บทที่ 4

อัลกอริธึมที่ใช้ในการสร้างข้อมูล

อัลกอริธึมที่ใช้สร้างข้อมูลแบบรายการจากรูป

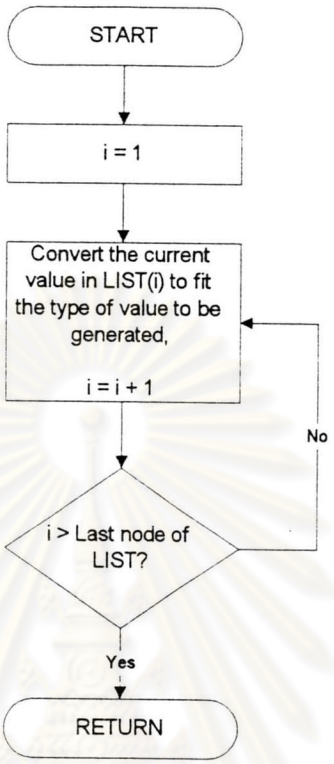
จากการที่ข้อมูลรายการมีการกำหนดขอบเขตโดยการใช้แกนนอนแทนตำแหน่งของข้อมูล โดยทางซ้ายจะเป็นตำแหน่งแรกของสมาชิกในรายการ ส่วนทางขวาสุดจะเป็นตำแหน่งของสมาชิกตัวสุดท้ายในรายการ ซึ่งแกนตั้งแทนค่าของข้อมูล โดยด้านล่างจะเป็นค่าต่ำสุดและด้านบนจะเป็นค่าสูงสุดในรายการ

การสร้างข้อมูลจะใช้วิธีการสร้างโดย

1. กำหนดตัวชี้ ให้ชี้ที่สมาชิกตัวแรกในรายการ
2. เขียนค่าของสมาชิกแต่ละตัวในรายการที่ตัวชี้ชี้ตำแหน่งของข้อมูลปัจจุบันอยู่
3. สร้างค่าของข้อมูลของสมาชิกแต่ละตัว และทำการแปลงค่าข้อมูลเพื่อให้ตรงกับชนิดของข้อมูลที่ระบุให้สร้าง ซึ่งการสร้างค่าของข้อมูลนี้จะทำโดยการเทียบตำแหน่งในแกนตั้งของจุดบนจอแสดงผล โดยตำแหน่งในแกนนอนนั้นจะสามารถคำนวณได้จากค่าของตัวชี้
4. ถ้าตัวชี้ยังไม่ชี้ที่สมาชิกตัวสุดท้ายของรายการ ให้เพิ่มค่าตัวชี้เพื่อทำการชี้ไปยังสมาชิกตัวต่อไปในรายการ แล้วกลับไปทำข้อ 2
5. จบการทำงาน

ซึ่งอัลกอริธึมที่ใช้สร้างข้อมูลแบบรายการจากรูปนี้สามารถแสดงเป็นแผนภูมิได้ดังรูปที่ 4.1

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.1 แผนภูมิของอัลกอริธึมที่ใช้สร้างข้อมูลแบบรายการจากรูป

อัลกอริธึมที่ใช้สร้างข้อมูลแบบต้นไม้จากรูป

หลักในการสร้างข้อมูลแบบต้นไม้โดยทั่วไปจากรูปหลายเหลี่ยมคือการเพิ่มโหนดทีละโหนดตั้งแต่โหนดที่เป็นรากจนกระทั่งถึงโหนดที่เป็นใบ โดยพยายามสร้างให้สอดคล้องกับรูปหลายเหลี่ยมที่สร้างไว้ ทั้งยังต้องสอดคล้องกับคุณสมบัติและชนิดต้นไม้นั้นๆ ด้วย และหากจะแจกแจงขั้นตอนออกไปก็สามารถแบ่งได้เป็น 5 ขั้นตอนดังต่อไปนี้

1) หาจุดยอดที่อยู่บนสุดของรูปหลายเหลี่ยมนั้นและถือว่าจุดนั้นเป็นจุดกำเนิด หรือจุดรากของต้นไม้ ซึ่งการหาจุดยอดของรูปหลายเหลี่ยมมีขั้นตอนดังต่อไปนี้

1. เริ่มจากจุดแรกในเซตของจุดยอดของรูปหลายเหลี่ยม ให้เก็บตำแหน่งในแกนตั้งของจุดแรกไว้

