โหนด W ไม่มีโหนดใดๆ ทางด้านซ้ายและด้านขวา ดังนั้น โหนด W จึงเป็นโหนดสุดท้ายทางด้านซ้ายของโหนด R และย้อนกลับไปยังโหนด U ทันที



ภาพที่ 2.29 ลำดับชั้นของโหนด W (แบบ Postorder)

เมื่อย้อนกลับไปยังโหนด U แล้วนั้น จะแวะผ่านต่อไปยังโหนด V ซึ่งเป็นโหนดด้านขวาของโหนด R

7. เมื่อแวะผ่านไปยังโหนด V พบว่าโหนด V เป็นโหนดสิ้นสุดของต้นไม้ เนื่องจากโหนด V ไม่มีโหนดใดๆ ทางด้านซ้ายและด้านขวา หลังจากที่แวะผ่านโหนด V แล้วนั้น จะย้อนกลับไปยังโหนด R และรูปโหนด P ตามลำดับ

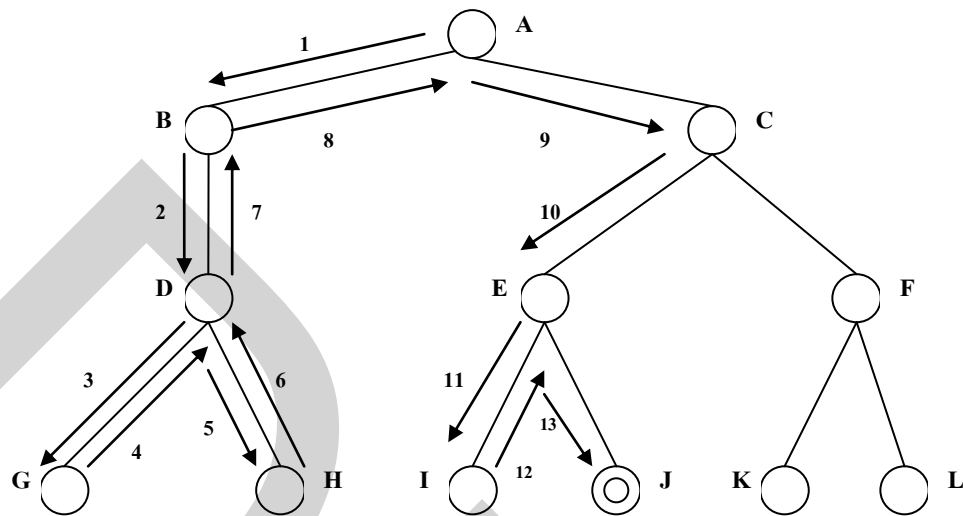


ภาพที่ 2.30 ลำดับชั้นของโหนด V (แบบ Postorder)

ดังนั้น การแวะผ่านต้นไม้แบบ Postorder ทางด้านขวา จะได้ผลลัพธ์ คือ WUVR เพราะฉะนั้น ผลลัพธ์ของการแวะผ่านต้นไม้แบบ Postorder จากภาพที่ 2.23 คือ TSQWUVRP

2.6 การค้นหาในแนวลึก (Depth-first Search)

เป็นการค้นหาโหนดแบบบนลงล่าง โดยจะเริ่มจากโหนดด้านซ้ายก่อนเสมอ หากไม่พบโหนดเป้าหมายด้านซ้าย ให้ย้อนโหนดไปยังรูปโหนดและค้นหาไปยังโหนดด้านขวาต่อไป จนกระทั่งพบโหนดเป้าหมาย



ภาพที่ 2.31 รูปแบบการค้นหาแนวลึก

ที่มา : Artificial Intelligence Illuminated books

จากภาพที่ 2.31 โหนดเป้าหมายคือ โหนด J ซึ่งการค้นหาจากภาพ จะมีรูทโหนด คือ โหนด A โดยโหนด A นั้นมีโหนดทางด้านซ้ายและด้านขวาประกอบ ซึ่งการค้นหาจะเริ่มจาก โหนดทางด้านซ้ายก่อนเสมอ

การค้นหาจึงเริ่มจาก โหนด A ไปยังโหนด B ซึ่งโหนด B มีโหนดถัดไปเพียงโหนดเดียว คือ โหนด D จึงไปยังโหนด D ได้ทันที เมื่อไปยังโหนด D พบว่าโหนด D มีโหนดลูกคือ โหนด G อยู่ฝั่งซ้าย และโหนด H อยู่ฝั่งขวา ซึ่งการค้นหาแนวลึกต้องเริ่มจากฝั่งซ้ายก่อนเสมอ จึงค้นหาไปยังโหนด G และกลับไปยังโหนด D เพื่อค้นหาไปยังโหนด H ถัดไป จากโหนด H ซึ่งเป็นโหนดสุดท้ายของโหนดด้านซ้ายของรูทโหนด A จะย้อนกลับเพื่อไปยังรูทโหนด A โดยจากโหนด H กลับไปยังโหนด D โหนด B และรูทโหนด A ตามลำดับ

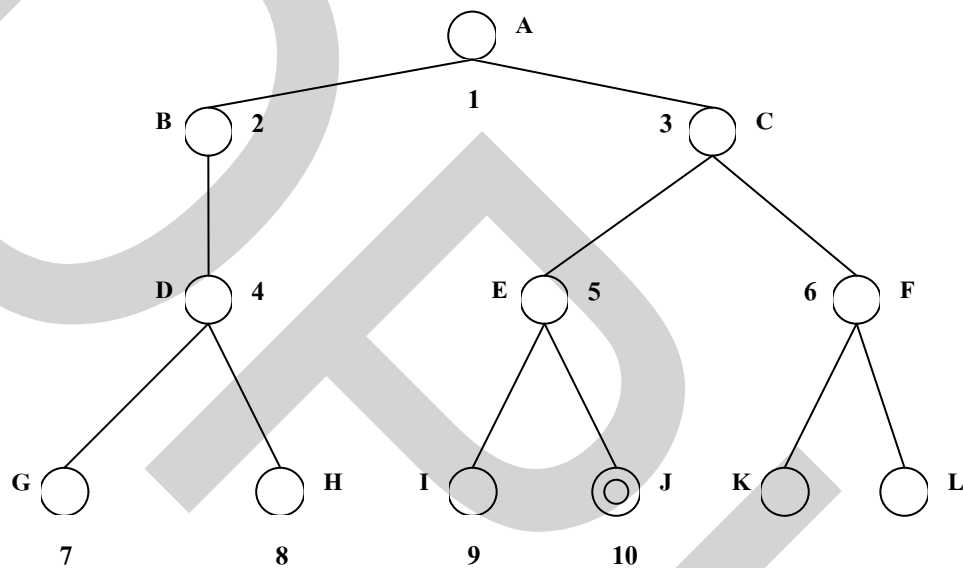
เมื่อสิ้นสุดการค้นหาโหนดทางด้านซ้ายแล้วนั้น พบว่าโหนดทางด้านซ้ายไม่มีโหนดเป้าหมาย จึงย้ายไปค้นหาฝั่งโหนดด้านขวาต่อไป

จากรูทโหนด A ไปยังโหนด C ซึ่งโหนด C มีโหนดทั้งทางด้านซ้ายและด้านขวา คือ โหนด E อยู่ทางด้านซ้ายและโหนด F อยู่ทางด้านขวา จากโหนด C ต้องค้นหาไปยังโหนด E ก่อน เนื่องจากเป็นโหนดทางด้านซ้าย เมื่อไปถึงโหนด E พบว่า โหนด E มีโหนด I และโหนด J เป็นโหนดลูก ซึ่งโหนด J เป็นโหนดเป้าหมาย การค้นหาจากโหนด E เพื่อไปยังโหนดเป้าหมายซึ่งอยู่

ด้านขวา จะต้องค้นหาจากโหนด E ไปยังโหนด I ซึ่งอยู่ด้านซ้ายก่อน และจึงย้อนกลับมายังโหนด E เพื่อไปยังโหนด J ซึ่งเป็นโหนดเป้าหมาย เมื่อพบโหนดเป้าหมายแล้วนั้นให้หยุดการค้นหาทันที

2.7 การค้นหาในแนวกว้าง (Breadth-first Search)

เป็นการค้นหาไปตามระดับของต้นไม้ จากรูทโหนดลงไป โดยจะค้นหาทุกโหนดในหนึ่งระดับ ซึ่งจะค้นหาจากโหนดด้านซ้ายไปยังโหนดด้านขวา จนกระทั่งพบโหนดเป้าหมาย



ภาพที่ 2.32 รูปแบบการค้นหาในแนวกว้าง

ที่มา : Artificial Intelligence Illuminated books

จากภาพที่ 2.32 โหนดเป้าหมายคือ โหนด J ซึ่งการค้นหาจากภาพ จะมีรูทโหนด คือ โหนด A โดยโหนด A นั้นมีโหนดทางด้านซ้ายและด้านขวาประกอบ ซึ่งการค้นหาจะเริ่มจากโหนดทางด้านซ้ายก่อน แต่เป็นการค้นหาแบบทีละระดับ จากภาพที่ 2.32 แสดงให้เห็นว่าต้นไม้มี 4 ระดับ

การค้นหาเริ่มจาก โหนด A ซึ่งเป็นรูทโหนด ไปยังระดับที่ 2 ประกอบด้วยโหนด B และ โหนด C การค้นหาจึงค้นหาจากโหนดด้านซ้ายคือโหนด B และจึงไปยังโหนด C ซึ่งเป็นโหนดสุดท้ายของระดับที่ 2 จากนั้นจึงไปค้นหาในระดับที่ 3 ซึ่งมีโหนด D, E และ F ตามลำดับ การค้นหาเริ่มจากโหนดซ้ายของระดับที่ 3 นั่นก็คือ โหนด D ไปยังโหนด E และโหนด F ซึ่งเป็น

โหนดสุดท้ายของระยะที่ 3 ก็ยังไม่พบโหนดเป้าหมาย จึงค้นหาไปยังระดับสุดท้าย คือระดับที่ 4 โดยจะเห็นว่าระดับที่ 4 มีโหนด G, H, I, J, K และ L ซึ่งมีโหนด J ที่เป็นโหนดเป้าหมายอยู่ การค้นหาเริ่มจากทางด้านซ้ายเช่นเดิม คือจากโหนด G ไปยังโหนด H, โหนด I และโหนด J โดยโหนด J เป็นโหนดเป้าหมาย เมื่อพบโหนดเป้าหมาย การค้นหาจะสิ้นสุดทันที

2.8 ต้นไม้ตัดสินใจ (Decision Tree)

เป็นวิธีการที่เป็นระบบ โดยใช้แผนภาพต้นไม้ ต้นไม้ตัดสินใจประกอบด้วย โหนดและ กิ่ง (branches) โดยแต่ละโหนดจะแตกโหนดออกไปโดยมีกิ่งเป็นส่วนเชื่อมต่อ ซึ่งจะแตกโหนด จากด้านซ้ายไปขวา ซึ่งต้นไม้ตัดสินใจ ช่วยลดความยุ่งยากซับซ้อนได้

ตัวอย่าง การใช้ต้นไม้ตัดสินใจในการอนุมัติสินเชื่อ

ตารางแสดง ข้อมูลผู้ขอสินเชื่อ มีคุณสมบัติ 4 ข้อ ดังนี้

1. Age : คุณสมบัตินี้จะมีค่าที่เป็นไปได้อยู่ 3 ค่า คือ Young, middle และ old
2. Has_job : คุณสมบัตินี้บ่งบอกถึงผู้ขอสินเชื่อมีงานทำหรือไม่ โดยจะมีค่าที่เป็นไปได้ อยู่ 2 ค่า คือ true หมายถึงมีงานทำ และ false หมายถึงไม่มีงานทำ
3. Own_house : คุณสมบัตินี้บ่งบอกถึงผู้ขอสินเชื่อเป็นเจ้าของบ้านเองหรือไม่
4. Credit_rating : คุณสมบัตินี้จะมีค่าที่เป็นไปได้อยู่ 3 ค่า คือ fair, good และ excellent สำหรับแถวสุดท้ายของตาราง (Class) แสดงถึง การสมัครสินเชื่อถูกอนุมัติหรือไม่จาก ในอดีต ซึ่งแสดงค่าโดยใช้ Yes และ No

ตารางที่ 2.1 แสดงข้อมูลผู้ขออนุมัติสินเชื่อ

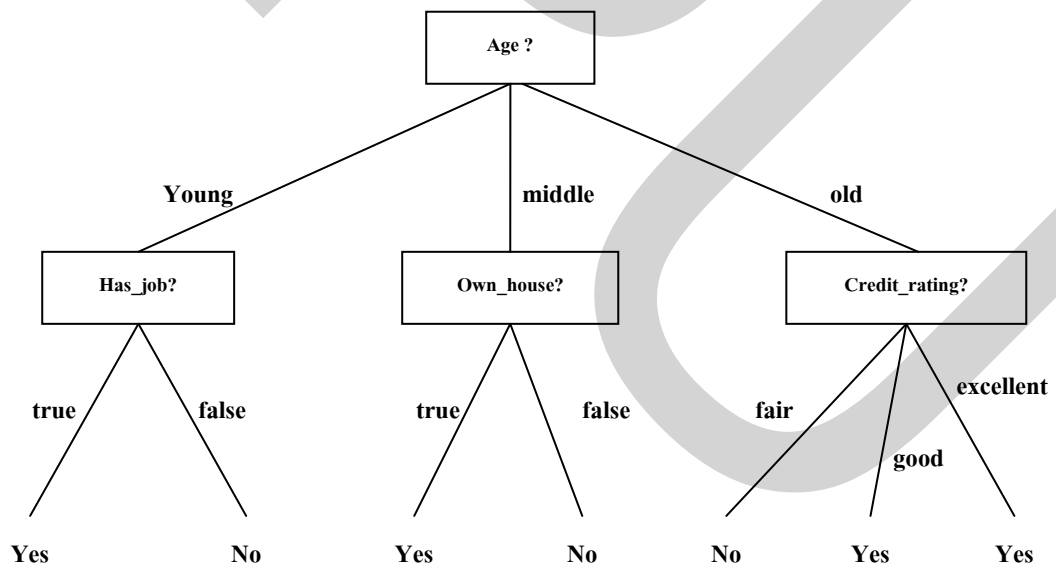
ID	Age	Has_job	Own_house	Credit_rating	Class
1	Young	false	false	fair	No
2	Young	false	false	good	No
3	Young	true	false	good	Yes
4	Young	true	true	fair	Yes
5	Young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes

ตารางที่ 2.1 (ต่อ)

10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

ที่มา : Data Mining Concepts. Methods and Applications in Management and Engineering Design books

จากตารางที่ 2.1 สามารถนำต้นไม้ตัดสินใจมาใช้ในการจำแนกหรือแบ่งประเภทข้อมูล และทำให้ข้อมูลมีประสิทธิภาพขึ้นได้ เมื่อนำข้อมูลในตารางที่ 2.1 มาสร้างต้นไม้ตัดสินใจ จะได้โครงสร้างต้นไม้ ดังภาพ



ภาพที่ 2.33 ต้นไม้ตัดสินใจสำหรับข้อมูลในตารางที่ 2.1

ที่มา : Data Mining Concepts. Methods and Applications in Management and Engineering Design books

จากภาพที่ 2.33 อธิบายได้ว่า รุทโหนดของต้นไม้ตัดสินใจนี้ คือ Age ซึ่งเป็นคำถามโดยทั่วไปที่ใช้ถามในกรณีการขออนุมัติสินเชื่อ โดยมีคำตอบหรือผลที่เป็นไปได้อยู่ 3 ค่า คือ Young, middle และ old โดยทั้ง 3 นี้จะแตกกิ่งออกเป็น 3 กิ่ง คือ Has_job?, Own_house? และ Credit_rating? โดยโหนดปลายจะเป็นแสดงถึง Class ซึ่งแสดงค่า Yes หรือ No

2.9 การตั้งชื่อจุลินทรีย์

ในระบบของระบบการตั้งชื่อ จุลินทรีย์มีสองส่วน ประกอบด้วย สกุล (Genus) และ ชนิด (Species) ของจุลินทรีย์ ตัวอย่างเช่น แบคทีเรียที่พบตามปกติในลำไส้ใหญ่ของมนุษย์ เรียกว่า Escherichia coli. ส่วนแรกของชื่อ คือ Escherichia เป็นชื่อสกุล ซึ่งเป็นชื่อที่ได้มาจากชื่อของนักวิทยาศาสตร์ Theodor Escherich โดยเป็นนักวิทยาศาสตร์คนแรกที่รายงานการดำรงอยู่ของจุลินทรีย์ ในปี ค.ศ.1888 ส่วนที่สองของชื่อ คือ coli เป็นชื่อที่ได้มาจากคำว่า colon (ลำไส้ใหญ่) ซึ่งเป็นแบคทีเรียที่พบครั้งแรกโดย Escherich ซึ่ง Escherichia coli สามารถเขียนสั้นๆ ได้ว่า E.coli

2.10 การค้นหาแบบต่างๆ

Linear Search หรือ การค้นหาแบบเชิงเส้น เป็นขั้นตอนของการค้นหองค์ประกอบเป้าหมายที่อยู่ในการเรียงลำดับรายชื่อ องค์ประกอบที่คาดหมายไว้คือการค้นหาในโครงสร้างข้อมูลทั้งหมดในขั้นตอนที่ต่อเนื่องกันจากองค์ประกอบแรกจนถึงองค์ประกอบสุดท้าย การค้นหาแบบเชิงเส้นนี้ไม่ยุ่งยากซับซ้อนและตรงไปตรงมา แต่มีข้อจำกัดของการค้นหา คือ ใช้เวลาในการค้นหามาก ชื่อของการค้นหาแบบเชิงเส้นนี้มีความหมายเป็นนัยว่ารายการหรือองค์ประกอบจะเก็บไว้ในวิธีการที่เป็นระบบ ซึ่งการค้นหาแบบเชิงเส้นนี้สามารถใช้กับโครงสร้างข้อมูลที่มีการเรียงลำดับหรือไม่มีการเรียงลำดับก็ได้

ประสิทธิภาพทางเวลาของการค้นหาแบบเชิงเส้น (Linear search) คือ $O(N)$ หมายถึงจำนวนของการเปรียบเทียบจะเพิ่มขึ้นทันทีกับจำนวนขององค์ประกอบ เช่น ใช้การค้นหาแบบโดยตรงในการค้นหาลิสต์ที่ไม่มีมีการเรียงลำดับจำนวน 512 องค์ประกอบ ในเหตุการณ์ที่แย่ที่สุดคือจำนวนสูงสุดของการเปรียบเทียบที่ต้องการหาค่า คือ 512 หรือหากจำนวนองค์ประกอบ คือ 1,024 จำนวนสูงสุดของการเปรียบเทียบ คือ 1,024 ดังนั้นประสิทธิภาพทางเวลาในการค้นหาแบบเชิงเส้นจะใช้เวลาามาก หรือหากข้อมูลมีอยู่มากจะทำให้ใช้เวลาในการค้นหาเหมือนกัน ซึ่งทำให้เกิดความล่าช้าขึ้นได้

Binary Search หรือ การค้นหาแบบพัวครึ่ง จะแตกต่างจากการค้นหาแบบเชิงเส้นและมีความเร็วกว่าการหาแบบเชิงเส้น การค้นหาแบบพัวครึ่งนี้เป็นเทคนิคการค้นหาที่มีประสิทธิภาพและทำงานกับข้อมูลที่มีการเรียงลำดับแล้วเท่านั้น การค้นหาแบบพัวครึ่งนี้องค์ประกอบเป้าหมายจะเปรียบเทียบกับองค์ประกอบที่อยู่ในตำแหน่งกลาง หากองค์ประกอบที่คาดหมายไว้มีน้อยกว่าองค์ประกอบที่อยู่ในตำแหน่งกลางจะค้นหาจากส่วนทางด้านซ้ายก่อนและค้นหาจากส่วนทางด้านขวาต่อไป

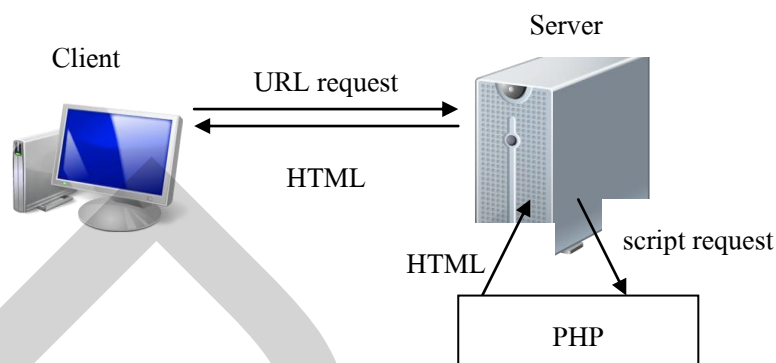
สำหรับประสิทธิภาพทางเวลาของการค้นหาแบบพัวครึ่ง คือ $O(\log_2 N)$ ซึ่งการค้นหาแบบพัวครึ่งจะต้องมีการเรียงลำดับข้อมูลก่อน จึงจะทำการค้นหาได้ เช่น การค้นหาลิสต์ที่มีการเรียงลำดับจำนวน 512 องค์ประกอบ โดยการเปรียบเทียบลิสต์ในแต่ละครั้งจะมีการพัวครึ่ง และจำนวนของการหารจะลดลงไปยังองค์ประกอบเดียวที่คงเหลือ นั่นคือ 9 จากความเป็นจริง 512 คือ 2^9 ซึ่งเลขยกกำลัง คือ 9 แทนจำนวนของเวลาที่ 2 หาร ได้ลงตัว ซึ่งก็คือ ลอการิทึมฐาน 2 ของ 512 หรือ $\log_2 512$ ซึ่งการค้นหาแบบพัวครึ่งนี้มีประสิทธิภาพดีกว่าการค้นหาแบบเชิงเส้น เพราะจำนวนของการเปรียบเทียบมีค่าน้อยกว่ามาก

การค้นหาแบบพัวครึ่ง (Binary search) ต้องเรียงลำดับข้อมูล ก่อนทำการค้นหา โดยการเรียงลำดับมีประสิทธิภาพทางเวลา $O(\log_2 N)$ และการค้นหาแบบพัวครึ่งจะมีประสิทธิภาพทางเวลา $O(\log_2 N)$ หมายความว่า การค้นหาแบบพัวครึ่ง (Binary search) ยังคงมีการเพิ่มขึ้นของจำนวนการเปรียบเทียบ เมื่อจำนวนชื่อจูลินทรีย์ที่เพิ่มขึ้น แต่มีอัตราการเติบโตของจำนวนการเปรียบเทียบที่ต่ำมาก เมื่อเปรียบเทียบกับการค้นหาแบบเชิงเส้น (Linear Search) ที่จำนวนครั้งในการเปรียบเทียบเติบโตตามจำนวนชื่อจูลินทรีย์โดยตรง $O(N)$

2.11 PHP (PHP:Hypertext Preprocessor)

PHP (PHP:Hypertext Preprocessor) คือภาษาสคริปต์ที่ใช้สำหรับสร้างเว็บเพจแบบไดนามิก ซึ่ง PHP นี้ถูกสร้างขึ้นในปี ค.ศ. 1995 โดย Rasmus Lerdorf ซึ่ง PHP เป็นเทคโนโลยีโอเพ่นซอร์สที่มีการสนับสนุนโดยสังคมกลุ่มใหญ่จากผู้ใช้งานและนักพัฒนา PHP คือแพลตฟอร์มที่เป็นอิสระสำหรับระบบปฏิบัติการต่างๆ เช่น ลินุกซ์ แม็ค วินโดวส์ เป็นต้น รวมทั้ง PHP ยังสนับสนุนฐานข้อมูลต่างๆ เช่น MySQL เป็นต้น

การทำงานของ PHP นั้น PHP คือภาษาสำหรับฝั่งเซิร์ฟเวอร์ ซึ่งหมายถึงโค้ดที่ถูกเขียนขึ้นใน PHP มีอยู่บนคอมพิวเตอร์แม่ข่ายที่ใช้เว็บเพจไปยังเว็บเบราว์เซอร์



ภาพที่ 2.34 กระบวนการการทำงานระหว่าง ลูกค้า เซิร์ฟเวอร์ และ โมดูล PHP

ที่มา : PHP for the web books

จากภาพที่ 2.34 เมื่อลูกค้าต้องการเข้าสู่เว็บไซต์ต่างๆ เช่น www.DMCinsights.com ผู้ให้บริการอินเทอร์เน็ตจะส่งรีเควสไปยังเซิร์ฟเวอร์ซึ่งมีข้อมูลของ www.DMCinsights.com จากนั้น เซิร์ฟเวอร์จะอ่านโค้ด PHP และดำเนินการตามคำสั่งสคริปต์ โค้ด PHP บอกเซิร์ฟเวอร์ที่จะส่งข้อมูลเว็บเพจที่จัดสรรไว้ไปยังเบราว์เซอร์ในรูปแบบของ HTML

2.12 งานวิจัยที่เกี่ยวข้อง

ชลธิศา พลทองมาก (2553) ศึกษาเรื่อง การวิเคราะห์ความเสี่ยงการเป็นโรคไวรัสตับอักเสบบี โดยต้นไม้อัดสติใจ การศึกษาครั้งนี้ใช้ต้นไม้อัดสติใจและทฤษฎีเบย์เซียนในการสร้างแบบพยากรณ์ความเสี่ยงในการเกิดโรคไวรัสตับอักเสบบี ซึ่งก่อนจะสร้างแบบพยากรณ์นั้น ได้ทำการศึกษาปัจจัยเสี่ยงในการเกิดโรค จากแพทย์และงานวิจัยทางการแพทย์ รวมไปถึงทำแบบสอบถามและเก็บข้อมูลจากกลุ่มตัวอย่างทั้งสิ้น 661 ราย ซึ่งแบ่งเป็นผู้ป่วยติดเชื้อ HCV 76 ราย และไม่ติดเชื้อ HCV 585 ราย และทำการวิเคราะห์ข้อมูลที่ได้มาด้วยโปรแกรม Weka จึงจะสร้างแบบพยากรณ์ขึ้น ซึ่งผลของการใช้ต้นไม้อัดสติใจในการพยากรณ์ให้ผลความแม่นยำ 92.99% และทฤษฎีเบย์เซียนให้ความแม่นยำ 93.86%

วาสนา วงษ์กำภู (2553) ศึกษาเรื่อง ระบบสืบค้นงานวิจัยทางด้านจิตเวชศาสตร์ โดยใช้ต้นไม้อัดสติใจ กรณีศึกษา สถาบันจิตเวชศาสตร์สมเด็จเจ้าพระยา การศึกษาครั้งนี้ใช้ต้นไม้อัดสติใจช่วยในการสืบค้นงานวิจัย โดยแบ่งเป็นหมวดหมู่ของงานวิจัย ซึ่งใช้การทดสอบแบบกล่องดำ (Black Box) และได้แบ่งการทดสอบเป็น 2 กลุ่ม ได้แก่ กลุ่มผู้ใช้งานทั่วไป และ