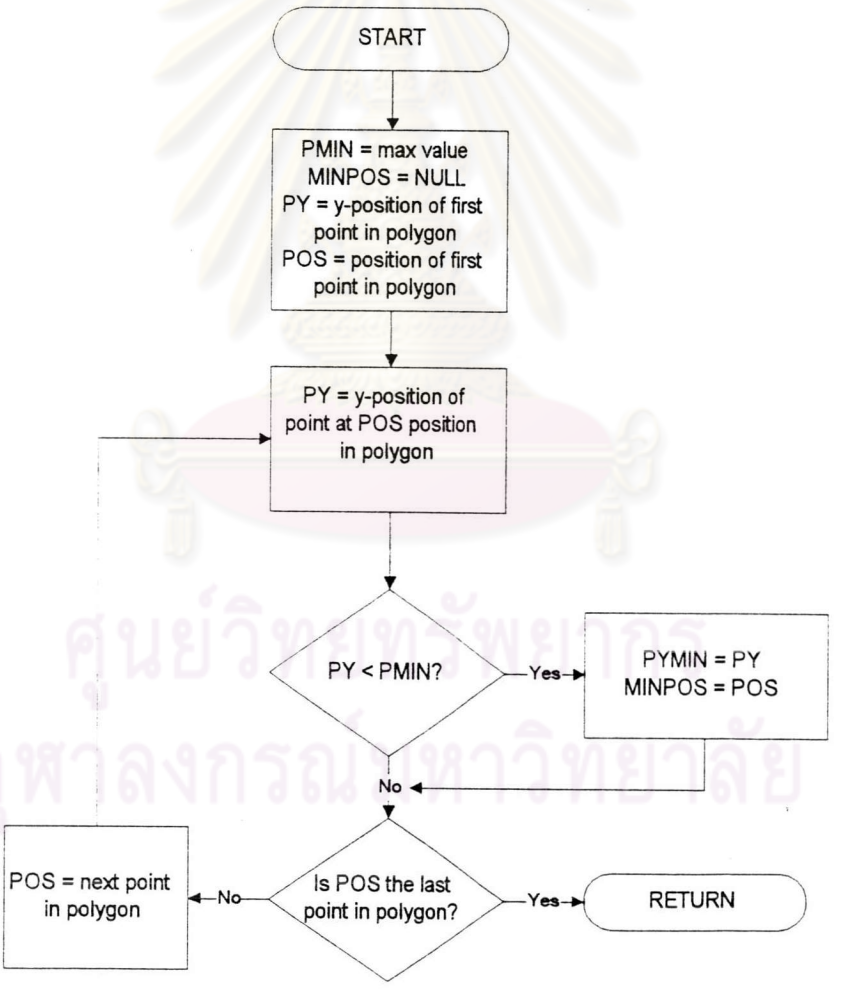
2. ให้พิจารณาไปจุดต่อไปในเซตของจุดยอดของรูปหลายเหลี่ยม ถ้าตำแหน่งในแกนตั้งของจุดปัจจุบันน้อยกว่าค่าที่เก็บไว้ ให้จำค่าของจุดปัจจุบัน และตำแหน่งของจุดปัจจุบันไว้แทน

(แกนตั้งในจอภาพของคอมพิวเตอร์จะกลับกันกับกราฟที่อยู่ในควอดแรนต์ (Quadrant) ที่ 1 แต่แกนนอนจะเหมือนกัน ดังนั้นจุดที่มีค่าในแกนตั้งที่น้อยหมายถึงว่าจุดนั้นอยู่สูงกว่าจุดที่มีค่าในแกนตั้งมาก)

3. ถ้ายังไม่ครบทุกจุดในเซตให้กลับไปข้อ 2

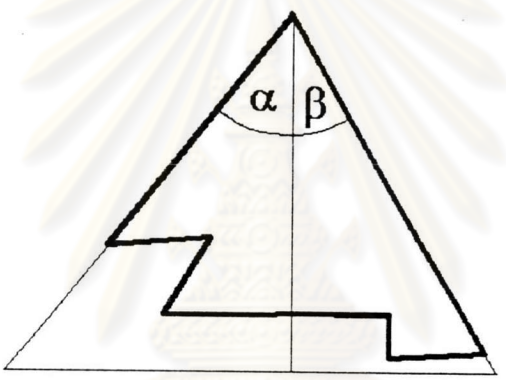
4. ครบทุกจุดแล้ว จะได้ตำแหน่งของจุดในเซตของจุดยอดของรูปหลายเหลี่ยมที่จะเป็นจุดรากของต้นไม้

ซึ่งขั้นตอนในการหาจุดยอดของรูปหลายเหลี่ยมสามารถแสดงเป็นแผนภูมิได้ดังรูปที่ 4.2



รูปที่ 4.2 แผนภูมิการหาจุดที่เป็นรากของต้นไม้ในรูปหลายเหลี่ยม

2) หาองศาของการสร้างข้อมูลต้นไม้ ซึ่งองศานี้จะเป็นตัวกำหนดขอบเขตของการบานออกของต้นไม้ทางด้านข้างซึ่งใช้ประกอบในการสร้างข้อมูลแบบต้นไม้ โดยหาจากมุมระหว่างเส้นตรงที่ลากผ่านจุดรากของต้นไม้และจุดที่อยู่ถัดไปตามเข็มนาฬิกากับแนวดิ่ง และมุมระหว่างเส้นตรงที่ลากผ่านจุดรากและจุดสุดท้ายที่อยู่ก่อนกลับมาบรรจบที่จุดยอดอีกครั้งกับแนวดิ่ง มุมที่บอกถึงการบานออกของต้นไม้ทางด้านข้างจะแบ่งออกเป็นสองส่วนคือมุมทางด้านซีกซ้ายของต้นไม้และมุมทางด้านซีกขวาของต้นไม้ จากตัวอย่างในรูปที่ 4.3 จะได้ว่ามุม α และ β จะถูกใช้เป็นมุมหลักเพื่อควบคุมการบานออกของต้นไม้ทางด้านซ้ายและขวาตามลำดับโดยไม่ให้การสร้างต้นไม้ล้นออกนอกกรอบของรูปหลายเหลี่ยมที่วาดไว้

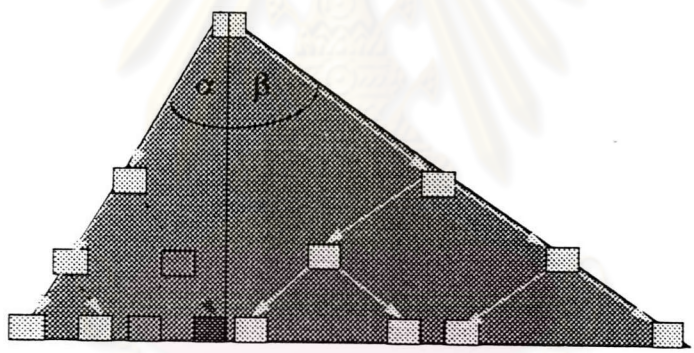


รูปที่ 4.3 มุมที่กำหนดขอบเขตของการบานออกของต้นไม้ทางด้านข้าง

3) การสร้างต้นไม้นี้ได้เลือกวิธีที่กำหนดให้ค่าองศาของมุมที่กำหนดขอบเขตของการบานออกของต้นไม้คงที่ จึงทำให้ต้องมีการเผื่อเนื้อที่ไว้ให้พอเพียงในการสร้างโหนดของต้นไม้ในแต่ละระดับ ดังนั้นเมื่อมีการวาดในระดับถัดมาจากบนลงล่างระยะห่างของโหนดในแต่ละระดับจะลดลงเป็นแบบลอกการิธึม (Logarithm) คือถ้าสมมุติให้ระยะห่างระหว่างระดับของโหนดรากและโหนดในระดับที่ 1 เป็น h จะได้ว่าให้ระยะห่างระหว่างระดับของโหนดในระดับที่ 2 กับโหนดในระดับที่ 1 เป็น $h/2$ และระยะห่างระหว่างระดับของโหนดในระดับที่ n กับโหนดในระดับที่ n-1 เป็น $h/(n\text{-}1)*2$ เมื่อ $n > 1$

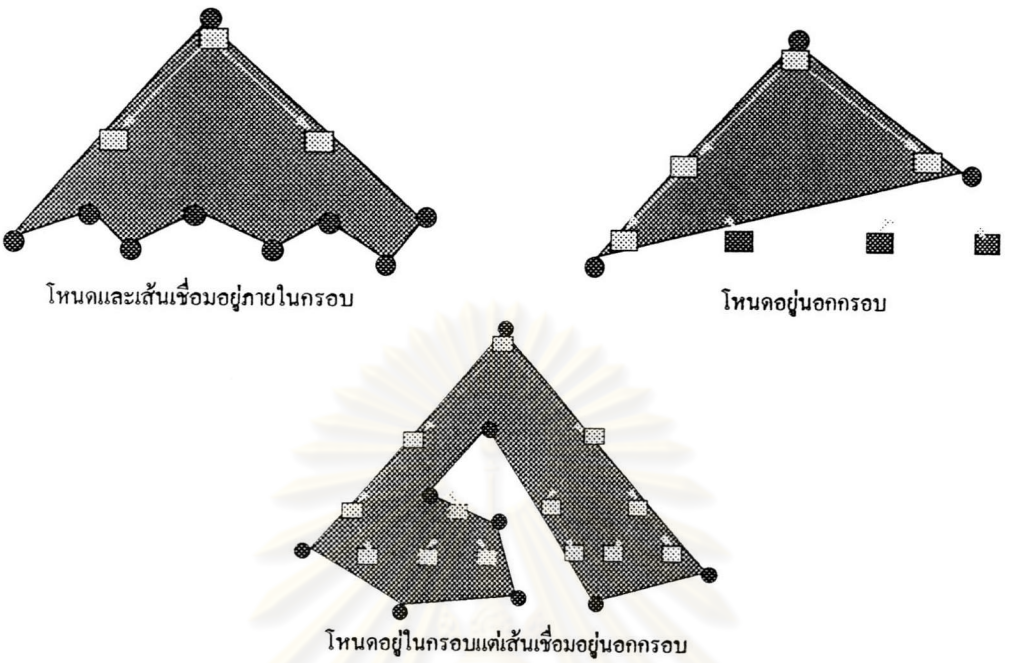
ส่วนการวาดในแนวนอนนั้นระยะห่างของแต่ละโหนดในระดับเดียวกันจะคิดจากองศาของการบานออกโดยการแบ่งการคำนวณออกเป็นสองส่วนคือทางด้านซีกซ้ายและซีกขวาของต้นไม้ (คือมุม α และ β ที่ได้จากข้อ 2 นั้นเอง) การตรวจสอบหาตำแหน่งของโหนดในแนวนอนจะ