กลุ่มผู้เชี่ยวชาญระบบ โดยผู้ใช้งานทั่วไปมีจำนวน 30 คน และผู้เชี่ยวชาญระบบมีจำนวน 10 คน โดยใช้ค่าเบี่ยงเบนมาตรฐาน และ ค่าเฉลี่ย มาใช้ในการวิเคราะห์ข้อมูล จากการทดสอบพบว่า ประสิทธิภาพของระบบ ซึ่งทดสอบจากผู้ใช้งานทั่วไปได้ค่าเบี่ยงเบนมาตรฐานเท่ากับ 0.73 และ ค่าเฉลี่ยเท่ากับ 4.04 สำหรับประสิทธิภาพของระบบ ซึ่งทดสอบจากผู้เชี่ยวชาญระบบได้ ค่าเบี่ยงเบนมาตรฐานเท่ากับ 0.76 และค่าเฉลี่ยเท่ากับ 4.32 จากสถิติทั้งสองค่านี้ที่ได้จากผู้ใช้งาน และผู้เชี่ยวชาญระบบ สรุปได้ว่าระบบสืบค้นงานวิจัยทางด้านจิตเวชศาสตร์ โดยใช้ต้นไม้ตัดสินใจ ที่มีการพัฒนาขึ้นนี้ มีประสิทธิภาพและสามารถนำไปใช้งานได้

พิภพ เต็งศิริวัฒนานนท์ (2552) ศึกษาเรื่อง ระบบฐานข้อมูลอิเล็กทรอนิกส์กลาง โดยใช้ เทคนิคต้นไม้ตัดสินใจช่วยในการค้นหา การศึกษาครั้งนี้ เป็นการศึกษาโดยเก็บรวบรวมข้อมูลจาก พนักงานในบริษัท ออมรอน อิเล็กทรอนิกส์ คอมโพเนนท์ จำกัด เนื่องจากระบบสืบค้นข้อมูลและ ฐานข้อมูลของบริษัท ออมรอน อิเล็กทรอนิกส์ คอมโพเนนท์ จำกัด เกิดความล่าช้าและระบบ ฐานข้อมูลมีระบบการจัดเก็บที่ยังไม่เป็นระบบมากนัก โดยแบ่งกลุ่มตัวอย่างในการเก็บข้อมูลเป็น 2 กลุ่ม คือ กลุ่มผู้ใช้งานทั่วไปและกลุ่มผู้เชี่ยวชาญระบบ โดยผู้ใช้งานทั่วไปมีจำนวน 13 คน และ ผู้เชี่ยวชาญระบบมีจำนวน 5 คน และใช้เทคนิคต้นไม้ตัดสินใจมาช่วยในการจัดทำระบบค้นหาเพื่อ เพิ่มความสะดวกและรวดเร็วในการค้นหาข้อมูลและเพิ่มประสิทธิภาพของระบบ จากผลการ ประเมินประสิทธิภาพของระบบที่ได้ถูกพัฒนาขึ้น พบว่าระบบฐานข้อมูลอิเล็กทรอนิกส์กลางมี ประสิทธิภาพดีและตรงตามความต้องการ

ฉัฐมณต์ สิริวัฒนานนท์ (2551) ศึกษาเรื่อง การตรวจสอบความเหมาะสมในการขนส่ง สินค้าโดยวิธีต้นไม้ตัดสินใจ การศึกษาครั้งนี้ ใช้อัลกอริทึมการเรียนรู้ของต้นไม้ตัดสินใจ C4.5 สร้างแบบจำลองเพื่อตรวจสอบความเหมาะสมในการขนส่งสินค้า เพื่อหาเงื่อนไขในการขนส่ง สินค้าที่เหมาะสม และเพิ่มโอกาสในการแข่งขันกับคู่แข่งอื่นๆ โดยการสร้างแบบจำลองใช้ข้อมูลใน การขนส่งในปี 2551 ในเขตกรุงเทพมหานคร จำนวน 8,316 รายการ และเขตปริมณฑล จำนวน 2,773 รายการ จากการทดสอบแบบจำลองโดยใช้ต้นไม้ตัดสินใจให้ค่าความถูกต้อง 97.23% รวมทั้งจากการนำแบบจำลองไปพัฒนาเป็นระบบตรวจสอบความเหมาะสมในการขนส่งสินค้านั้น พบว่าผลการประเมินระบบนั้น อยู่ในระดับดี โดยได้ค่าเฉลี่ยเท่ากับ 4.67

ไพฑูรย์ จันทรเรือง (2550) ศึกษาเรื่อง ระบบสนับสนุนการตัดสินใจเลือกสาขาการ เรียนของนักศึกษาระดับปริญญาตรี โดยใช้เทคนิคต้นไม้ตัดสินใจ การศึกษาครั้งนี้ใช้ต้นไม้ตัดสินใจ ในการสร้างตัวแบบเพื่อให้ได้ตัวแบบที่สามารถทำนายแนวโน้มของผลการเรียนที่เหมาะสมในแต่ละ สาขา ซึ่งการสร้างตัวแบบนั้นต้องสร้างแยกตัวแบบสำหรับแต่ละสาขาเนื่องจากแต่ละสาขานั้นมี คุณสมบัติของผู้เรียนที่แตกต่างกัน จากคะแนนเฉลี่ยที่ได้นั้นมีคะแนนอยู่ในช่วงกลางของข้อมูล คือ

2.00 – 3.00 ทำให้ผลการตัดสินใจจะโน้มเอียงไปในเกณฑ์ปานกลาง คือ 2.50-2.99 และเกณฑ์พอใช้ คือ 2.00 -2.49

ศักดิ์ชาย ตั้งประเสริฐ (2550) ศึกษาเรื่อง การพัฒนาระบบจำแนกแบบทดสอบสำหรับผู้ทดสอบสุขภาพจิตด้วยเทคนิค Decision Tree ผ่าน Web Application แบบ AJAX การศึกษาครั้งนี้ใช้ต้นไม้ตัดสินใจเพื่อสร้างโมเดลแบบทดสอบที่เหมาะสมกับผู้ทดสอบ โดยนำแบบทดสอบของกรมสุขภาพจิตมาวิเคราะห์และใช้ทฤษฎีความน่าจะเป็นในการหาขนาดของประชากรกลุ่มตัวอย่างที่มีความเชื่อมั่นในระดับ 95% ซึ่งรวบรวมข้อมูลจากประชากรจำนวน 550 คนและใช้การจำแนกข้อมูลแบบต้นไม้ตัดสินใจเพื่อสร้างโมเดล ผลการวิเคราะห์สามารถจำแนกโมเดลได้ทั้งสิ้น 15 โมเดล จากนั้นได้พัฒนาระบบทดสอบแบบออนไลน์ในรูปแบบของเว็บโดยใช้ AJAX ซึ่งผู้ใช้งานจะได้รับแบบทดสอบที่เหมาะสมโดยอิงองค์ความรู้จากโมเดลที่สร้างขึ้น เพื่อป้องกันการคาดเดาในการตอบคำถามที่ผู้ทดสอบหลีกเลี่ยงการแปลผลการทดสอบที่ตนเองไม่พึงประสงค์ เมื่อผู้ทดสอบทำแบบทดสอบเรียบร้อยแล้ว ระบบจะทำการตรวจสอบและแปรผลถึงสภาพจิตใจ รวมทั้งให้คำแนะนำในการปฏิบัติแก่ผู้ทดสอบเช่นกัน

กวีวัฒน์ อ่องล่อ (2554) ศึกษาเรื่อง การวิเคราะห์ระดับความน่าเชื่อถือลูกค้านี้เกษตรกรของมูลนิธิโครงการหลวงด้วยต้นไม้ตัดสินใจ การศึกษาครั้งนี้ใช้ต้นไม้ตัดสินใจในการจำแนกกลุ่มของลูกค้านี้เกษตรกร โดยใช้ข้อมูลในอดีตของเกษตรกร ข้อมูลหนี้สินขยกมาของเกษตรกร ประเภทผลผลิตที่รับซื้อจากเกษตรกร และหนี้สินจากการซื้อขายของเกษตรกร ซึ่งผลการวิเคราะห์ไปเป็นส่วนหนึ่งของการทำงานบนโปรแกรมวิเคราะห์ระดับความน่าเชื่อถือลูกค้านี้เกษตรกรของมูลนิธิโครงการหลวงด้วยต้นไม้ตัดสินใจ จากการทดสอบโปรแกรม พบว่าระบบสามารถทำงานได้ถูกต้อง โดยใช้ข้อมูลจริงบางส่วนของศูนย์พัฒนาโครงการหลวง และเมื่อนำข้อมูลไปใช้ในการวิเคราะห์ระดับความน่าเชื่อถือ พบว่ามีความถูกต้องมากกว่าร้อยละ 50 ซึ่งเป็นระดับความน่าเชื่อถือที่ยอมรับได้ และนำไปประยุกต์ใช้งานได้จริง

สำหรับแนวคิดในงานค้นคว้าอิสระนี้ จะใช้วิธีการค้นหาชื่อจุลินทรีย์โดยใช้ต้นไม้ตัดสินใจ โดยมีการสร้างต้นไม้ก่อนการค้นหา ซึ่งระบบต้องสร้างต้นไม้เพียงครั้งเดียวเท่านั้นและสามารถใช้ต้นไม้ในค้นหาครั้งต่อไป แนวคิดการค้นหาที่พัฒนาขึ้นนี้สามารถทำงานได้รวดเร็ว โดยมีประสิทธิภาพทางเวลาเป็น $O(1)$ เท่านั้น ซึ่งไม่ได้แปรตามจำนวนชื่อจุลินทรีย์ที่มีการเพิ่มขึ้น ดังนั้นจึงสามารถทำงานได้เร็วกว่าการค้นหาแบบ โดยตรงและการค้นหาแบบพบครึ่ง

บทที่ 3

ระเบียบวิธีวิจัย

3.1 ขั้นตอนการดำเนินการวิจัย

ขั้นตอนการดำเนินการวิจัย มีดังต่อไปนี้

1. รวบรวมข้อมูลจูลินทรีย์ที่มีในศูนย์พันธรัฐวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ
2. ศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องต่างๆ ในการทำงานค้นคว้าอิสระนี้ ได้แก่ DOM Tree, XML, AJAX, Tree Traversal, Depth-first Search, Breath-first Search, Decision Tree, PHP, การตั้งชื่อจูลินทรีย์ และการค้นหาแบบต่างๆ
3. เขียนโปรแกรมเพื่อหาจำนวนความแตกต่างของตัวอักษรในชื่อจูลินทรีย์ แบ่งเป็น
 - 3.1.1 ความแตกต่างของตัวอักษรของชื่อสกุลจูลินทรีย์
 - 3.1.2 ความแตกต่างของตัวอักษรของชื่อชนิดจูลินทรีย์ในสกุลเดียวกัน
4. สร้างต้นไม้ตัดสินใจเพื่อใช้ค้นหาชื่อจูลินทรีย์
5. ทดลองค้นหาชื่อจูลินทรีย์จากต้นไม้ตัดสินใจที่สร้างขึ้น
6. วิเคราะห์และสรุปผลการวิจัย

3.2 อุปกรณ์และเครื่องมือที่ใช้ในการวิจัย

- 3.2.1 อุปกรณ์ฮาร์ดแวร์ที่จะนำมาใช้ คือ คอมพิวเตอร์ Acer Intel Core2 Duo
- 3.2.2 ซอฟต์แวร์ที่จะนำมาใช้ คือ Dreamweaver CS3
- 3.2.3 ภาษาที่ใช้ในการเขียนโปรแกรม คือ PHP Version 4

3.3 ระยะเวลาในการดำเนินการวิจัย

ตารางที่ 3.1 ระยะเวลาในการดำเนินการวิจัย

ระยะเวลาดำเนินงาน (เดือน)	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. รวบรวมข้อมูลจูลินทรีย์ที่มี ในศูนย์พันธุวิศวกรรมและ เทคโนโลยีชีวภาพแห่งชาติ	██████████				
2. ศึกษาแนวคิด ทฤษฎี และงานวิจัยที่ เกี่ยวข้อง ได้แก่ DOM Tree, XML, AJAX, Tree Traversal, Depth-first Search, Breath-first Search, Decision Tree, PHP, การตั้งชื่อจูลินทรีย์ และ การค้นหาแบบต่างๆ		██████████			
3. เขียนโปรแกรมเพื่อหาจำนวนความ แตกต่างของตัวอักษรในชื่อจูลินทรีย์			██████████		
4. สร้างต้นไม้ตัดสินใจเพื่อใช้ในการ ค้นหาชื่อจูลินทรีย์			██████████		
5. ทดลองค้นหาชื่อจูลินทรีย์จากต้นไม้ ตัดสินใจที่สร้างขึ้น				██████████	
6. วิเคราะห์และสรุปผลการวิจัย				██████████	

3.4 สรุป

ขั้นตอนในการดำเนินการวิจัย ผู้วิจัยได้แบ่งเป็น 6 ขั้นตอน ได้แก่ รวบรวมข้อมูลของจูลินทรีย์ที่มีจำหน่ายให้กับบุคคลภายนอก ซึ่งมีอยู่ในศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ จากนั้นจึงศึกษาเกี่ยวกับเทคนิคต้นไม้ตัดสินใจเพื่อนำมาค้นหาจูลินทรีย์ได้ง่ายและรวดเร็ว ต่อมา คือ หาจำนวนความแตกต่างของตัวอักษรจากข้อมูลจูลินทรีย์ที่มี โดยแบ่งเป็นความแตกต่างของตัวอักษรของสกุลจูลินทรีย์ และความแตกต่างของตัวอักษรของชนิดในสกุลจูลินทรีย์ ซึ่งผลที่