เริ่มจากมุม -α ถึงมุม +β จากแกนแนวดิ่งของจุดยอดของรูปหลายเหลี่ยม (ซึ่งก็คือจุดที่เป็นโหนดรากของต้นไม้นั่นเอง) ซึ่งจะได้ระยะห่างของแต่ละโหนดในระดับเดียวกันและในซีกเดียวกันเท่ากันในกรณีที่ทุกโหนดถูกสร้างขึ้นในระดับนั้น ถ้ามุม α และ β มีขนาดใกล้เคียงกัน ระบบจะสร้างต้นไม้ทั้งฝั่งขวาและซ้ายโดยมีจำนวนโหนดเท่าๆ กัน (ถ้าไม่มีการกำหนดตัวเลือกในการสร้างต้นไม้เป็นอย่างอื่น) แต่ถ้ามุมทั้งสองมีความแตกต่างกันมาก (ในระบบกำหนดไว้ที่ 13% ซึ่งสายตาจะสามารถแยกออกได้ค่อนข้างชัดเจนว่ามุมทั้งสองมีขนาดแตกต่างกัน โดยระบบจะถือว่าผู้ใช้จงใจวาดรูปต้นไม้ให้มีความแตกต่างกันระหว่างข้างซ้ายและข้างขวา) ระบบจะลดจำนวนโหนดที่สร้างในฝั่งที่มีขนาดมุมน้อยกว่าลง เพื่อให้สอดคล้องกับรูปต้นไม้ที่ผู้ใช้วาด เช่นตัวอย่างในรูปที่ 4.4 ซึ่งต้นไม้ย่อยทางซ้ายควรมีจำนวนโหนดน้อยกว่าต้นไม้ย่อยทางขวา เนื่องจากมุม α มีขนาดเล็กกว่า β มาก ระบบจะทำให้ความน่าจะเป็นที่จะเกิดโหนดทางขวาลดลงมากกว่าโหนดทางซ้ายในต้นไม้ย่อยทางซ้าย