ได้คือความลึกของต้นไม้ จากนั้นจึงสร้างต้นไม้ตัดสินใจขึ้นเพื่อแสดงความเร็วในการค้นหา โดยแสดงเป็นความลึกของต้นไม้ จากนั้นทดลองค้นหาชื่ออุลทินทรีย์จากต้นไม้ที่สร้างขึ้นและสุดท้ายทำการวิเคราะห์และสรุปผลการวิจัย

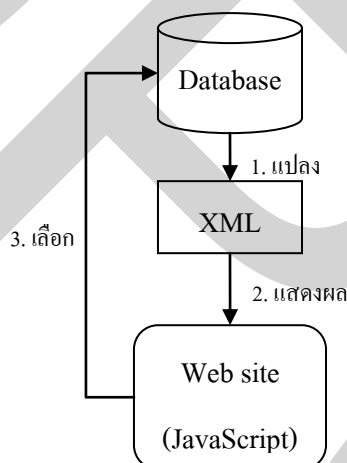


บทที่ 4

ผลการวิเคราะห์และออกแบบระบบ

เนื้อหาของบทนี้ กล่าวถึงภาพรวมของระบบ แนวคิดในการค้นหาชื่อจูลินทรีย์จาก
ต้นไม้ตัดสินใจ และ การออกแบบระบบ

4.1 ภาพรวมของระบบ



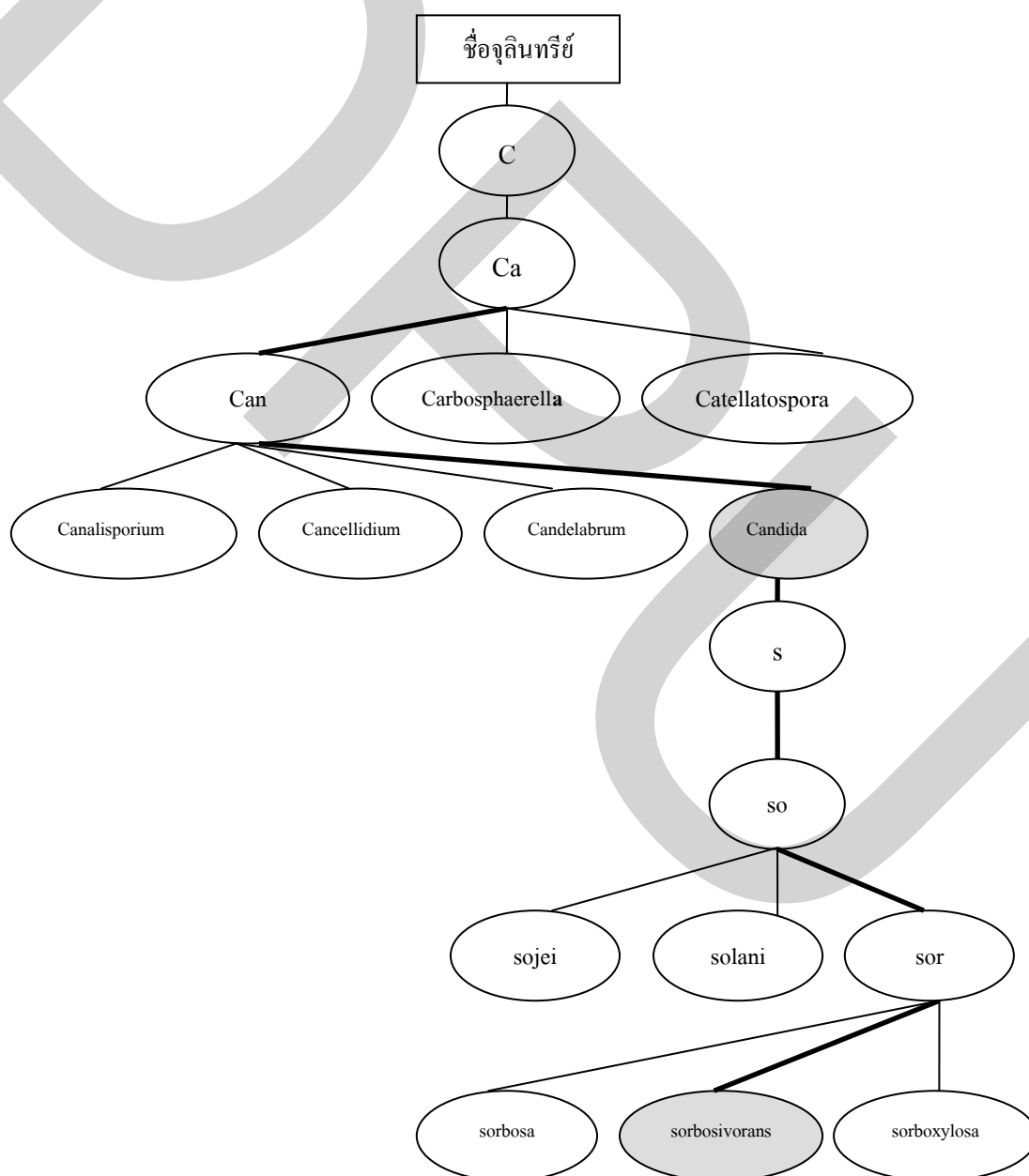
ภาพที่ 4.1 ภาพรวมของเว็บไซต์ค้นหาชื่อจูลินทรีย์

การทำงานโดยรวมของเว็บไซต์ค้นหาชื่อจูลินทรีย์ ระบบจะต้องมีการแปลงฐานข้อมูล
ชื่อจูลินทรีย์เป็น ไฟล์ XML และไฟล์ XML นี้จะถูกเรียกใช้งานจากเว็บไซต์เพื่อค้นหาจูลินทรีย์
ชนิดต่างๆ โดยใช้ภาษา Java script ซึ่งไฟล์ XML ชื่อจูลินทรีย์ สามารถมองในรูปของ DOM Tree
ซึ่งมีการแตกเป็น โหนดลูกถ้าตัวอักษรในชื่อจูลินทรีย์ต่างกัน

4.2 แนวคิดในการค้นหาชื่อจุลินทรีย์

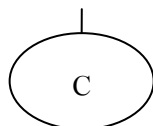
การค้นหาชื่อจุลินทรีย์นี้ ได้ใช้เทคนิคต้นไม้ตัดสินใจเข้ามาใช้ในการค้นหา โดยเทคนิคนี้ให้ผลลัพธ์ที่รวดเร็วและแม่นยำ ซึ่งการค้นหาชื่อจุลินทรีย์ตามแผนภาพต้นไม้ตัดสินใจ จะเป็นการค้นหาโดยต้องเรียงชื่อจุลินทรีย์ตามลำดับตัวอักษรตามพจนานุกรมแล้วจึงนำมาสร้างต้นไม้ตัดสินใจ ดังตัวอย่าง

ตัวอย่าง การค้นหาจุลินทรีย์ *Candida sorbosivorans*



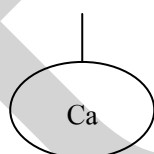
ภาพที่ 4.2 DOM Tree ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 4.2 คือตัวอย่างของ DOM Tree ในการค้นหาชื่อจุลินทรีย์ *Candida sorbosivorans* DOM Tree นี้แบ่งเป็น 8 ระดับ ดังนี้



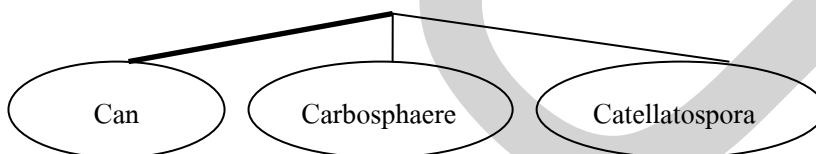
ภาพที่ 4.3 DOM Tree ระดับที่ 1 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 4.3 การค้นหาชื่อจุลินทรีย์จะเริ่มจากตัวอักษรแรกของชื่อจุลินทรีย์ที่ผู้ใช้พิมพ์เข้ามา ในที่นี้คือ ตัวอักษร C



ภาพที่ 4.4 DOM Tree ระดับที่ 2 ของจุลินทรีย์ *Candida sorbosivorans*

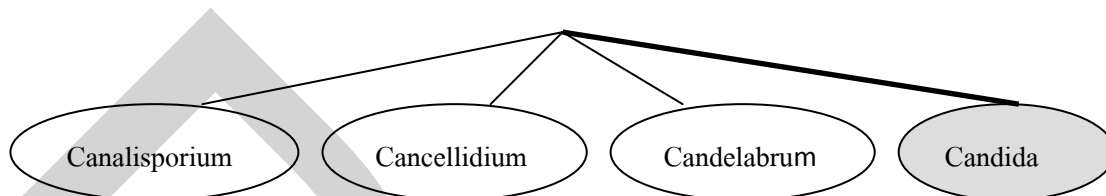
จากภาพที่ 4.4 เมื่อค้นหาจากตัวอักษรแรกแล้ว จะค้นหาจากตัวอักษรถัดมา คือ ตัวอักษร a จึงเป็นการค้นหาจากชื่อจุลินทรีย์ที่ขึ้นต้นด้วย Ca



ภาพที่ 4.5 DOM Tree ระดับที่ 3 ของจุลินทรีย์ *Candida sorbosivorans*

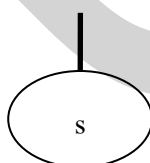
จากภาพที่ 4.5 ชื่อสกุลจุลินทรีย์ที่ขึ้นต้นด้วย Ca มีด้วยกันทั้งหมด 6 สกุล ซึ่งมีความต่างของตัวอักษรถัดมา คือ ตัวอักษร n, ตัวอักษร r และตัวอักษร t ดังจะเห็นจากภาพ คือ ตัวอักษร r

และ ตัวอักษร t นั้น มีสกุลอย่างละ 1 สกุล จึงไม่ต้องแตกโหนดใดๆ ส่วน ตัวอักษร n นั้น มีทั้งหมด 4 สกุล จึงต้องแตกเป็น 4 โหนด



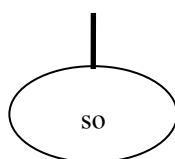
ภาพที่ 4.6 DOM Tree ระดับที่ 4 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 4.6 สกุลของจุลินทรีย์ที่ขึ้นต้นด้วยตัวอักษร Can นั้น แบ่งเป็น 4 สกุล จึงแตกเป็น 4 โหนด คือ Canalisporium, Cancellidium, Candelabrum และ Candida จากตัวอย่างที่ต้องการค้นหา คือ สกุล Candida ซึ่งพบชื่อสกุลในระดับที่ 4 นี้



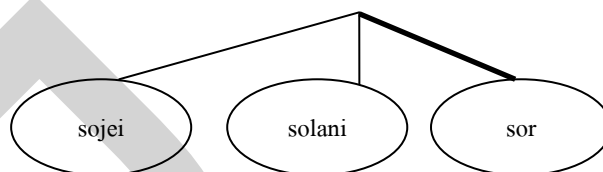
ภาพที่ 4.7 DOM Tree ระดับที่ 5 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 4.7 เมื่อค้นหาเจอสกุลของจุลินทรีย์แล้ว จึงได้ค้นหาชนิดของจุลินทรีย์ในสกุล ซึ่งจากตัวอย่างที่ต้องการค้นหา คือ ชนิด sorbosivorans จากโหนดของชื่อสกุลที่พบนั้น จะแตกโหนดเป็นชนิดของสกุล ซึ่งจะค้นหาจากตัวอักษรแรกของชื่อชนิดของจุลินทรีย์ นั่นคือ ตัวอักษร s



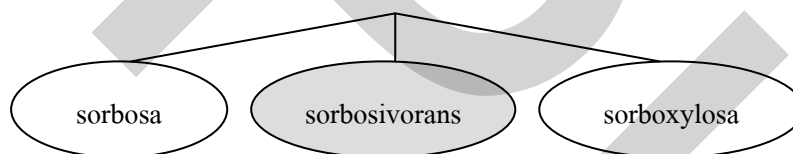
ภาพที่ 4.8 DOM Tree ระดับที่ 6 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 4.8 เมื่อค้นหาจากตัวอักษรแรกแล้ว จะค้นหาจากตัวอักษรถัดมา คือ ตัวอักษร o จึงเป็นการค้นหาจากชื่อชนิดจุลินทรีย์ที่ขึ้นต้นด้วย so



ภาพที่ 4.9 DOM Tree ระดับที่ 7 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 4.9 ชื่อชนิดจุลินทรีย์ที่ขึ้นต้นด้วย so มีด้วยกันทั้งหมด 5 สกุล ซึ่งมีความต่างของตัวอักษรถัดมา คือ ตัวอักษร j, ตัวอักษร l และตัวอักษร r ดังจะเห็นจากภาพ คือ ตัวอักษร j และตัวอักษร l นั้น มีชนิดอย่างละ 1 ชนิด จึงไม่ต้องแตกโหนดใดๆ ส่วน ตัวอักษร r นั้น มีทั้งหมด 3 ชนิด จึงต้องแตกเป็น 3 โหนด



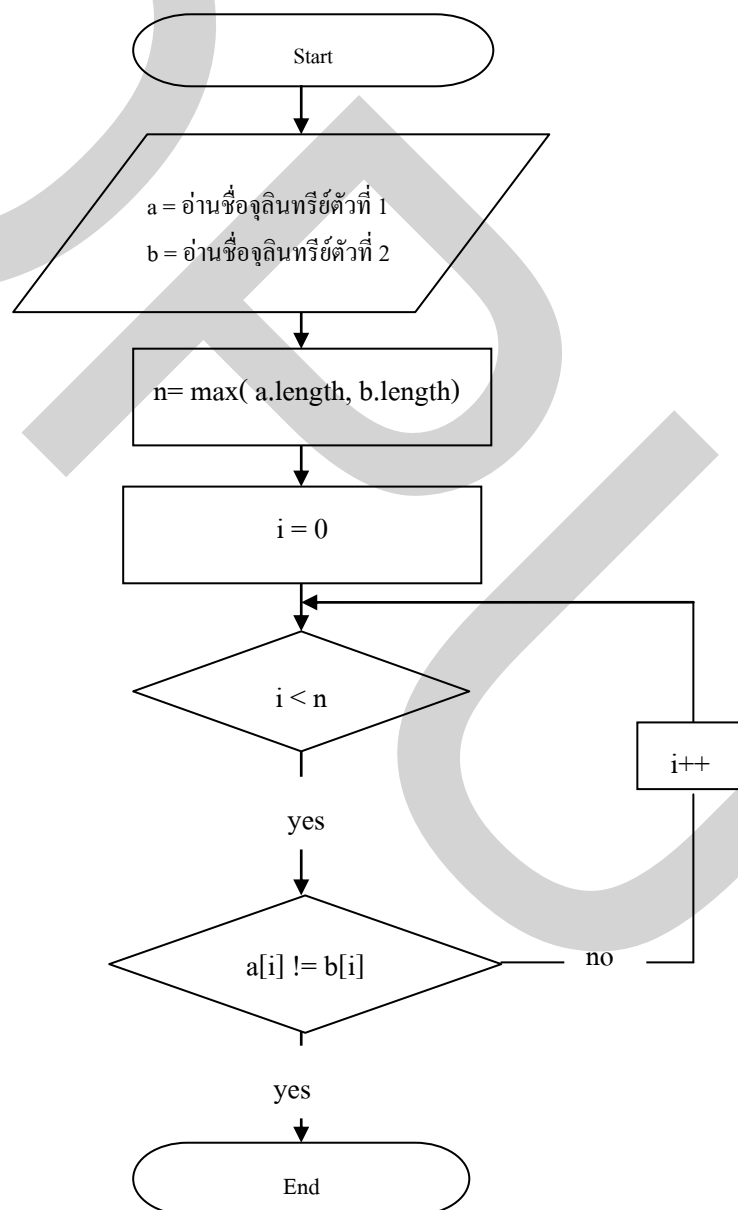
ภาพที่ 4.10 DOM Tree ระดับที่ 8 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 4.10 ชนิดของจุลินทรีย์ที่ขึ้นต้นด้วย sor นั้น แตกเป็น 3 โหนด ดังภาพ ซึ่งจากระดับที่ 8 นี้ จะพบชนิดของจุลินทรีย์ในสกุล *Candida* ที่ต้องการค้นหา คือ ชนิด *sorbosivorans*

จาก DOM Tree ในภาพที่ 4.2 เมื่อเราต้องการค้นหาชื่อจุลินทรีย์ในต้นไม้ นี้ จะใช้วิธีการค้นหาแบบเดียวกับการค้นหาคำตอบในต้นไม้ตัดสินใจ ดังนั้นประสิทธิภาพในการค้นหาชื่อจุลินทรีย์จะขึ้นอยู่กับความลึกของต้นไม้ ยิ่งความลึกของต้นไม้ น้อยเท่าไร การค้นหาชื่อจุลินทรีย์ จะมีความรวดเร็วตามไปด้วย

4.3 การออกแบบระบบ

แนวคิดการค้นหาชื่อจูลินทรีย์ที่พัฒนาขึ้น มีการออกแบบ DOM Tree ให้สามารถค้นหาได้รวดเร็ว โดยหาจากตำแหน่งตัวอักษรที่แตกต่างกันของจูลินทรีย์ ซึ่งแยกเป็น 2 ส่วน คือ ตำแหน่งตัวอักษรที่แตกต่างกันของสกุลจูลินทรีย์ และ ตำแหน่งตัวอักษรที่แตกต่างกันของชนิดในสกุลจูลินทรีย์ และมีการเขียนโปรแกรมเข้ามาช่วยในการค้นหาตำแหน่งตัวอักษรที่แตกต่างกันของสกุลจูลินทรีย์และชนิดในสกุลจูลินทรีย์ ดัง flow chart ต่อไปนี้



ภาพที่ 4.11 Flow chart แสดงขั้นตอนการทำงานของโปรแกรม

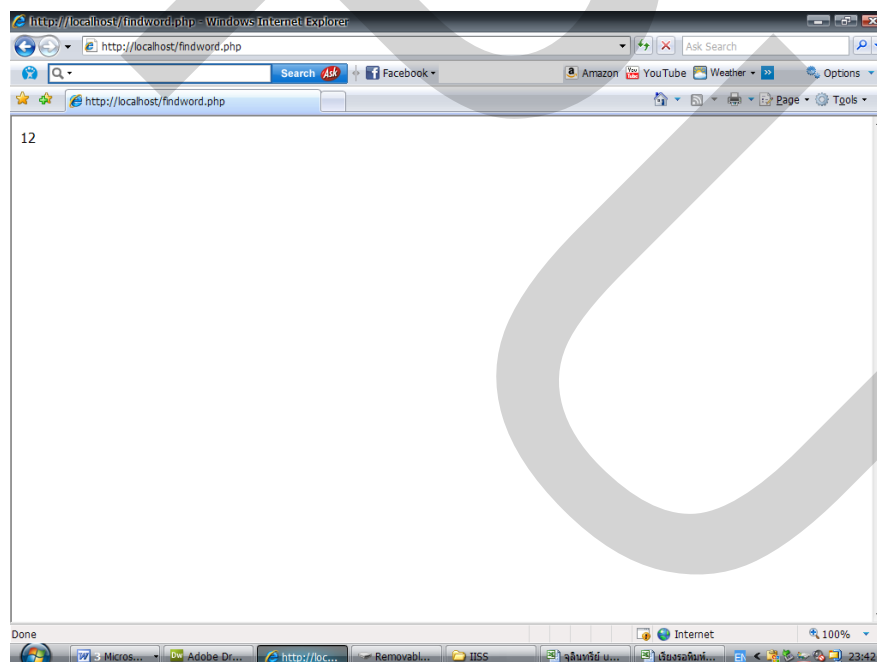
```

1 <?
2 $a='Dactylospora';
3 $b='Dactylosporangium';
4 $n= max($a.length,$b.length);
5
6 for($i=0;$i<$n;$i++)
7 {
8     if($a[$i]!==$b[$i])
9     {
10        break;
11    }
12 }
13 echo($i);
14 ?>
15

```

The screenshot shows the Adobe Dreamweaver CS3 interface with a code view of a file named 'findword.php'. The code is a PHP script that compares two strings, 'Dactylospora' and 'Dactylosporangium', character by character. It finds the first index where the characters differ and outputs that index. The code is as follows:

ภาพที่ 4.12 โค้ดโปรแกรมการหาตำแหน่งของตัวอักษรที่แตกต่างของสกุลลินทรีย์



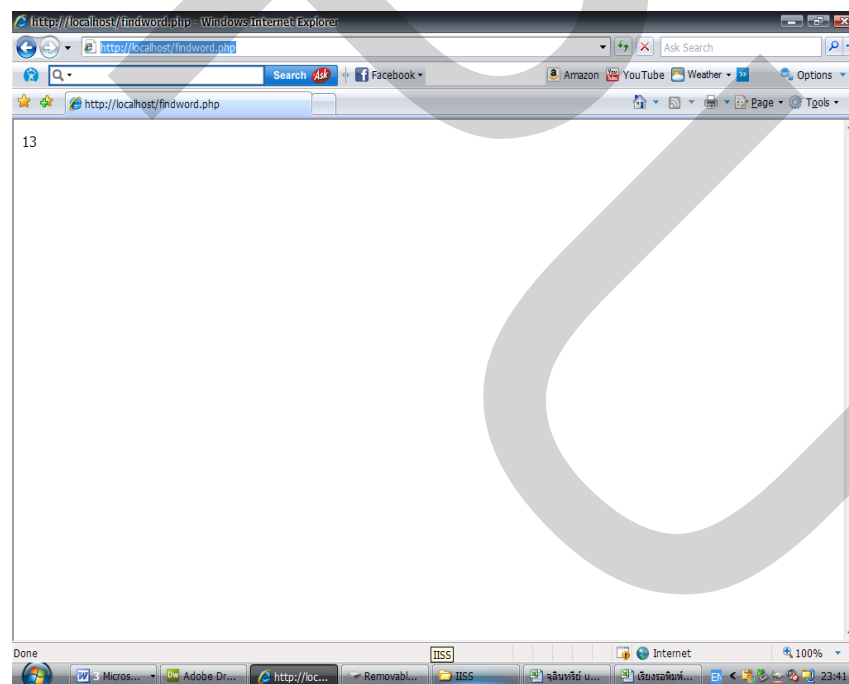
ภาพที่ 4.13 หน้าจอแสดงผลการหาตำแหน่งของตัวอักษรที่แตกต่างของสกุลลินทรีย์

```

1 <?
2 $a='saturnus var. mrakii';
3 $b='saturnus var. saturnus!';
4 $n= max($a.length,$b.length);
5
6 for($i=0;$i<$n;$i++)
7 {
8     if($a[$i]!=$b[$i])
9     {
10        break;
11    }
12 }
13 echo($i);
14 ?>
15

```

ภาพที่ 4.14 โค้ดโปรแกรมการหาตำแหน่งตัวอักษรที่แตกต่างของชนิดในสกุลจูลินทรีย์



ภาพที่ 4.15 หน้าจอผลการหาตำแหน่งตัวอักษรที่แตกต่างของชนิดในสกุลจูลินทรีย์

ผลการทดสอบโปรแกรม พบว่า ส่วนของชื่อสกุลจุนทรีย์มีตำแหน่งที่แตกต่างกันของตัวอักษรมากที่สุดในตำแหน่งที่ 12 และ ชนิดในสกุลจุนทรีย์มีตำแหน่งที่แตกต่างกันของตัวอักษรมากที่สุดในตำแหน่งที่ 13 ดังนั้น DOM Tree ของชื่อจุนทรีย์ที่ได้จากฐานข้อมูลชื่อจุนทรีย์จะมีความลึกไม่เกิน $12+13 = 25$ ระดับ



บทที่ 5

ผลการจัดทำและการทดสอบระบบ

เนื้อหาในบทที่ 5 นี้ เป็นการทดสอบการค้นหาชื่อจุลินทรีย์ โดยใช้ต้นไม้ตัดสินใจในการค้นหา ซึ่งเป็นการทดสอบจากชื่อจุลินทรีย์ที่สุ่มขึ้นมาทดสอบทั้งหมด 30 ชนิด ที่มีความยาวของตัวอักษรที่แตกต่างกัน ตั้งแต่ 16 ถึง 40 ตัวอักษร

5.1 ตารางชื่อจุลินทรีย์

การทดสอบการค้นหาชื่อจุลินทรีย์นี้จะสุ่มชื่อจุลินทรีย์มาทดสอบจำนวน 30 ชนิด ซึ่งจะมีความยาวของตัวอักษรที่แตกต่างกัน ดังแสดงในตารางที่ 5.1

ตารางที่ 5.1 ชื่อจุลินทรีย์ที่ใช้ในการทดสอบ

ชื่อจุลินทรีย์	ความยาวของตัวอักษร
Acetobacter aceti	16
Micrococcus luteus	17
Exobasidium vexans	17
Bacillus megaterium	18
Trichosporiella sp.	18
Candida sorbosivorans	20
Debaryomyces siamensis	21
Streptosporangium album	22
Enterococcus raffinosus	22
Bensingtonia thailandica	23
Annulohyphoxylon atroseum	25

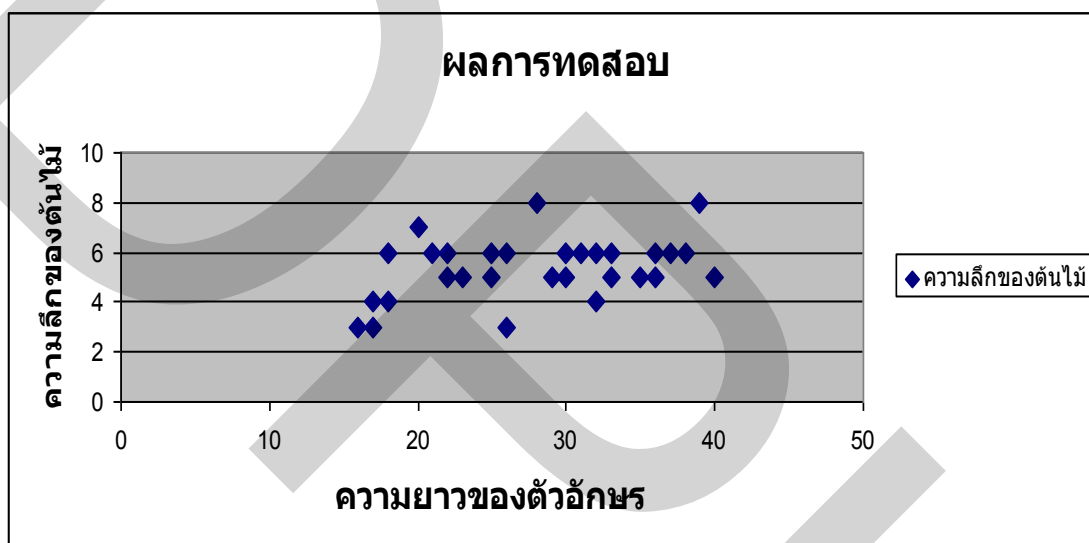
ตารางที่ 5.1 (ต่อ)