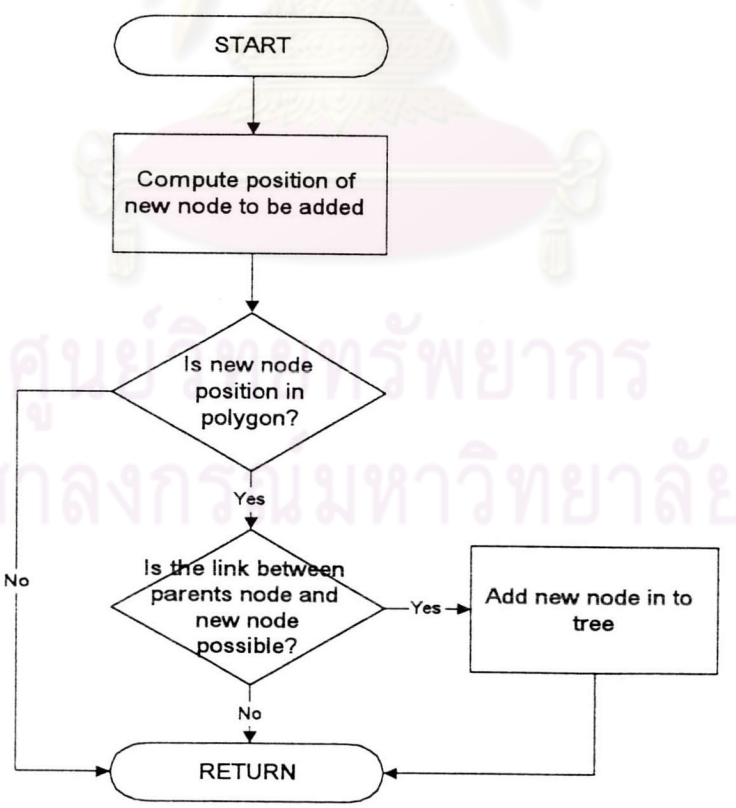


รูปที่ 4.4 เงื่อนไขในการสร้างโหนดในต้นไม้ เมื่อมุม α และ β ต่างกันมาก

4) เมื่อจะมีการเพิ่มโหนดใหม่เข้าไปในต้นไม้ ก่อนอื่นจะต้องมีการตรวจดูตำแหน่งก่อนว่าอยู่ในกรอบของรูปหลายเหลี่ยมนั้นหรือไม่ ถ้าไม่อยู่จะถือว่าโหนดนั้นไม่คงอยู่ หรือไม่เป็นสมาชิกของต้นไม้นั่นเอง แต่แม้ว่าโหนดนั้นอยู่ในรูปหลายเหลี่ยม แต่โหนดแม่ของโหนดใหม่นี้ไม่ได้ถูกสร้างขึ้นในต้นไม้ หรือเส้นที่เชื่อมระหว่างโหนดใหม่กับโหนดแม่มีการตัดกับเส้นขอบของรูปหลายเหลี่ยมอันจะทำให้การเชื่อมต่อระหว่างโหนดแม่กับโหนดใหม่นั้นเป็นไปไม่ได้ โหนดนั้นก็จะไม่ถูกเพิ่มเข้าไปในต้นไม้ ซึ่งเงื่อนไขเหล่านี้แสดงอยู่ในรูปที่ 4.5 โดยอัลกอริธึมที่ใช้ในการเพิ่มโหนดใหม่เข้าไปในต้นไม้สามารถแสดงเป็นแผนภูมิได้ดังรูปที่ 4.6



รูปที่ 4.5 เงื่อนไขในการเพิ่มโหนดใหม่เข้าไปในต้นไม้

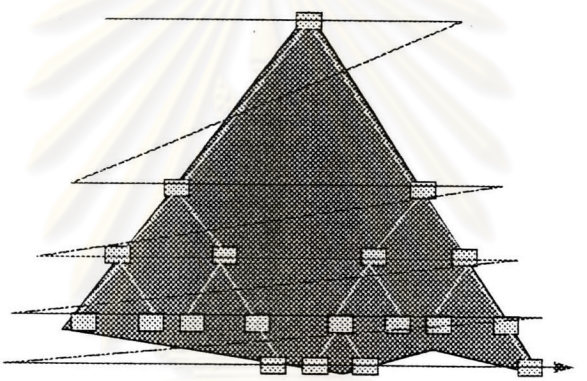


รูปที่ 4.6 แผนภูมิแสดงวิธีการเพิ่มโหนดใหม่เข้าไปในต้นไม้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5) เมื่อเพิ่มโหนดใหม่แต่ละโหนดเข้าไปในต้นไม้ ต้องมีการตรวจสอบคุณสมบัติและนิยามของต้นไม้นั้นๆ ว่าตรงหรือไม่ ถ้าการเพิ่มโหนดใหม่นั้นเข้าไปในต้นไม้ทำให้ต้นไม้ที่ได้ไม่เป็นที่สอดคล้องกับนิยามและคุณสมบัติของต้นไม้ที่กำหนดให้สร้าง โหนดที่ถูกเพิ่มเข้าไปใหม่นั้นก็จะถูกลบออกไป หรือเสมือนว่าไม่มีการเพิ่มโหนดนั้นเข้าไปนั่นเอง

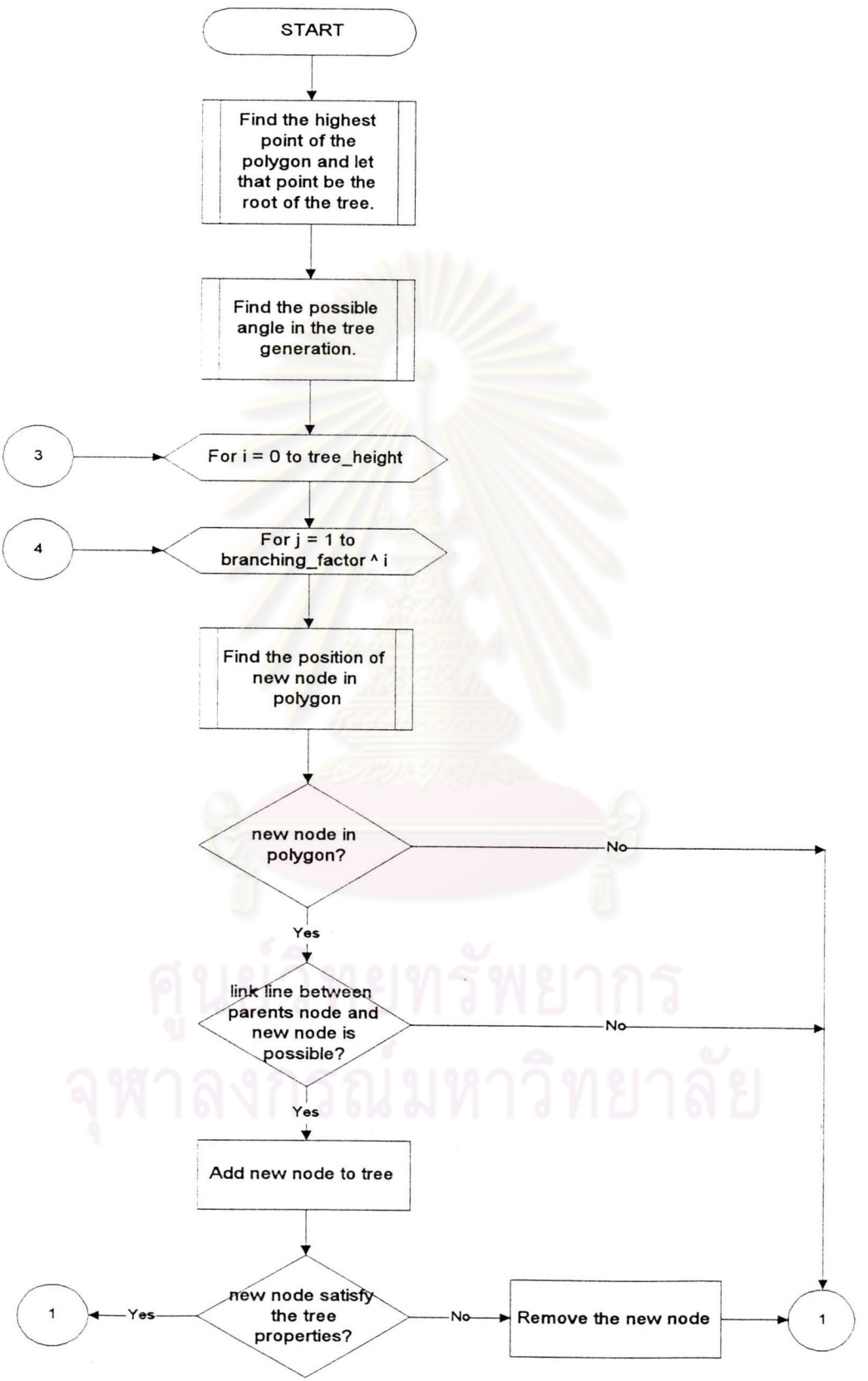
การหาตำแหน่งของการเพิ่มโหนดในต้นไม้นั้นจะกระทำจากบนลงล่าง (โดยเริ่มจากระดับรากลงมาถึงระดับที่เป็นใบล่างสุด) และจากโหนดซ้ายสุดมาขวาสุดในแต่ละระดับ ดังรูปที่ 4.7