ชื่อจุลินทรีย์	ความยาวของตัวอักษร
<i>Ophiocordyceps irangiensis</i>	25
<i>Millerozyma phetchabunensis</i>	26
<i>Gluconacetobacter europaeus</i>	26
<i>Akanthomyces pistillariiformis</i>	29
<i>Trichosporon mycotoxinivorans</i>	28
<i>Kibdelosporangium philippinense</i>	30
<i>Gluconobacter kanchanaburiensis</i>	30
<i>Pseudoclathrosphaerina evamariae</i>	31
<i>Carbosphaerella leptosphaerioides</i>	32
<i>Gluconacetobacter sucrofermentans</i>	32
<i>Dioszegia zsołtii</i> var. <i>yunnanensis</i>	33
<i>Hanseniaspora pseudoguilliermondii</i>	33
<i>Debaryomyces vanrijiae</i> var. <i>yarrowii</i>	35
<i>Halophytophthora spinosa</i> var. <i>Spinosa</i>	36
<i>Biscogniauxia nummularia</i> var. <i>exutans</i>	36
<i>Williopsis saturnus</i> var. <i>subsufficiens</i>	37
<i>Debaryomyces polymorphus</i> var. <i>Africanus</i>	38
<i>Lactobacillus plantarum</i> subsp. <i>plantarum</i>	39
<i>Biscogniauxia citrifomis</i> var. <i>Macrospora</i>	40

จากตารางที่ 5.1 แสดงรายชื่อจุลินทรีย์ที่ใช้ในการค้นหา ซึ่งเป็นการค้นหาความลึกของต้นไม้ เพื่อแสดงถึงระดับในการค้นหาชื่อจุลินทรีย์

5.2 ผลการทดสอบระบบ

จากผลการทดสอบเพื่อหาความลึกของต้นไม้ โดยใช้ชื่อจูลินทรีย์ที่มีความยาวต่างกัน ทั้งหมด 30 ชนิด ในการค้นหา ดังแสดงในตารางที่ 5.1 พบว่า ความลึกของต้นไม้ในการค้นหาจูลินทรีย์ที่ทดสอบมีความลึกอยู่ระหว่าง 3 - 8 ระดับเท่านั้น และเป็นไปตามแนวคิดที่ว่าความลึกจะน้อยกว่าความสูงของต้นไม้ซึ่งจะไม่เกิน 25 ระดับ ดังนั้นการใช้ต้นไม้ DOM Tree ในการค้นหาชื่อจูลินทรีย์สามารถทำได้อย่างรวดเร็ว ถึงแม้ว่าจะมีชื่อจูลินทรีย์ในฐานข้อมูลหลายพันชนิดก็ตาม



ภาพที่ 5.1 ผลการทดสอบการหาความลึกของต้นไม้

5.3 วิเคราะห์ผลการทดสอบ

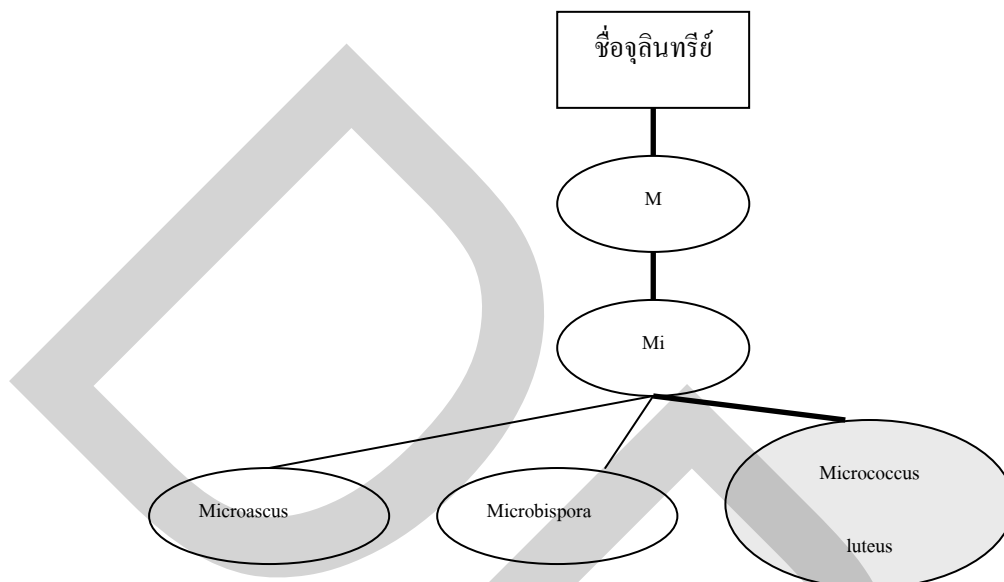
จากรายชื่อจูลินทรีย์ที่นำมาทดสอบ สามารถนำมาสร้างต้นไม้ตัดสินใจในการค้นหาชื่อจูลินทรีย์ โดยแบ่งเป็น 3 กรณี ต่อไปนี้

5.3.1 กรณีที่พบชื่อจูลินทรีย์ทันที

5.3.2 กรณีชื่อสกุลมีตัวอักษรที่ซ้ำกันและสามารถพบชนิดของจูลินทรีย์ในสกุลนั้นได้ทันที

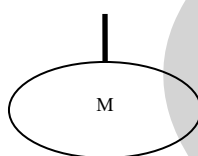
5.3.3 กรณีชื่อสกุลมีตัวอักษรที่ซ้ำกันและชนิดของจูลินทรีย์ในสกุลนั้นมีตัวอักษรที่ซ้ำกัน

ตัวอย่างที่ 1 การค้นหาจุลินทรีย์ *Micrococcus luteus* (กรณีพบชื่อจุลินทรีย์ทันที)



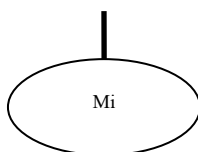
ภาพที่ 5.2 DOM Tree ของการค้นหาจุลินทรีย์ *Micrococcus luteus*

จากภาพที่ 5.2 คือ DOM Tree ในการค้นหาชื่อจุลินทรีย์ *Micrococcus luteus* ซึ่ง DOM Tree นี้แบ่งเป็น 3 ระดับ ดังนี้



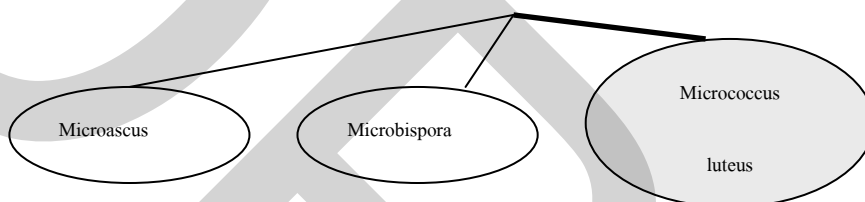
ภาพที่ 5.3 DOM Tree ระดับที่ 1 ของจุลินทรีย์ *Micrococcus luteus*

จากภาพที่ 5.3 จากชื่อจุลินทรีย์ *Micrococcus luteus* จะเริ่มค้นหาจากตัวอักษรแรกของชื่อจุลินทรีย์ นั่นคือ ตัวอักษร M



ภาพที่ 5.4 DOM Tree ระดับที่ 2 ของจุลินทรีย์ *Micrococcus luteus*

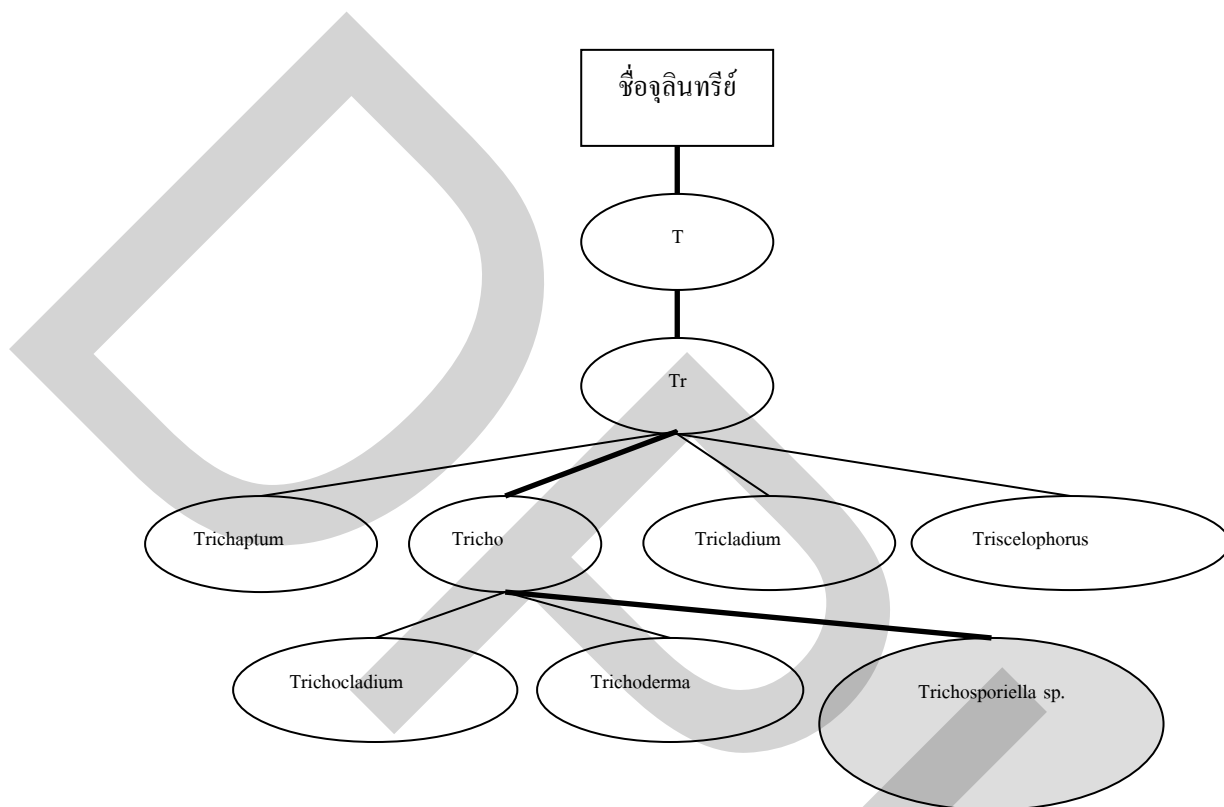
จากภาพที่ 5.4 เมื่อค้นหาจากตัวอักษรแรกแล้ว จะค้นหาจากตัวอักษรถัดมา คือ ตัวอักษร *i* จึงเป็นการค้นหาจากชื่อจุลินทรีย์ที่ขึ้นต้นด้วย *Mi*



ภาพที่ 5.5 DOM Tree ระดับที่ 3 ของจุลินทรีย์ *Micrococcus luteus*

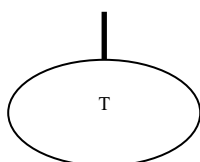
จากภาพที่ 5.5 ชื่อสกุลจุลินทรีย์ที่ขึ้นต้นด้วย *Mi* มีหลายสกุล ต้นไม้ไม่มีการแตกเป็นหลาย โหนด เพื่อให้ง่ายต่อการค้นหา และพบชื่อสกุลจุลินทรีย์ *Micrococcus* ที่ระดับที่ 3 นี้ สำหรับชนิดของจุลินทรีย์ในสกุลนี้ มีเพียงชนิดเดียว คือ *luteus* จึงนำมารวมกับ โหนด *Micrococcus* ซึ่งเป็นชื่อสกุลจุลินทรีย์นี้ได้ทันที

ตัวอย่างที่ 2 การค้นหาจุลินทรีย์ Trichosporiella sp. (กรณีชื่อสกุลมีตัวอักษรที่ซ้ำกันและสามารถพบชนิดของจุลินทรีย์ในสกุลนั้นได้ทันที)



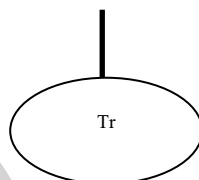
ภาพที่ 5.6 DOM Tree ของการค้นหาจุลินทรีย์ Trichosporiella sp.

จากภาพที่ 5.6 คือ DOM Tree ในการค้นหาชื่อจุลินทรีย์ Trichosporiella sp. ซึ่ง DOM Tree นี้แบ่งเป็น 4 ระดับ ดังนี้



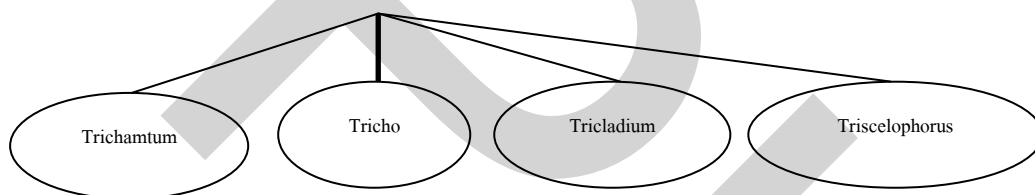
ภาพที่ 5.7 DOM Tree ระดับที่ 1 ของจุลินทรีย์ Trichosporiella sp.