รูปที่ 4.7 ลำดับการสร้างโหนดในต้นไม้

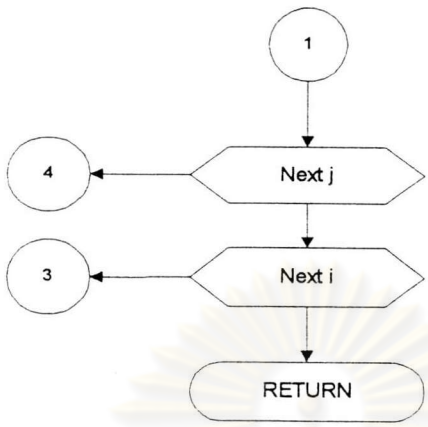
ส่วนการทำงานของอัลกอริธึมที่ใช้สร้างข้อมูลแบบต้นไม้จากรูปสามารถแสดงเป็นแผนภูมิได้ดังรูปที่ 4.8

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.8 แผนภูมิของอัลกอริธึมที่ใช้สร้างข้อมูลแบบต้นไม้จากรูป

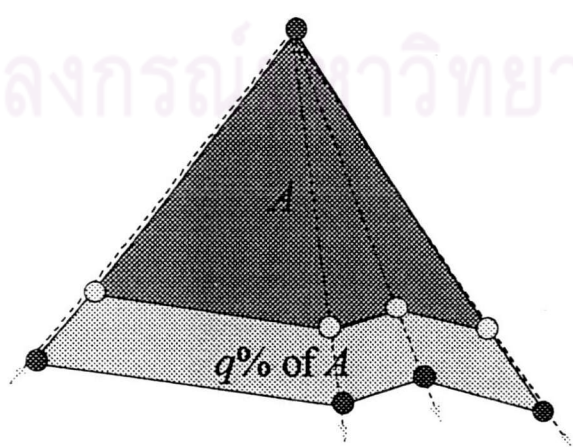
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.8 แผนภูมิของอัลกอริธึมที่ใช้สร้างข้อมูลแบบต้นไม้จากรูป

ตัวเลือกในการสร้างต้นไม้

1. ตัวเลือกเพื่อกำหนดขอบเขตใหม่ของรูปหลายเหลี่ยมโดยใช้การขยายพื้นที่ เนื่องมาจากรูปทรงของรูปหลายเหลี่ยมทำให้จำนวนโหนดที่สร้างขึ้นมาอาจไม่เพียงพอแก่ความต้องการ เนื่องจากหลักการสร้างโหนดในต้นไม้จะพยายามสร้างโหนดให้อยู่ภายในกรอบของรูปหลายเหลี่ยม ด้วยตัวเลือกนี้ ผู้ใช้สามารถกำหนดจำนวนโหนดที่ต้องการสร้าง (n) ซึ่งจะมีได้ไม่เกิน $\sum_{i=0}^{m} b^i$ เมื่อ b คือจำนวนโหนดลูกที่สามารถมีได้มากที่สุด m คือความสูงของต้นไม้ และกำหนดเปอร์เซนต์ของพื้นที่รูปหลายเหลี่ยนที่สามารถยืดออกได้ (q) เพื่อให้สามารถสร้างโหนดของต้นไม้ลงในรูปหลายเหลี่ยมให้ได้ครบตามจำนวนที่ต้องการ ดังรูปที่ 4.9



รูปที่ 4.9 การขยายพื้นที่เพื่อกำหนดของเขตใหม่ในการสร้างโหนด

การขยายพื้นที่มีหลักการดังต่อไปนี้คือ

ก. หาพื้นที่ของรูปหลายเหลี่ยมเดิม (A)

ข. หาพื้นที่ใหม่ที่เพิ่มขึ้น $q\%$ จากพื้นที่ A

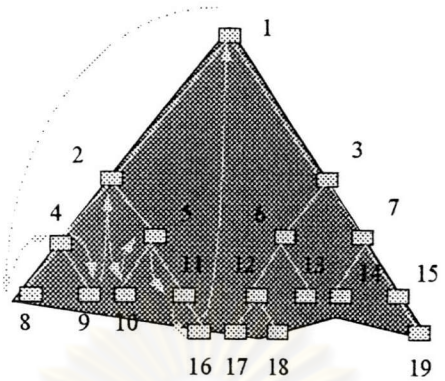
ค. หาอัตราส่วนของรูปหลายเหลี่ยมที่ต้องขยายออกตามแนวของโหนดใบ ซึ่งสาเหตุที่ต้องขยายออกข้างล่างตามแนวโหนดใบ ก็เพื่อการคงรูปทรงของต้นไม้ (รูปหลายเหลี่ยม) ไว้ เพื่อให้ต้นไม้ที่สร้างขึ้นมาใหม่หลังจากการขยายพื้นที่แล้วยังคงมีรูปร่างใกล้เคียงกับรูปทรงที่ยังไม่ได้ใช้การขยายพื้นที่เพื่อเพิ่มจำนวนโหนด ส่วนอัลกอริธึมที่ใช้สร้างโหนดยังคงใช้แบบเดิม ดังที่ได้กล่าวไว้ข้างต้น

2. ตัวเลือกการตัดโหนดที่เกินความต้องการออก ตัวเลือกนี้ใช้เพื่อตัดโหนดที่สร้างมาเกินความต้องการออก โดยเมื่อมีการเลือกตัวเลือกเพื่อขยายพื้นที่อาจทำให้จำนวนโหนดที่สร้างขึ้นตามพื้นที่ใหม่เพิ่มขึ้น

อัลกอริธึมที่ใช้หาอันดับของโหนดในต้นไม้เพื่อสร้างค่าข้อมูลให้

การนำค่าของข้อมูลเข้าไปใส่ในโหนดหลังจากที่ได้สร้างต้นไม้และเชื่อมต่อระหว่างโหนดแม่และลูกเรียบร้อยแล้ว ก็ต้องหาลำดับของโหนดที่ถูกต้องในการใส่ค่าของข้อมูล สำหรับข้อมูลของต้นไม้แบบทวิภาคค่าของข้อมูลของโหนดในต้นไม้แบบทวิภาคนี้ไม่มีข้อกำหนดใดๆ ดังนั้นค่าของข้อมูลจึงถูกสุ่มแล้วใส่ลงไปในแต่ละโหนด ส่วนข้อมูลของต้นไม้ค้นแบบทวิภาค ต้นไม้แบบเอวีแอล ต้นไม้ค้นแบบหลายทาง และต้นไม้แบบบี ได้ถูกกำหนดให้เรียงจากน้อยไปหามาก ดังนั้นเมื่อใช้การท่องไปในต้นไม้แบบอินออร์เดอร์ (Inorder Tree Traversal) เพื่อหาลำดับของโหนดของโหนดที่ถูกต้องได้แล้วจึงใส่ค่าข้อมูลลงไปในแต่ละโหนด ดังตัวอย่างในรูปที่ 4.10 จะได้ลำดับของโหนดเป็น 8, 4, 9, 2, 10, 5, 11, 16, 1, 17, ...

สำหรับค่าของข้อมูลในโหนดของต้นไม้ที่จำเป็นต้องอยู่ในรูปแบบที่ได้จัดเรียงลำดับ (Sort) ไว้แล้ว ระหว่างการทำงานของการท่องไปในต้นไม้แบบอินออร์เดอร์ ซึ่งจะมีการเยี่ยมโหนดในต้นไม้ในลักษณะที่เรียงลำดับไว้แล้วนั้น จะทำให้สามารถใส่ค่าข้อมูลลงไปในแต่ละโหนดที่เยี่ยมได้ทันที โดยใช้ตัวนับ (Counter) เพื่อนับค่าเพิ่มขึ้นตามแต่ที่ผู้ใช้กำหนดแต่ละครั้งไปเรื่อยๆ เมื่อมีการเยี่ยมโหนดแต่ละครั้ง เช่นตัวอย่างในรูปที่ 4.10 ถ้ากำหนดให้ค่าเริ่มต้นเป็น 0 และให้เพิ่มค่าทีละ 3 จะได้ว่า โหนด 8 มีค่า 0, โหนด 4 มีค่า 3, โหนด 9 มีค่า 6, โหนด 2 มีค่า 9, ... เป็นต้น