จากภาพที่ 5.7 จากชื่อจุลินทรีย์ *Trichosporiella* sp. จะเริ่มค้นหาจากตัวอักษรแรกของชื่อจุลินทรีย์ นั่นคือ ตัวอักษร T



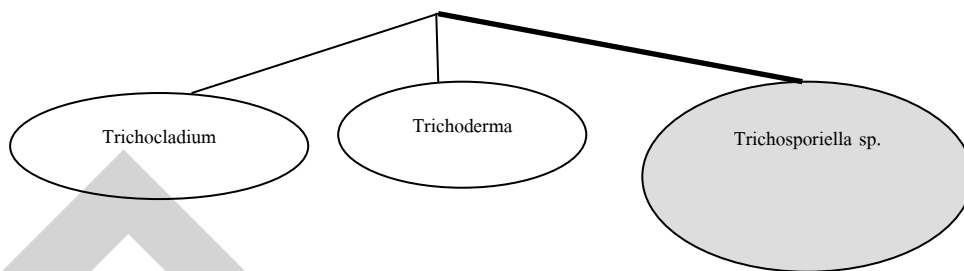
ภาพที่ 5.8 DOM Tree ระดับที่ 2 ของจุลินทรีย์ *Trichosporiella* sp.

จากภาพที่ 5.8 เมื่อค้นหาจากตัวอักษรแรกแล้ว จะค้นหาจากตัวอักษรถัดมา คือ ตัวอักษร r จึงเป็นการค้นหาจากชื่อจุลินทรีย์ที่ขึ้นต้นด้วย Tr



ภาพที่ 5.9 DOM Tree ระดับที่ 3 ของจุลินทรีย์ *Trichosporiella* sp.

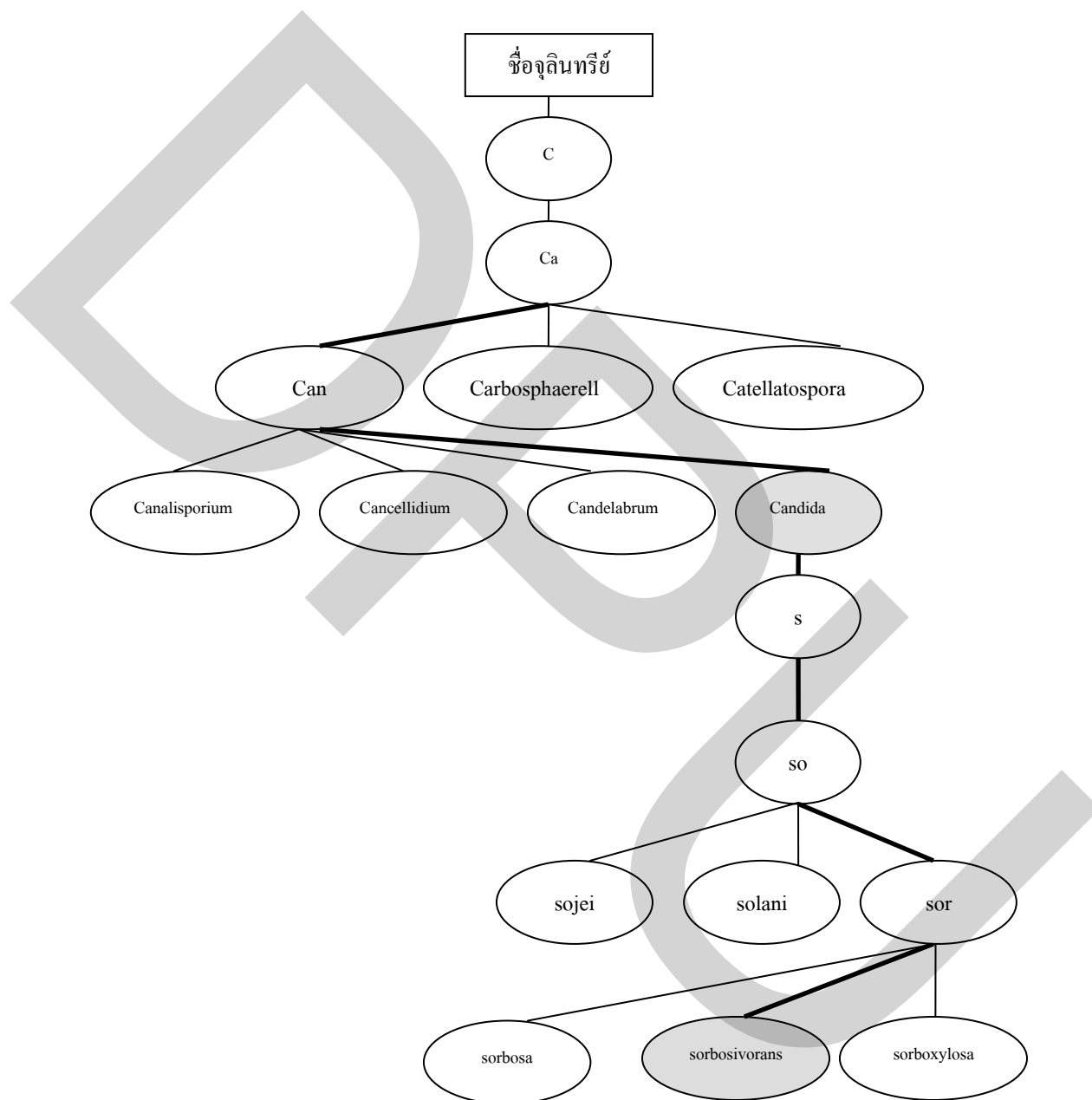
จากภาพที่ 5.9 จากชื่อสกุล *Trichosporiella* ซึ่งขึ้นต้นด้วย Tr และมีตัวอักษรถัดมาคือตัวอักษร i โดยสกุลที่มีชื่อสกุลขึ้นต้นด้วย Tri มีหลากหลายสกุล จึงทำการแตกเป็นหลายโหนด และพบว่าตัวอักษรที่มีลักษณะตามชื่อสกุล มีความแตกต่างกันของตัวอักษรอยู่มาก จึงสร้างโหนด Tricho ขึ้น เพื่อทำการแตกโหนดไปยังชื่อสกุล เพื่อให้ง่ายต่อการค้นหา



ภาพที่ 5.10 DOM Tree ระดับที่ 4 ของจุลินทรีย์ Trichosporiella sp.

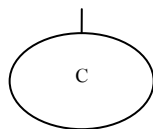
จากภาพที่ 5.10 เมื่อแตกโหนดจากโหนด Tricho พบชื่อสกุลจุลินทรีย์ Trichosporiella ที่ระดับที่ 4 นี้ สำหรับชนิดของจุลินทรีย์ในสกุลนี้มีเพียงชนิดเดียว คือ sp. จึงนำมารวมกับโหนด Trichosporiella ซึ่งเป็นชื่อสกุลนี้ได้ทันที

ตัวอย่างที่ 3 การค้นหาจุลินทรีย์ *Candida sorbosivorans* (กรณีชื่อสกุลมีตัวอักษรที่ซ้ำกันและชนิดของจุลินทรีย์ในสกุลนั้นมีตัวอักษรที่ซ้ำกัน)



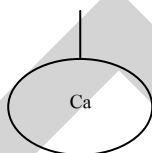
ภาพที่ 5.11 DOM Tree ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 5.11 คือตัวอย่างของ DOM Tree ในการค้นหาชื่อจุลินทรีย์ *Candida sorbosivorans* DOM Tree นี้แบ่งเป็น 7 ระดับ ดังนี้



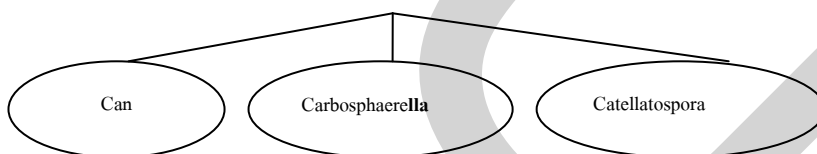
ภาพที่ 5.12 DOM Tree ระดับที่ 1 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 5.12 การค้นหาชื่อจุลินทรีย์จะเริ่มจากตัวอักษรแรกของชื่อจุลินทรีย์ นั่นคือตัวอักษร C



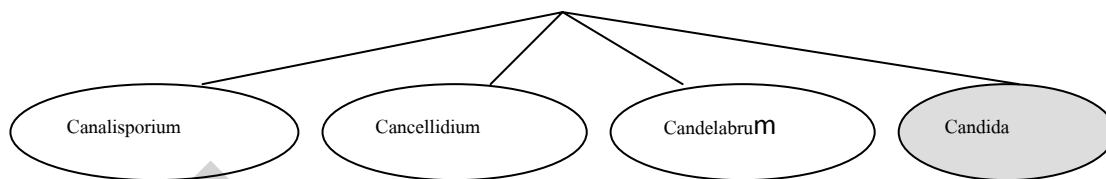
ภาพที่ 5.13 DOM Tree ระดับที่ 2 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 5.13 เมื่อค้นหาจากตัวอักษรแรกแล้ว จะค้นหาจากตัวอักษรถัดมา คือตัวอักษร a จึงเป็นการค้นหาจากชื่อจุลินทรีย์ที่ขึ้นต้นด้วย Ca



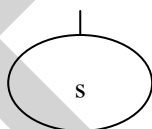
ภาพที่ 5.14 DOM Tree ระดับที่ 3 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 5.14 ชื่อสกุลจุลินทรีย์ที่ขึ้นต้นด้วย Ca มีด้วยกันทั้งหมด 6 สกุล ซึ่งมีความต่างของตัวอักษรถัดมา คือ ตัวอักษร n, ตัวอักษร r และตัวอักษร t ดังจะเห็นจากภาพ คือ ตัวอักษร r และ ตัวอักษร t นั้น มีสกุลอย่างละ 1 สกุล จึงไม่ต้องแตกโหนดใดๆ ส่วน ตัวอักษร n นั้น มีทั้งหมด 4 สกุล จึงต้องแตกเป็น 4 โหนด



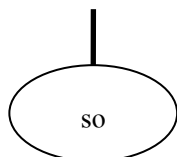
ภาพที่ 5.15 DOM Tree ระดับที่ 4 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 5.15 สกวลจุลินทรีย์ที่ขึ้นต้นด้วยตัวอักษร Can นั้น แบ่งเป็น 4 สกวล จึงแตกเป็น 4 โหนด คือ Canalisporium, Cancellidium, Candelabrum และ Candida จากตัวอย่างที่ต้องการค้นหา คือ สกวล Candida ซึ่งพบชื่อสกวลระดับที่ 4 นี้



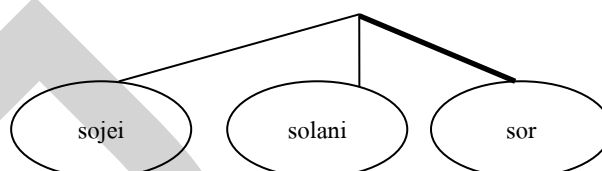
ภาพที่ 5.16 DOM Tree ระดับที่ 5 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพ ที่ 5.16 เมื่อค้นหาเจอสกวลของจุลินทรีย์แล้ว จึงได้ค้นหาชนิดของจุลินทรีย์ในสกวล ซึ่งจากตัวอย่างที่ต้องการค้นหา คือ ชนิด sorbosivorans จากโหนดของชื่อสกวลที่พบนั้น จะแตกโหนดเป็นชนิดของสกวล ซึ่งจะค้นหาจากตัวอักษรแรกของชื่อชนิดของจุลินทรีย์ นั่นคือตัวอักษร s



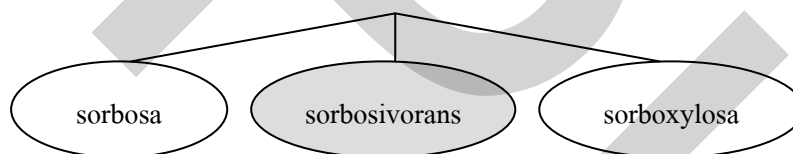
ภาพที่ 5.17 DOM Tree ระดับที่ 6 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 5.17 เมื่อค้นหาจากตัวอักษรแรกแล้ว จะค้นหาจากตัวอักษรถัดมา คือ ตัวอักษร o จึงเป็นการค้นหาจากชื่อชนิดจุลินทรีย์ที่ขึ้นต้นด้วย so



ภาพที่ 5.18 DOM Tree ระดับที่ 7 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 5.18 ชื่อชนิดจุลินทรีย์ที่ขึ้นต้นด้วย so มีด้วยกันทั้งหมด 5 สกุล ซึ่งมีความต่างของตัวอักษรถัดมา คือ ตัวอักษร j, ตัวอักษร l และตัวอักษร r ดังจะเห็นจากภาพ คือ ตัวอักษร j และ ตัวอักษร l นั้น มีชนิดอย่างละ 1 ชนิด จึงไม่ต้องแตกโหนดใดๆ ส่วน ตัวอักษร r นั้น มีทั้งหมด 3 ชนิด จึงต้องแตกเป็น 3 โหนด



ภาพที่ 5.19 DOM Tree ระดับที่ 8 ของจุลินทรีย์ *Candida sorbosivorans*

จากภาพที่ 5.19 ชนิดของจุลินทรีย์ที่ขึ้นต้นด้วย sor นั้น แตกเป็น 3 โหนด ดังภาพ ซึ่งจากระดับที่ 8 นี้ จะพบชนิดของจุลินทรีย์ในสกุล *Candida* ที่ต้องการค้นหา คือ ชนิด *sorbosivorans*

5.4 การเปรียบเทียบประสิทธิภาพของต้นไม้ตัดสินใจ

งานค้นคว้าอิสระเล่มนี้ เสนอแนวคิดในการค้นหาชื่อจุลินทรีย์โดยใช้ต้นไม้ตัดสินใจ ซึ่งมีการเรียงลำดับรายชื่อจุลินทรีย์ก่อนการค้นหา คล้ายกับการค้นหาแบบพบบครั้ง ซึ่งต้องมีการเรียงลำดับข้อมูลก่อน โดยประสิทธิภาพทางเวลาของแนวคิดนี้ คือ $O(1)$ เท่านั้น นั่นคือระดับของต้นไม้ที่ทำได้ (25 ระดับ) การค้นหาสามารถค้นหาได้รวดเร็วเนื่องจากไม่ได้แปรตามจำนวนชื่อจุลินทรีย์ หรือ หากมีชื่อจุลินทรีย์เพิ่มขึ้นมา ก็จะต้องสร้างต้นไม้ตัดสินใจขึ้นมาใหม่ แต่ระดับของ

การค้นหาค่าจะไม่เกิน 25 ระดับ หรือ หากข้อมูลอินทรีย์ที่ต้องการค้นหาอยู่รายชื่อสุดท้ายในลิสต์ของการเรียงลำดับ การค้นหาก็จะน้อยกว่าหรือไม่เกิน 25 ระดับ

ตารางที่ 5.2 ตารางเปรียบเทียบกับการค้นหาประสิทธิภาพแบบอื่น

การค้นหาแบบ	ประสิทธิภาพ	มีการเรียงลำดับข้อมูลก่อน
แบบต้นไม้ตัดสินใจ (Decision Tree)	$O(1)$	ใช่
แบบพื้ครึ่ง (Binary Search)	$O(\log_2 N)$	ใช่
แบบเชิงเส้น (Linear Search)	$O(N)$	ไม่ใช่

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

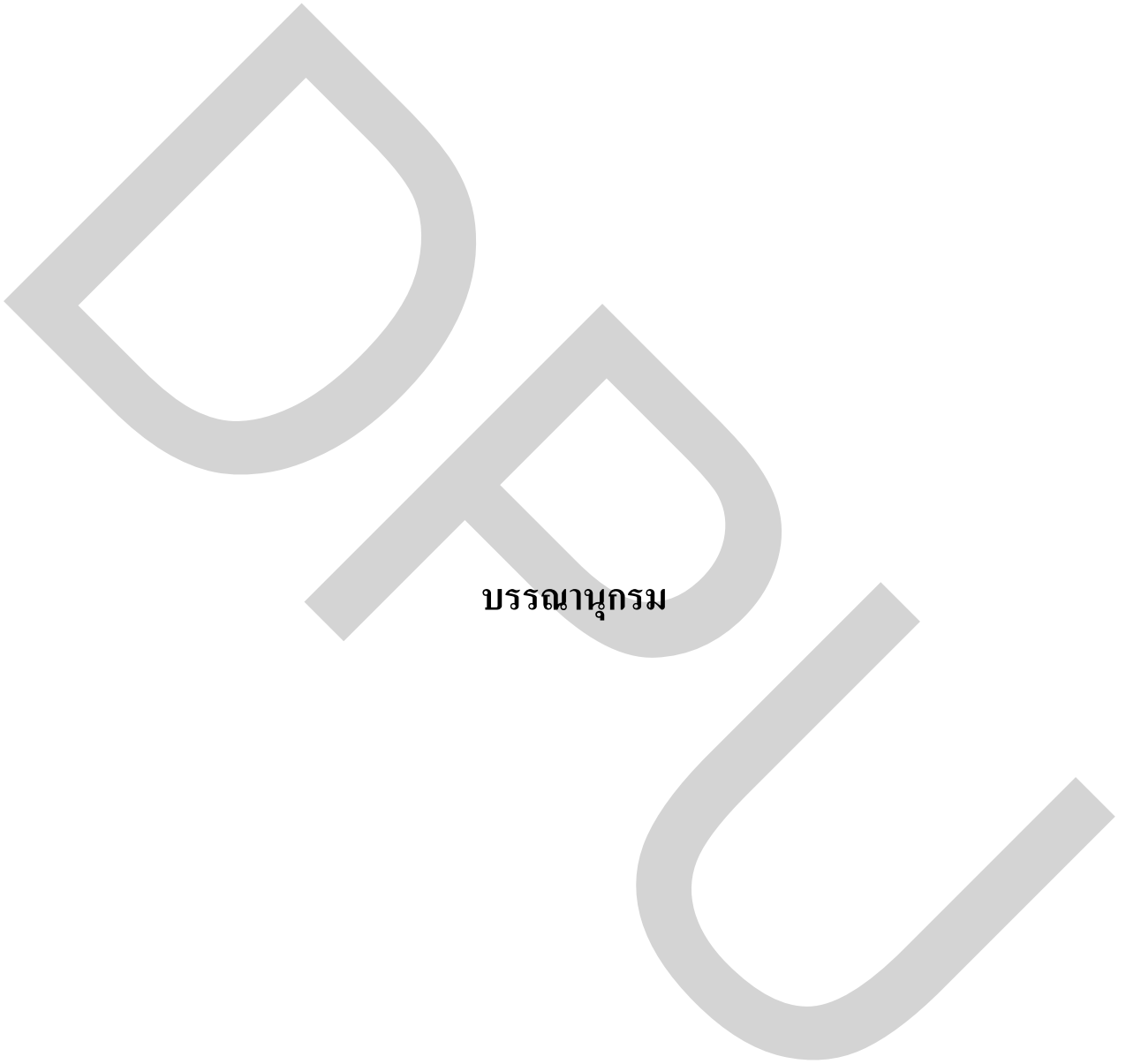
จูลินทรีย์มีหลากหลายสกุลและหลากหลายชนิด รวมถึงมีชื่อทางวิทยาศาสตร์เฉพาะซึ่งมีความยาวของตัวอักษรมาก ทำให้การค้นหาจูลินทรีย์ตามสกุลและชนิดต่างๆ ทำได้ช้าและเกิดความผิดพลาดได้ง่ายในการสะกดตัวอักษรและการแสดงผลในการค้นหา

สำหรับแนวคิดของระบบที่พัฒนาขึ้นนี้ จะสร้างต้นไม้ตัดสินใจในการค้นหาจูลินทรีย์ชนิดต่างๆ โดยแตกเป็น โหนดลูกเมื่อมีตัวอักษรในชื่อต่างกัน จนกระทั่งเป็น ชื่อจูลินทรีย์ที่ค้นหา โดยเราพบว่าความสูงของต้นไม้ที่สร้างขึ้นจากแนวคิดนี้ มีความสูงไม่เกิน 25 ระดับเท่านั้น

จากผลการทดลองการค้นหาจูลินทรีย์ โดยใช้ต้นไม้ตัดสินใจ ทำการค้นหาจูลินทรีย์จำนวน 30 ชนิด พบว่าการค้นหาพบโหนดที่ต้องการที่ความลึกของต้นไม้เพียง 3 – 8 ระดับ และไม่เกิน 25 ระดับ ดังนั้นการค้นหาจูลินทรีย์ด้วยต้นไม้ตัดสินใจที่สร้างขึ้นทำให้ค้นหาได้อย่างรวดเร็วถึงแม้ว่าจะมีจูลินทรีย์อยู่ในฐานข้อมูลหลายพันชนิดก็ตาม

6.2 ข้อเสนอแนะ

แนวทางการพัฒนาต่อของระบบนี้ สามารถพัฒนาได้ คือ สามารถพัฒนาระบบเพื่อสร้างต้นไม้ตัดสินใจอัตโนมัติ เพื่อสามารถนำไปสร้าง DOM Tree โดยใช้เทคโนโลยี XML และนำไปแสดงผลบนเว็บสั่งชื่อจูลินทรีย์แบบเวลาจริงได้ ด้วยภาษา Java Script



บรรณานุกรม

บรรณานุกรม

ภาษาไทย

วิทยานิพนธ์

- ชลธิศา พลทองมาก. (2553). การวิเคราะห์ความเสี่ยงการเป็นโรคไวรัสตับอักเสบซี โดย
 ต้นไม้การตัดสินใจ. วิทยานิพนธ์ปริญญาโท สาขาเทคโนโลยีสารสนเทศ.
 ขอนแก่น: มหาวิทยาลัยขอนแก่น.
- พิภพ เต็งศิริวัฒนานนท์. (2552). การพัฒนาระบบฐานข้อมูลเอกสารอิเล็กทรอนิกส์กลาง
 โดยใช้เทคนิคต้นไม้ตัดสินใจในการค้นหา กรณีศึกษา บริษัท ออมรอน อิเล็กทรอนิกส์
 คอมโพเนนท์ จำกัด. วิทยานิพนธ์ปริญญาโท สาขาเทคโนโลยีสารสนเทศเพื่อการ
 จัดการ. กรุงเทพมหานคร: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.
- วาสนา วงษ์กำภู. (2552). ระบบสืบค้นงานวิจัยทางด้านจิตเวชศาสตร์โดยใช้ต้นไม้ตัดสินใจ
 กรณีศึกษา สถาบันจิตเวชศาสตร์สมเด็จเจ้าพระยา. วิทยานิพนธ์ปริญญาโท สาขา
 เทคโนโลยีสารสนเทศเพื่อการจัดการ. กรุงเทพมหานคร: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้า
 พระนครเหนือ.

เอกสารอื่นๆ

- กวีวัฒน์ อ่องล่อ. (2554). การวิเคราะห์ระดับความน่าเชื่อถือดูหนังสือของกรมการ
 หลวงด้วยต้นไม้ตัดสินใจ. การค้นคว้าแบบอิสระปริญญาโท สาขาวิทยาการ
 คอมพิวเตอร์. เชียงใหม่: มหาวิทยาลัยเชียงใหม่.
- ณัฐมณต์ สิริวัฒนานันท์. (2551). การตรวจสอบความเหมาะสมในการขนส่งสินค้าโดยวิธีต้นไม้
 ตัดสินใจ. สารนิพนธ์ปริญญาโท สาขาวิทยาการคอมพิวเตอร์. กรุงเทพมหานคร:
 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.
- ไพฑูรย์ จันทร์เรือง. (2550). ระบบสนับสนุนการตัดสินใจเลือกสาขาการเรียนของนักศึกษาระดับ
 ปริญญาตรี. สารนิพนธ์ปริญญาโท สาขาวิทยาการคอมพิวเตอร์.
 กรุงเทพมหานคร: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

ศักดิ์ชาย ตั่งประเสริฐ. (2550). การพัฒนาระบบจำแนกแบบทดสอบสำหรับผู้ทดสอบสุขภาพจิต ด้วยเทคนิค Decision Tree ผ่าน Web Application แบบ AJAX. การศึกษาอิสระปริญญา มหาวิทยาลัยเทคโนโลยีสารสนเทศ. ขอนแก่น: มหาวิทยาลัยขอนแก่น.

สารสนเทศจากสื่ออิเล็กทรอนิกส์

Liam R. E. Quin. (2010). XML Technology (XML Essentials). สืบค้นเมื่อ 23 กุมภาพันธ์ 2555, จาก <http://www.w3.org/standards/xml/core>

ภาษาต่างประเทศ

BOOKS

Ashok N. Kamthane. (2007). **Introduction to Data Structure in C**. India: Dorling Kindersley (India) Pvt. Ltd.

BEN COPPIN. (2004). **Artificial Intelligence Illuminated**. London: John and Bartlett Publishers International.

BENJAMIN S. WEEKS and I. EDWARD ALCAMO. (2008). **MICROBES AND SOCIETY**. (2nd ed.). CANADA: Jonhs and Bartlett Publishers, Inc.

Bing Liu. (2007). **Web Data Mining Explorin Hyperlinks, Contents, and Usage Data**. (2nd ed.). New York: Springer-Verlag Berline London.

Erik T. Ray. (2003). **Learning XML**. (2nd ed.). USA: O' Reilly Media, Inc.

G A V VAI. (2008). **Data Structures and algorithms Concepts, Techniques and Applications**. (2nd ed.). New Delhi: Tata McGraw-Hill.

Larry Ullman. (2009). **PHP for the Web**. (3rd ed.). USA: Larry Ullman.

Paul J. Deitel and Harvey M. Deitel. (2008). **AJAX, Rich Internet Applications and Web Development for programmers**. Boston : Pearson Education, Inc.,

Yong Yin, Ikuu Kaku, Jiafu Tang and JianMing Zhu. (2011). **Data Mining Concepts.**

Methods and Applications in Management and Engineering Design. New York:

Springer-Lerlag London.



ประวัติผู้เขียน**ชื่อ - นามสกุล**

กิริติ เต็มชำนาญ

ประวัติการศึกษา

ปริญญาวิทยาศาสตรบัณฑิต
สาขาเทคโนโลยีสารสนเทศ
คณะวิทยลัยนานาชาติ
มหาวิทยาลัยธุรกิจบัณฑิต
ปีการศึกษา 2552

ตำแหน่งและสถานที่ทำงานปัจจุบัน

พนักงานธุรการ
ศูนย์พันธุวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ
113 อุทยานวิทยาศาสตร์ประเทศไทย ถ.พหลโยธิน
ต.คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี 12120

ทุนการศึกษา

ทุนปริญญามหาบัณฑิต ตัวแทนจังหวัด
ปีการศึกษา 2553