รูปที่ 4.10 การท่องไปในต้นไม้แบบอินออร์เดอร์

การท่องไปในต้นไม้แบบอินออร์เดอร์เพื่อทำการใส่ค่าข้อมูลลงในโหนดของต้นไม้ตามลำดับที่ถูกต้องมีหลักการดังนี้คือ

1. ท่องไปในต้นไม้ย่อยทางซ้ายให้ครบก่อน
2. กลับมาที่รากของต้นไม้ย่อยทางซ้ายนั้นเพื่อทำการเยี่ยมโหนด
3. ท่องไปในต้นไม้ย่อยทางขวา

(สุชาย ธนวเสถียร และ วิชัย จิวังกูร, 2531)

กำหนดให้

T เป็นตำแหน่งรากของต้นไม้ใดๆ

P เป็นตัวชี้ ซึ่งชี้ไปยังตำแหน่งปัจจุบันของโหนดที่กำลังพิจารณาอยู่

ฟังก์ชัน PUSH(*X*) สำหรับเก็บค่า *X* ลงสแตก (Stack) โดยสมมุติว่ามีสแตกให้ใช้ได้อยู่แล้ว

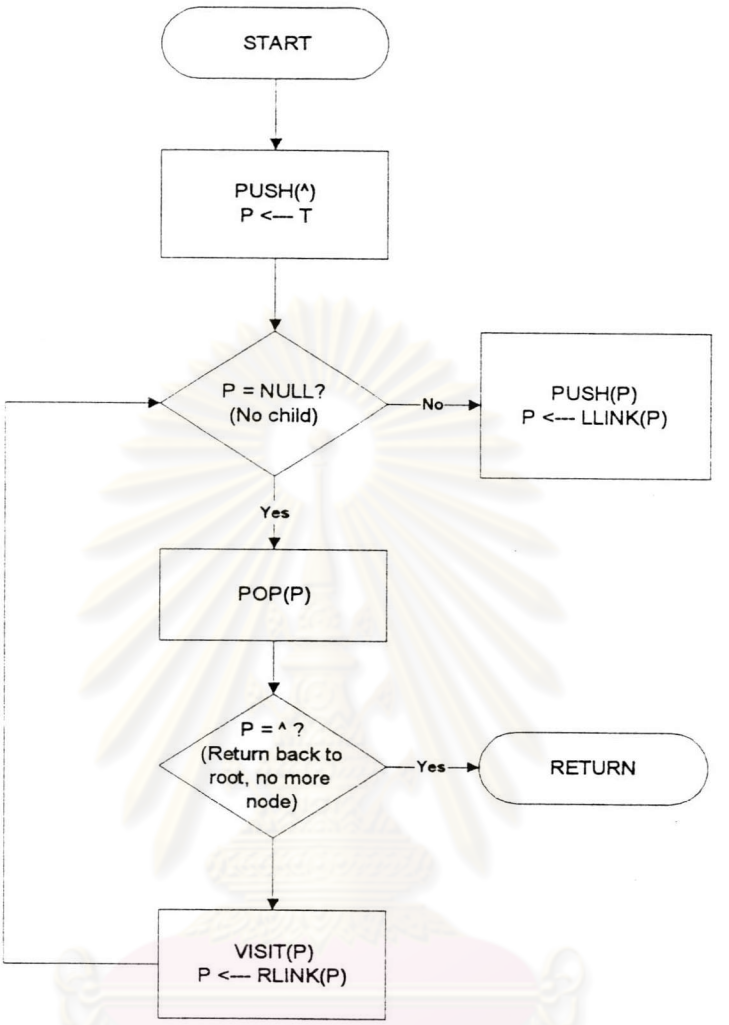
ฟังก์ชัน POP(*X*) สำหรับเอาค่าจากสแตกตัวบนสุดมาใส่ใน *X*

ฟังก์ชัน VISIT(*X*) สำหรับการเข้าเยี่ยมโหนด *X* รวมทั้งใส่ค่าของข้อมูลลงไปในโหนดด้วย

LLINK(*X*) ให้ตำแหน่งโหนดทางซ้ายของ *X*

RLINK(*X*) ให้ตำแหน่งโหนดทางขวาของ *X*

ดังนั้นจะได้แผนภูมิการทำงานของการท่องไปในต้นไม้แบบอินออร์เดอร์เป็นดังรูปที่ 4.11



รูปที่ 4.11 แผนภูมิของการท่องไปในต้นไม้แบบอินออร์เดอร์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย