บทที่ 3

การออกแบบระบบการตรวจรู้ลายพิมพ์ดีเอ็นเอ

ในบทนี้ จะได้บรรยายถึงวิธีการ แนวทางการแก้ปัญหา (approach) ตลอดจนการออกแบบระบบการตรวจรู้ลายพิมพ์ดีเอ็นเอ โดยจะกล่าวถึงรายละเอียดในวิธีการของแต่ละขั้นตอน

3.1 ลักษณะที่สำคัญของลายพิมพ์ดีเอ็นเอ

จากบทที่ 2 ได้ทราบถึงรายละเอียดของลายพิมพ์ดีเอ็นเอ ซึ่งมีลักษณะสำคัญที่ต้องคำนึงถึง สำหรับการตรวจรู้เรียงตามลำดับคือ

1. ตำแหน่งของลายพิมพ์
2. ความกว้างของลายพิมพ์
3. ความเข้มของลายพิมพ์ ซึ่งจะแบ่งออกเป็น 3 ระดับคือ
   - ก. ลายพิมพ์จาง
   - ข. ลายพิมพ์เข้ม
   - ค. ลายพิมพ์เข้มมาก

ตำแหน่งของลายพิมพ์จะมีความสำคัญมากกว่าลักษณะอย่างอื่น ดังนั้นจะใช้ตำแหน่งของลายพิมพ์ดีเอ็นเอเป็นคีย์ในการตรวจรู้

3.2 ขั้นตอนของการตรวจรู้ลายพิมพ์ดีเอ็นเอ

จากการศึกษารายละเอียดของลายพิมพ์ดีเอ็นเอ ทำให้สามารถกำหนดขั้นตอนของการตรวจรู้ได้ดังนี้

1. การแปลงมาตราส่วนตำแหน่งของลายพิมพ์ให้เป็นมาตรฐานเดียวกัน เพื่อที่จะสามารถนำลายพิมพ์ดีเอ็นเอที่ได้จากการทดลองมาเปรียบเทียบกันได้ ในที่นี้จะประยุกต์ใช้วิธีลากรานจ์ (Lagrange) (Atkinson และ Harley, 1983) ในการแปลงมาตราส่วนตำแหน่ง

2. การจัดแบ่งกลุ่มของลายพิมพ์ดีเอ็นเอ เพื่อประโยชน์ในการค้นหารูปแบบลายพิมพ์ที่ใกล้เคียง สำหรับในกรณีที่ไม่สามารถตรวจรู้ลายพิมพ์นั้นได้โดยตรง ในที่นี้จะประยุกต์ใช้วิธีหาระยะเลเวนช์ทีน (Levenshtein distance) (Dougherty และ Giardina, 1988) รวมกับจำนวนของลายพิมพ์ ในการจัดแบ่งกลุ่ม

3. การตรวจรู้ลายพิมพ์ดีเอ็นเอจากตำแหน่งของลายพิมพ์ที่ใช้เป็นคีย์ ในที่นี้จะประยุกต์ใช้ต้นไม้ค้นหาเชิงเลขฐานสอง (Binary Digital Search Tree) โดยใช้การแทนอย่างกระชับ (compact) (De Jonge, Tanenbaum และ Van de Riet, 1987)

4. การเปรียบเทียบหารูปแบบลายพิมพ์ที่ใกล้เคียง ในกรณีที่ไม่พบรูปแบบลายพิมพ์ที่ต้องการ เพื่อประโยชน์ในการตรวจสอบความเกี่ยวข้องทางสายเลือด หรือค้นหารูปแบบลายพิมพ์อื่นที่ใกล้เคียงเพราะอาจเป็นได้ว่ารูปแบบลายพิมพ์ที่ต้องการตรวจรู้นั้น อาจจะมีบางตำแหน่งของลายพิมพ์ที่หายไปหรือเกินมา ซึ่งในที่นี้จะประยุกต์ใช้วิธีแก้ไขชื่อที่สะกดผิดโดยอัตโนมัติ (Bickel, 1987)

รายละเอียดของแต่ละขั้นตอน มีดังนี้

3.2.1 <u>การแปลงมาตราส่วนตำแหน่งของลายพิมพ์ให้เป็นมาตรฐานเดียวกัน</u>

เนื่องจากลายพิมพ์ดีเอ็นเอเป็นผลที่ได้จากการทดลอง มาตราส่วนตำแหน่งของลายพิมพ์ที่ได้แต่ละครั้ง แม้ว่าจะเป็นของบุคคลเดียวกันก็อาจไม่เหมือนกันขึ้นอยู่กับระยะเวลาที่ใช้ แต่ทว่าในการทดลองจะมีการใช้ชิ้นของดีเอ็นเอที่รู้ขนาดที่แน่นอนเป็นตัวอ้างอิง ซึ่งใช้อ้างอิงตำแหน่งต่าง ๆ ได้เรียกว่า แถบอ้างอิง ดังนั้นก่อนที่จะสามารถเปรียบเทียบลายพิมพ์ดีเอ็นเอของบุคคลต่าง ๆ ได้ จึงต้องทำการแปลงให้เป็นมาตรฐานเดียวกันเสียก่อน โดยจะทำการกำหนดมาตราส่วนที่จะใช้เป็นมาตราส่วนมาตรฐาน มีขนาดดังรูปที่ 3.1 ลายพิมพ์ทุกขนาดจะถูกแปลงมาตราส่วนให้เป็นมาตราส่วนมาตรฐาน ซึ่งในที่นี้จะใช้วิธีลากรานจ์ อันเป็นวิธีการวิเคราะห์เชิงตัวเลข (Numerical Analysis) ในการแปลงมาตราส่วนตำแหน่งนี้

วิธีลากรานจ์เป็นเทคนิคหนึ่งของ การประมาณค่าด้วยโพลีโนเมียล (Polynomial interpolation) โดยที่รู้ค่าของ $f_0, f_1, \ldots, f_n$ ของฟังก์ชัน $f(x)$ ที่ $x_0, x_1, \ldots, x_n$ ตามลำดับ แต่ต้องการหาค่าของ $f(x)$ ที่ค่า x อื่น เช่น $x = x_a$ การประมาณค่าจะทำโดยหาสมการโพลีโนเมียล ที่มีกราฟผ่านทุกจุดที่รู้ค่าคือ $f_0, f_1, \ldots, f_n$ จากนั้นก็จะใช้สมการโพลีโนเมียลนั้นในการประมาณค่าของ $f(x)$ ที่ค่า x อื่น ที่ต้องการทราบค่า

สมการโพลีโนเมียลตามวิธีลากรานจ์ มีรูปแบบดังนี้

$$P_n(x) = f_0 l_0(x) + f_1 l_1(x) + \ldots + f_n l_n(x)$$

เมื่อ $$l_i(x) = \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)}$$

โดยฟังก์ชัน $l_i(x)$, เมื่อ $i = 0, 1, \ldots, n$ มีคุณสมบัติ

$l_i(x_j) = 1$, เมื่อ i เท่ากับ j

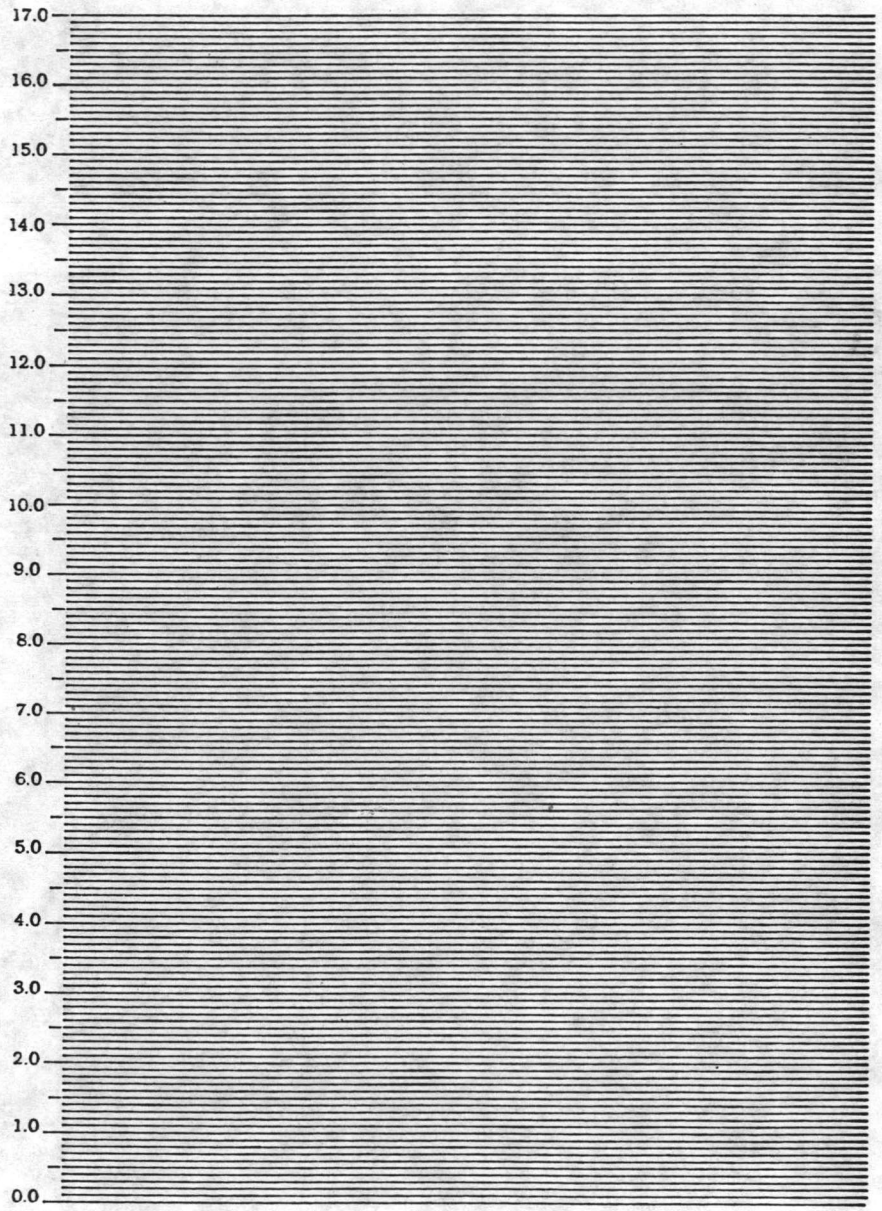
0, เมื่อ i ไม่เท่ากับ j ; $i, j = 0, 1, \ldots, n$

ซึ่งจะทำให้ได้ว่า

$$P_n(x_j) = f_j, \quad j = 0, 1, \ldots, n$$

อันหมายความว่า สมการโพลีโนเมียลที่สร้างขึ้นมานั้นให้ค่าเดียวกันกับ $f(x)$ ที่ทุกจุดที่ทราบค่า ดังนั้นสามารถที่จะใช้สมการโพลีโนเมียลนี้ในการประมาณค่าของ $f(x)$ ที่ค่า x อื่นได้

ลายพิมพ์ดีเอ็นเอบนแผ่นฟิล์ม จะถูกนำมาวางทาบลงบนมาตราส่วนมาตรฐานที่กำหนดขึ้นดังรูปที่ 3.1 ตำแหน่งของแถบอ้างอิงและตำแหน่งของลายพิมพ์บนแผ่นฟิล์มจะถูกอ่านเป็นตำแหน่งบนมาตราส่วนมาตรฐาน โดยจะใช้ตำแหน่งของแถบอ้างอิงเป็น $x_0, x_1, \ldots, x_n$ และตำแหน่งของลายพิมพ์เป็นค่า x อื่น ๆ ที่จะใช้สมการโพลีโนเมียลที่ได้จากวิธีลากรานจ์ในการประมาณค่า ซึ่งจากการทดลองนำลายพิมพ์ดีเอ็นเอขนาดหนึ่งมาทำการย่อและขยายแล้วใช้วิธีลากรานจ์แปลงค่ากลับพบว่าสามารถประมาณค่าได้แม่นยำ จะมีผิดพลาดบ้างเพียงทศนิยมตำแหน่งที่หนึ่งเท่านั้น ดังนั้นในขั้นตอนของการแบ่งกลุ่มลายพิมพ์ดีเอ็นเอและการแปลงลายพิมพ์ดีเอ็นเอเป็นคีย์นั้น จะไม่สนใจทศนิยมตำแหน่งที่หนึ่งนี้ (ตัดทศนิยมตำแหน่งที่หนึ่งทิ้ง)



รูปที่ 3.1 แสดงขนาดมาตราส่วนมาตรฐาน ที่กำหนดขึ้น

3.2.2 การจัดแบ่งกลุ่มของลายพิมพ์ดีเอ็นเอ

รูปแบบของลายพิมพ์ดีเอ็นเอ ซึ่งได้แปลงให้อยู่ในมาตราส่วนมาตรฐานเดียวกันแล้วจะถูกนำมาแบ่งกลุ่มตามความคล้ายของรูปแบบลายพิมพ์ ทั้งนี้ก็เพื่อประโยชน์ในการค้นหารูปแบบลายพิมพ์ที่ใกล้เคียง สำหรับในกรณีที่ไม่สามารถตรวจรู้ลายพิมพ์นั้นได้โดยตรง โดยจะนำตำแหน่งของลายพิมพ์เหล่านั้นมาคำนวณค่าความแตกต่าง (Dissimilarity) ซึ่งในที่นี้ได้ประยุกต์ใช้วิธีหาระยะเลเวนช์ทีน รวมกับจำนวนของลายพิมพ์ มาใช้ในการแบ่งกลุ่มตามค่าความแตกต่างนั้น โดยจะเรียกว่ากลุ่มความคล้าย ซึ่งค่าความแตกต่างนั้นจะคำนวณโดยเปรียบเทียบกับรูปแบบลายพิมพ์ที่กำหนดให้เป็นหลักรูปแบบหนึ่ง

ระยะเลเวนช์ทีน เป็นสิ่งที่สามารถใช้วัดความแตกต่างของสายวลี (string) ถ้าหากมีสายวลี X และ Y ระยะเลเวนช์ทีนจะถูกกำหนดเป็น จำนวนครั้งของการแก้ไขที่น้อยที่สุดในการทำให้สายวลี X เปลี่ยนเป็นสายวลี Y ในที่นี้แสดงด้วยสัญญลักษณ์ Ld(X,Y) ซึ่งการแก้ไขในที่นี้หมายถึง การเพิ่ม, การลบ และการแทนที่ของตัวอักขระ ถ้า Ld(X,Y) มีค่าน้อยหมายถึง สายวลี X และ Y มีความคล้ายกัน ในขณะค่าที่มากจะหมายถึง สายวลี X และ Y มีความแตกต่างกัน

ถ้าหากมีสายวลี X และ Y ดังนี้

$$X = x_1x_2x_3\ldots x_n$$

$$Y = y_1y_2y_3\ldots y_m$$

จะนิยามระยะระหว่างอักขระ x_i และ y_i ในที่นี้แสดงด้วยสัญญลักษณ์ $R(x_i,y_i)$ ดังนี้

$$R(x_i,y_i) = \begin{cases} 0, & \text{ถ้า } x_i \text{ เท่ากับ } y_i \\ 1, & \text{ถ้า } x_i \text{ ไม่เท่ากับ } y_i \end{cases}$$

และจะนิยาม ฟังก์ชันระยะระหว่างกลาง (intermediate distance function) ในที่นี้แสดงด้วยสัญญลักษณ์ D(i,j) เป็นค่าน้อยที่สุดระหว่าง $D(i-1,j)+1$, $D(i,j-1)+1$ และ $D(i-1,j-1)+R(x_i,y_i)$ ดังนี้

$$D(i,j) = \text{Min}\ [\ D(i-1,j)+1,\ D(i,j-1)+1,\ D(i-1,j-1)+R(x_i,y_i)\ ]$$

โดย $D(i,0) = i,\ i = 0,1,\ldots,n.$

และ $D(0,j) = j,\ j = 0,1,\ldots,m.$

ขั้นตอนวิธีการหาระยะเลเวนช์ทีนจะเป็นดังนี้

1. Let D(i,0) = i, i = 0,1,...,n.
2. Let D(0,j) = j, j = 0,1,...,m.
3. For i = 1,2,...,n and j = 1,2,...,m

   Let D(i,j) = Min [D(i-1,j)+1, D(i,j-1)+1,

   D(i-1,j-1)+R(x_i,y_i)]
4. Ld(X,Y) = D(n,m).

<u>ตัวอย่าง</u>

กำหนดให้ X = ababa

Y = aabbaa

สามารถหาค่าความแตกต่างของสายวลี โดยวิธีการหาระยะเลเวนช์ทีนได้ ดังนี้

ระยะระหว่างอักขระ x_i และ y_i, R(x_i,y_i) แสดงได้ดังตาราง

R	y_1=a	y_2=a	y_3=b	y_4=b	y_5=a	y_6=a
x_1=a	0	0	1	1	0	0
x_2=b	1	1	0	0	1	1
x_3=a	0	0	1	1	0	0
x_4=b	1	1	0	0	1	1
x_5=a	0	0	1	1	0	0

จาก $R(x_i,y_i)$ สามารถหาฟังก์ชันระยะระหว่างกลาง D(i,j) ตามขั้นตอนวิธีข้างต้น แสดงเป็นตารางได้ดังนี้

D	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	0	1	2	3	4	5
2	2	1	1	1	2	3	4
3	3	2	1	2	2	2	3
4	4	3	2	1	2	3	3
5	5	4	3	2	2	2	3

จะได้ระยะเลเวนช์ทีน $Ld(X,Y) = D(5,6) = 3$ ซึ่งหมายถึง สายวลี X และ Y มีความแตกต่างเท่ากับ 3 นั่นคือจะใช้การแก้ไข 3 ครั้งในการเปลี่ยน สายวลี X ไปเป็น สายวลี Y

ในงานวิจัยนี้ ได้กำหนดรูปแบบลายพิมพ์หลักให้เป็นลายพิมพ์ดีเอ็นเอที่มี จำนวนลายพิมพ์ 13 ลายพิมพ์ ซึ่งมีลายพิมพ์ที่ตำแหน่ง 28, 27, 26, 24, 22, 20, 18, 16, 14, 12, 10, 8 และ 6 โดยจะใช้ลายพิมพ์หลัก (X) นี้เป็นหลักในการเปรียบเทียบ ส่วนลายพิมพ์ดีเอ็นเอที่ต้องการตรวจรู้ (Y) ก็จะถูกนำมาเปรียบเพื่อหาค่าความแตกต่าง ค่าความแตกต่างที่คำนวณได้จะถูกนำมาจัดเป็นกลุ่มความคล้าย ๆ ละ 2 ค่า เช่น

ค่าความแตกต่าง 1-2 จัดเป็นกลุ่มความคล้ายระดับ 1

ค่าความแตกต่าง 3-4 จัดเป็นกลุ่มความคล้ายระดับ 2

ค่าความแตกต่าง 5-6 จัดเป็นกลุ่มความคล้ายระดับ 3

ค่าความแตกต่าง 7-8 จัดเป็นกลุ่มความคล้ายระดับ 4

ค่าความแตกต่างอื่น ๆ ก็จะถูกจัดกลุ่มในทำนองเดียวกัน

3.2.3 การตรวจรู้ลายพิมพ์ดีเอ็นเอจากตำแหน่งของลายพิมพ์ที่ใช้เป็นคีย์

จากการที่ลายพิมพ์ดีเอ็นเอมีความเป็นเอกลักษณ์ในแต่ละบุคคล จึงก่อให้เกิดรูปแบบลายพิมพ์ที่มีความแตกต่างกันอย่างมากมายมหาศาล โครงสร้างข้อมูลที่จะใช้เก็บลายพิมพ์ดีเอ็นเอที่ดีจึงควรจะใช้หน่วยความจำน้อย แต่สามารถเก็บรูปแบบของลายพิมพ์ได้มาก และยังต้องตรวจรู้ได้ถูกต้อง รวดเร็ว

De Jonge และคณะ (1987) ได้เสนอแนวทางของการแทนต้นไม้ค้นหาเชิงเลขฐานสอง (Binary Digital Search Tree) ซึ่งจะเรียกย่อว่า ต้นไม้แบบบีดีเอส (BDS-Tree) โดยใช้การแทนอย่างกระชับ (compact) อันจะทำให้การใช้หน่วยความจำเป็นไปอย่างมีประสิทธิภาพ ดังนี้

3.2.3.1 โครงสร้างต้นไม้ค้นหาเชิงเลขฐานสอง

ในเบื้องต้นจะขอนิยามความหมายของคำต่าง ๆ ดังนี้

คีย์ (key) : สายวลีบิต (bit string) ของ 0 และ 1 ซึ่งไม่จำเป็นต้องมีความยาวคงที่

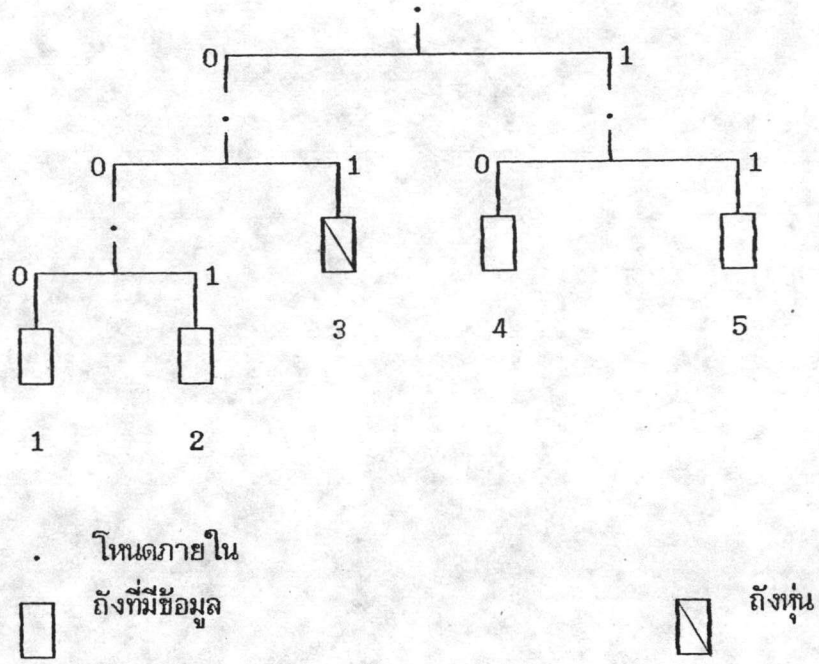
ถัง (bucket) : หน่วยของข้อมูลที่ถูกถ่ายโอน ระหว่างหน่วยความจำ (memory) และจานบันทึก (disk) ในการอ่านหรือเขียน 1 ครั้งโดยจะให้ b แทน ความจุ (capacity) ของถังซึ่งหมายถึงจำนวนระเบียนที่สามารถเก็บได้ใน 1 ถังนั่นเอง ซึ่งแต่ละถังจะมีเลขที่อยู่ทางกายภาพ (physical address) ที่ไม่ซ้ำกัน (unique)

ถังหุ่น (dummy bucket) : ถังที่ว่างไม่มีข้อมูล

โหนดภายใน (internal node) : โหนดที่มีโหนดลูก (descendant) อีก 2 โหนด

โหนดภายนอก (external node) หรือใบ (leaf) : โหนดที่ไม่มีโหนดลูก ซึ่งในที่นี้จะใช้ในความหมายว่าเป็นถัง

ตัวอย่างของต้นไม้แบบบีดีเอสแสดงดังรูปที่ 3.2 ซึ่งจะให้วิถี (path) ด้านซ้ายเป็น 0, วิถีด้านขวาเป็น 1 ดังนั้นจากรูป ถังหมายเลข 1, 2, 3, 4 และ 5 ก็จะมีวิถีที่นับจากราก (root) เป็น 000, 001, 01, 10 และ 11 ตามลำดับ



รูปที่ 3.2 แสดงวิถีของต้นไม้แบบบีดีเอส

3.2.3.1.1 <u>การค้นหา (Searching)</u>

ขั้นตอนการค้นหาข้อมูลในต้นไม้แบบบีดีเอสนั้นค่อนข้างง่าย โดยจะเริ่มจากรากของต้นไม้ ค่าของคีย์ที่ต้องการค้นหาจะถูกอ่านไปทีละบิต ๆ เริ่มจากปลายข้างซ้าย ถ้าเป็นบิต 0 ก็จะไปวิถีด้านซ้าย แต่ถ้าเป็นบิต 1 ก็จะไปวิถีด้านขวา เมื่อถึงใบก็หยุดซึ่งตรงกับถังหมายเลขเท่าไร ถังหมายเลขนั้นก็จะเป็นถังที่มีข้อมูลของคีย์ที่ต้องการอยู่ ยกตัวอย่างเช่น จากต้นไม้ดังรูปที่ 3.2 คีย์ 00101 ก็จะอยู่ในถังหมายเลข 2, คีย์ 110011 ก็จะอยู่ในถังหมายเลข 5 เป็นต้น ซึ่งอาจกล่าวได้ว่าคีย์จะอยู่ในถังที่มีวิถีตามรูปแบบของคีย์นั้น

ให้สังเกตว่าหมายเลขของถัง (bucket number) และเลขที่อยู่ของถัง (bucket address) นั้นไม่เหมือนกัน, หมายเลขของถังจะเริ่มจาก 1 ไปจนถึงค่ามากที่สุดค่าหนึ่ง ซึ่งก็คือจำนวนของถังที่พบทั้งหมดเมื่อท่องไปในต้นไม้จนทั่วแล้ว ซึ่งวิธีที่จะเปลี่ยนจากหมายเลขของถัง ไปเป็นเลขที่อยู่ของถังนั้น จะได้กล่าวถึงต่อไป

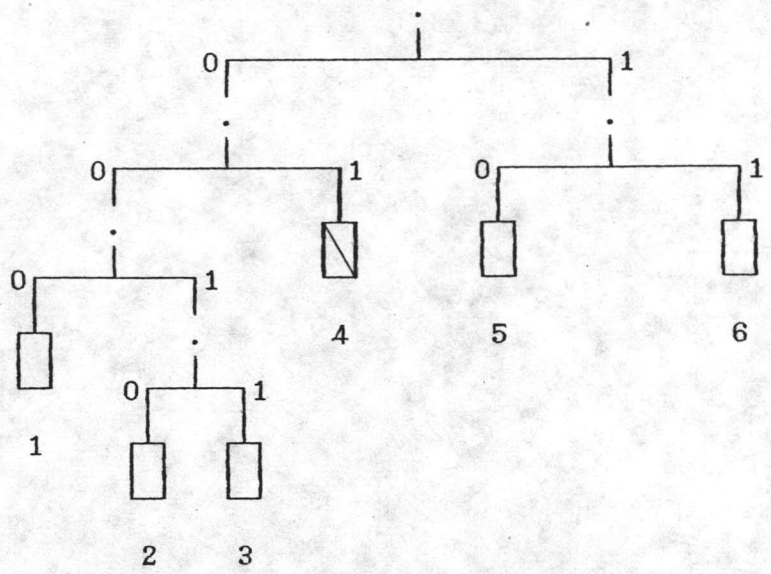
3.2.3.1.2 การแทรก (Insertion)

ในการแทรกข้อมูลใหม่นั้น สามารถแยกพิจารณาได้เป็น 3 กรณี คือ

1. ถังนั้นมีข้อมูลอยู่บ้างแล้วแต่ยังไม่เต็ม การแทรกข้อมูลในกรณีนี้เป็นเรื่องธรรมดาคือ อ่านข้อมูลในถังที่ต้องการเข้ามาแล้วแทรกข้อมูลระเบียนใหม่ลงไปจากนั้นก็บันทึกถังนั้นกลับลงไปในจานบันทึกใหม่

2. ถังนั้นเป็นถังหุ่นดังนั้นข้อมูลที่จะแทรกใหม่นี้ก็จะเป็นข้อมูลแรก เนื่องจากถังหุ่นนั้นยังไม่เคยได้รับการจัดสรรเนื้อที่ในจานบันทึก จึงต้องมีการจัดสรรเนื้อที่ใหม่ในจานบันทึกให้กับถังหุ่นนี้ หลังจากนั้นถังนั้นก็จะไม่เป็นถังหุ่นอีกต่อไป

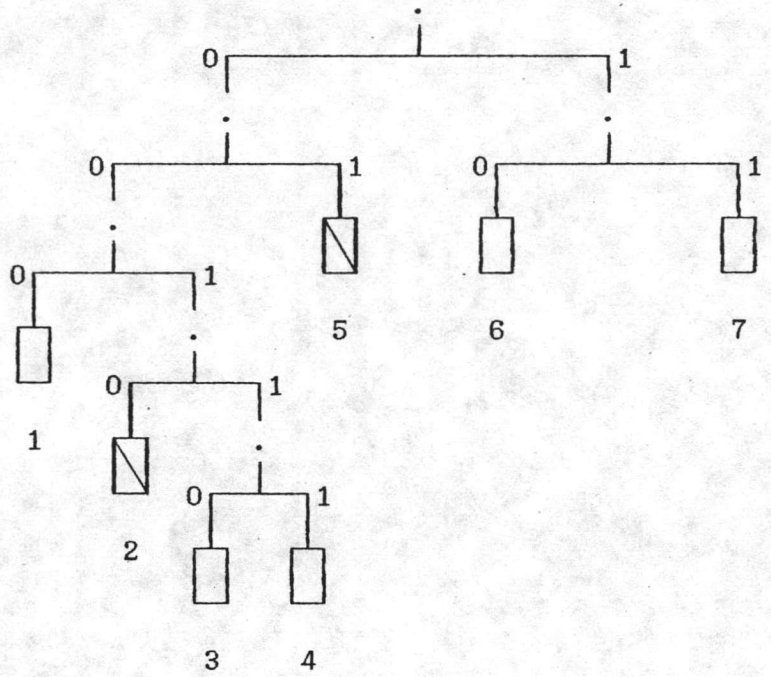
3. ถังนั้นเต็มแล้ว จึงต้องมีการแยก (split) ถังที่เต็มนั้นออกเป็นถังใหม่ 2 ถัง และจำนวนคีย์ทั้ง b+1 (ของเดิมในถังมีอยู่ b รวมกับของใหม่อีก 1) นั้นก็จะกระจายไปอยู่ในถังใหม่ 2 ถังนี้ พิจารณาต้นไม้ดังรูปที่ 3.2 สมมติว่าต้องการแทรกข้อมูลใหม่ลงในถังหมายเลข 2 ซึ่งเต็มแล้ว สิ่งที่เกิดขึ้นก็คือโหนดภายนอกก็จะถูกเปลี่ยนเป็นโหนดภายในที่มี 2 ถังใหม่อยู่ภายใต้โหนดนี้ดังรูปที่ 3.3 (สังเกตว่าเมื่อมีการแยกหมายเลขของแต่ละถังก็จะเปลี่ยนไป) ก่อนที่จะมีการแยก, คีย์ที่เริ่มต้นด้วย 001 จะถูกเก็บอยู่ที่ถังหมายเลข 2 แต่หลังจากการแยกจะเป็นคีย์ที่เริ่มต้นด้วย 0010 ที่จะถูกเก็บอยู่ที่ถังหมายเลข 2 และคีย์ที่เริ่มต้นด้วย 0011 จะถูกเก็บอยู่ที่ถังหมายเลข 3



รูปที่ 3.3 แสดงต้นไม้แบบบีดีเอส จากรูปที่ 3.2 เมื่อมีการแยก

ในกรณีปกติ จำนวนคีย์ประมาณครึ่งหนึ่งจะถูกเก็บอยู่ในถังหมายเลข 2 และที่เหลือทั้งหมดก็จะถูกเก็บอยู่ในถังหมายเลข 3 แต่อย่างไรก็ตาม กรณีที่เลวที่สุดคือคีย์ทั้งหมดอาจจะไปลงที่ถังเดียวกันอีกก็ได้ ไม่กระจายไปในอีกถังหนึ่ง ยกตัวอย่างเช่น ต้นไม้ดังรูปที่ 3.2 ถ้าให้ b (ความจุ) เป็น 4 และเดิมถังหมายเลข 2 เก็บคีย์ 0011000, 0011001, 0011011 และ 0011110 คีย์ที่ต้องการแทรกคือ 0011101 เมื่อถังหมายเลข 2 เกิดการแยกคีย์ทั้ง 5 นั้นก็จะไปลงที่ถังหมายเลข 3 (ถังใหม่ที่เกิดจากการแยกดังรูปที่ 3.3) หมด ซึ่งตัวมันเองก็จะต้องทำการแยกอีกครั้งหนึ่งจนกระทั่งต้นไม้เป็นดังรูปที่ 3.4 ซึ่งจะมีจำนวนคีย์ 3 คีย์อยู่ที่ถังหมายเลข 3, จำนวนคีย์ 2 คีย์อยู่ที่ถังหมายเลข 4 และถังหมายเลข 2 ก็จะกลายเป็นถังหุ่นไป

อาจกล่าวได้ว่า ถ้าคีย์ทั้งหมดในถังที่ต้องการแยกมีรูปแบบของคีย์ที่เหมือนกันยาวมากเท่าไร ลำดับชั้น (level) ใหม่ ๆ ก็จะถูกสร้างขึ้นเพื่อแยกความแตกต่างมากเท่านั้น (แต่ละลำดับชั้น หมายถึงแต่ละบิตของคีย์)

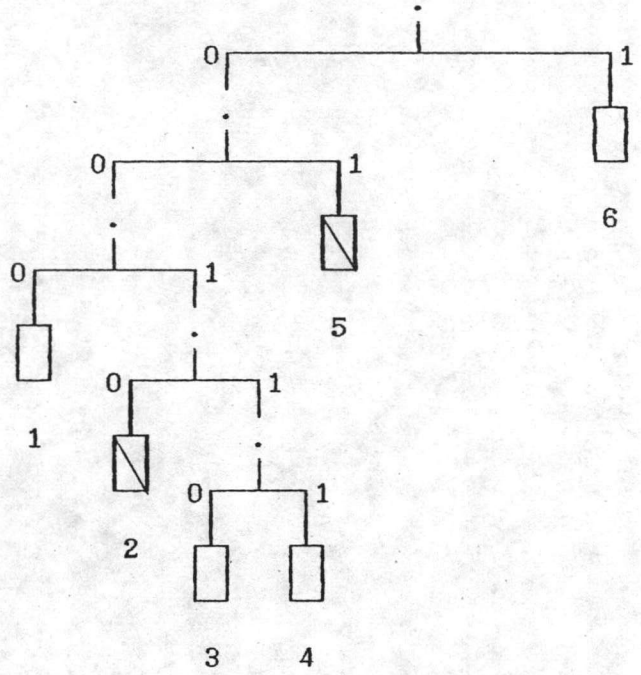


รูปที่ 3.4 แสดงต้นไม้แบบบีดีเอส จากรูปที่ 3.3 เมื่อมีการแยก

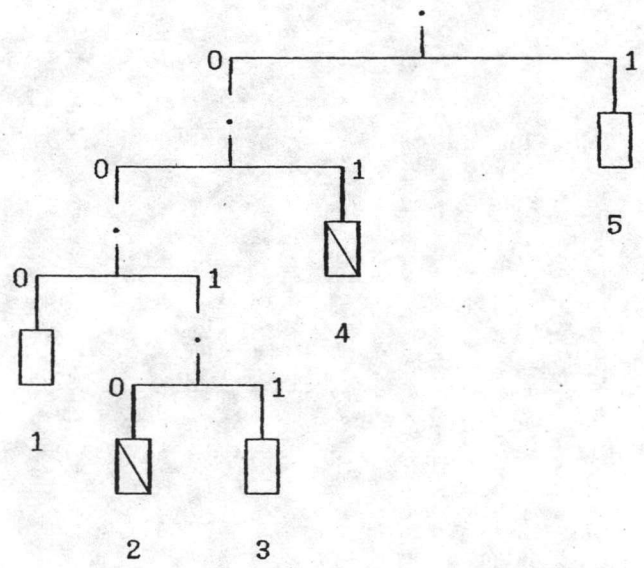
3.2.3.1.3 การลบ (Deletion)

การลบนอกจากจะนำระเบียน ที่มีคีย์ตามที่ระบุออกแล้ว ยังต้องตรวจสอบหลังจากการลบแต่ละครั้งว่าถังนั้นสามารถที่จะรวมกับถังข้างเคียงได้หรือไม่ เป็นต้นว่าคีย์ที่เหลืออยู่ (ในถังนั้นหลังจากลบแล้ว) มีไม่ถึงครึ่งหรือไม่เหลือเลย ยกตัวอย่างเช่นจากต้นไม้ดังรูปที่ 3.5 (ก) ถ้าลบคีย์บางส่วนออกจากถังหมายเลข 4 และถ้าสามารถรวมถังหมายเลข 3 และถังหมายเลข 4 เข้าด้วยกันก็จะได้ต้นไม้ดังรูปที่ 3.5 (ข) ซึ่งสามารถเปลี่ยนเป็นต้นไม้ดังรูปที่ 3.5 (ค) โดยการรวมถังหมายเลข 3 เข้ากับถังหมายเลข 2 ที่เป็นถังหุ่น

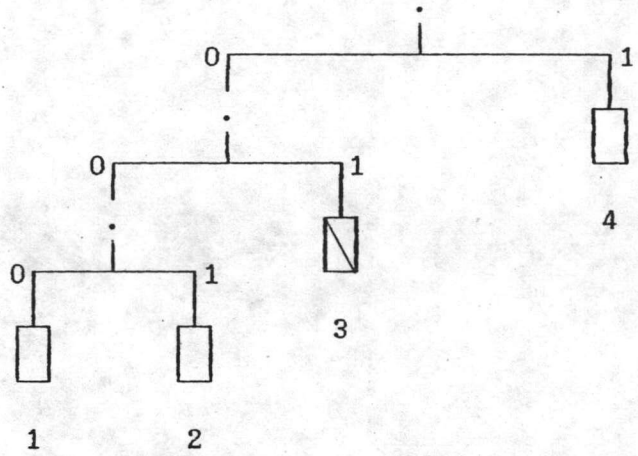
แต่สำหรับในกรณีของ ลายนิมน์ดีเอ็นเอนั้น ถังที่ควรจะถูกรวมเข้ากับถังข้างมีโอกาสที่จะถูกอ้างถึงได้อีกมาก ทั้งนี้เพราะว่าลายนิมน์ดีเอ็นเอมีรูปแบบที่หลากหลาย ดังนั้นการลบข้อมูลที่เพียงนำระเบียนที่ต้องการออกจากถังก็ถือว่าพอเพียงแล้ว



รูปที่ 3.5 (ก) แสดงต้นไม้แบบบีดีเอส ก่อนการลบ



รูปที่ 3.5 (ข) แสดงต้นไม้แบบบีดีเอส จากรูปที่ 3.5 (ก) หลังจากการลบ



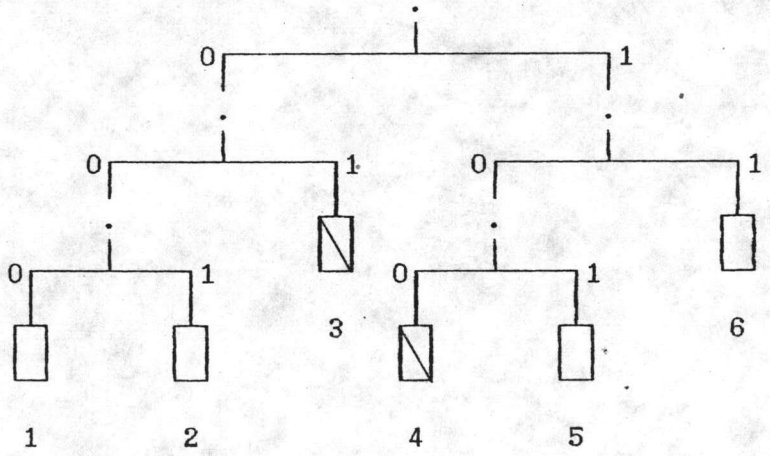
รูปที่ 3.5 (ค) แสดงต้นไม้แบบบีดีเอส เมื่อรวมกับถังหุ่นหมายเลข 2 จากรูปที่ 3.5 (ข)

3.2.3.2 <u>การแทนต้นไม้แบบบีดีเอสอย่างกระชับ</u>

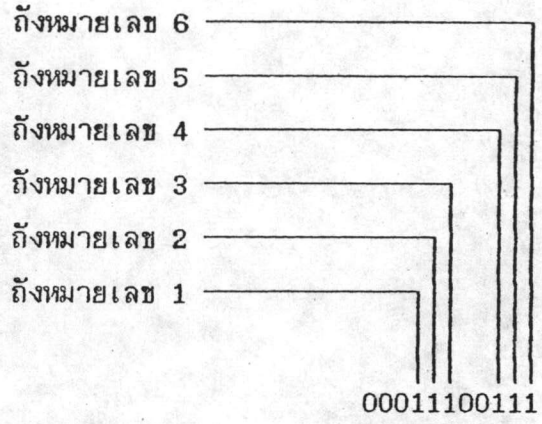
โดยปกติวิธีที่ง่าย ๆ ในการแทนต้นไม้แบบแตกสอง (binary tree) ทั่ว ๆ ไป ทำได้โดยให้โหนดภายในแต่ละโหนดมีเข็มชี้ (pointer) 2 ตัว และมีบิตเครื่องหมาย (sign bit) สำหรับเข็มชี้แต่ละตัว เพื่อใช้แยกความแตกต่างระหว่างเข็มชี้ไปยังโหนดภายในและเข็มชี้ไปยังเลขที่อยู่ของถัง ซึ่งการแทนต้นไม้ด้วยวิธีนี้เป็นวิธีที่ใช้กันทั่วไปแต่ผลที่ได้ยังไม่กระชับเพียงพอ วิธีที่จะทำให้กระชับได้มากกว่านี้ก็คือจะใช้การแทนต้นไม้ด้วยตัวแทนแบบเชิงเส้น (linear representation) ที่ประกอบด้วยสายวลีบิต 0 และ 1 ซึ่งได้จากการท่องไปในต้นไม้แบบพรีออร์เดอร์ (preorder) โดยให้ 0 แทนโหนดภายในที่พบและ 1 แทนถังที่พบไม่ว่าจะเป็นถังหุ่นหรือไม่ก็ตาม ยกตัวอย่างเช่น ต้นไม้ดังรูปที่ 3.2 สามารถแทนในรูปแบบตัวแทนแบบเชิงเส้นได้เป็น 000111011 ซึ่งได้มาจากการท่องไปในต้นไม้แบบพรีออร์เดอร์ โดยตอนแรกจะพบโหนดภายใน 3 โหนด (รวมรากของต้นไม้ด้วย, 000) จากนั้นก็เป็นถังหมายเลข 1, 2 และถังหุ่นหมายเลข 3 (111) จากนั้นก็เป็นโหนดภายในอีก 1 โหนด (0) และในที่สุดก็จะเป็นถังหมายเลข 4 และ 5 (11) ซึ่งในตัวแทนแบบเชิงเส้นนี้บิต 1 แรกจะหมายถึงถังหมายเลข 1, บิต 1 ที่สองก็จะหมายถึงถังหมายเลข 2 เช่นนี้เรื่อยไป

3.2.3.2.1 การเก็บเลขที่อยู่ของถัง

จะเก็บเลขที่อยู่ของถังไว้ในอีกตารางหนึ่ง แยกต่างหากเรียกว่าตารางเลขที่อยู่ (address table) ซึ่งจะใช้หมายเลขของถังเป็นดรรชนี (index) พิจารณารูปที่ 3.6 (ข) ซึ่งแสดงตัวแทนแบบเชิงเส้นของต้นไม้รูปที่ 3.6 (ก) หมายเลขของแต่ละถังก็คือลำดับที่ของบิต 1 ในตัวแทนแบบเชิงเส้นนั้น เมื่อมีการเปลี่ยนแปลงต้นไม้หมายเลขของถังอาจจะเปลี่ยน แต่เลขที่อยู่ของถังจะไม่เปลี่ยนเช่น ถ้าถังหมายเลข 2 เต็มและเกิดการแยก ถังหมายเลข 3, 4, 5 และ 6 ก็จะถูกจัดใหม่เป็น 4, 5, 6 และ 7 ตามลำดับ ตารางเลขที่อยู่ของต้นไม้รูปที่ 3.6 (ก) แสดงดังรูปที่ 3.6 (ค) โดยมีการใช้สมนัยบิต (bit map) ช่วยซึ่งสมนัยบิตจะมีจำนวนบิตเท่ากับจำนวนถังทั้งหมดของต้นไม้ สำหรับแต่ละถังถ้าค่าของบิตในสมนัยบิตที่สมนัยเป็น 0 แสดงว่าถังนั้นเป็นถังหุ่นแต่ถ้าเป็น 1 แสดงว่าถังนั้นมีข้อมูลเช่น สมนัยบิตของต้นไม้ดังรูปที่ 3.6 (ก) คือ 110011 หมายความว่าถังหมายเลข 3, 4 เป็นถังหุ่นนอกนั้นถังหมายเลข 1, 2, 5 และ 6 มีข้อมูล



รูปที่ 3.6 (ก) แสดงต้นไม้แบบบีดีเอส



รูปที่ 3.6 (ข) แสดงตัวแทนแบบเชิงเส้นของต้นไม้แบบบีดีเอส รูปที่ 3.6 (ก)

เลขที่อยู่ของถังหมายเลข 1
เลขที่อยู่ของถังหมายเลข 2
เลขที่อยู่ของถังหมายเลข 5
เลขที่อยู่ของถังหมายเลข 6

สมนัยบิต : 110011

รูปที่ 3.6 (ค) แสดงตารางเลขที่อยู่ และสมนัยบิต
ของต้นไม้แบบบีดีเอส รูปที่ 3.6 (ก)

3.2.3.2.2 ขั้นตอนวิธีในการค้นหา

เนื่องจากต้นไม้ในที่นี้แทนด้วย ตัวแทนแบบเชิงเส้น ดังนั้นการค้นหาข้อมูลจึงต้องมีขั้นตอนเป็นพิเศษ ในการค้นหาต้องมีตัวแทนแบบเชิงเส้นของต้นไม้ และตัวคีย์ที่ต้องการค้นหา ในที่นี้จะใช้ตัวชี้ (marker) 2 ตัว ตัวแรกให้ชี้อยู่ที่ตำแหน่งปัจจุบัน (current position) ของตัวแทนแบบเชิงเส้นเรียกว่าตัวชี้ต้นไม้ (tree marker) ตัวที่สองให้ชี้อยู่ที่ตำแหน่งปัจจุบันของคีย์เรียกว่าตัวชี้คีย์ (key marker) โดยในตอนเริ่มต้นตัวชี้ต้นไม้ จะชี้อยู่ที่ตำแหน่งรากของต้นไม้ ซึ่งก็คือบิตแรกของตัวแทนแบบเชิงเส้น และตัวชี้คีย์จะชี้อยู่ที่บิตแรกของคีย์

ในขณะใด ๆ ถ้าตัวชี้ต้นไม้ชี้ที่บิต 0 ซึ่งหมายถึงโหนดภายใน บิตของคีย์ที่ถูกชี้โดยตัวชี้คีย์ จะเป็นตัวกำหนดว่าจะให้เลื่อนตัวชี้ต้นไม้ไปยังต้นไม้ย่อยซ้าย (left subtree) หรือต้นไม้ย่อยขวา (right subtree) ถ้าบิตนี้ของคีย์เป็น 0 ก็จะเป็นการเลื่อนตัวชี้ต้นไม้ไปต้นไม้ย่อยซ้าย แต่ถ้าเป็น 1 ก็จะเป็นการเลื่อนตัวชี้ต้นไม้ไปต้นไม้ย่อยขวา

ในการเลื่อนตัวชี้ต้นไม้ไปยังต้นไม้ย่อยซ้ายนั้นทำได้ง่าย โดยเพียงขยับตัวชี้ต้นไม้ไปยังตำแหน่งถัดไปเท่านั้น ทั้งนี้เพราะในตัวแทนแบบเชิงเส้นของต้นไม้ ซึ่งได้มาจากการท่องไปในต้นไม้แบบพรีออร์เดอร์จะไปเยี่ยม (visit) ต้นไม้ย่อยซ้ายก่อนต้นไม้ย่อยขวา ดังนั้นต้นไม้ย่อยซ้ายก็จะอยู่ถัดจากโหนดภายในนั้นเลย แต่ในการเลื่อนตัวชี้ต้นไม้ไปยังต้นไม้ย่อยขวานั้น จะต้องกระโดดข้ามต้นไม้ย่อยซ้ายไป ซึ่งก็สามารถทำได้โดยอาศัยหลักที่ว่าต้นไม้ย่อยใด ๆ จะมีจำนวนใบ (บิต 1) มากกว่าจำนวนโหนดภายใน (บิต 0) อยู่ 1 เสมอ

การค้นหาจะทำต่อไปเรื่อย ๆ จนกระทั่งตัวชี้ต้นไม้มาชี้ที่บิต 1 ซึ่งหมายถึงถังที่เก็บข้อมูล (ซึ่งอาจจะเป็นถังหุ่นก็ได้) การที่จะรู้ว่าบิต 1 นี้แทนถังหมายเลขใด ก็สามารถทำได้โดยใช้วิธีนับจำนวนบิต 1 ที่พบ ในขณะที่เลื่อนตัวชี้ต้นไม้ไปยังตำแหน่งต่าง ๆ บนตัวแทนแบบเชิงเส้น เมื่อทราบว่าเป็นถังหมายเลขใดแล้ว ก็นำไปเทียบกับสมนัยบิตว่าถังนี้เป็นถังหุ่นหรือไม่ ถ้าไม่ใช่ก็นำค่านั้นไปเป็นดรรชนีเปิดดูค่าในตารางเลขที่อยู่ว่า ถังหมายเลขที่ต้องการมีเลขที่อยู่ของถังเป็นเท่าไร แต่ถ้าเป็นถังหุ่นก็หมายความว่าไม่พบคีย์ที่ต้องการค้นหา

3.2.3.3 ตัวอย่างการแปลงตำแหน่งของลายพิมพ์เป็นคีย์

ตำแหน่งของลายพิมพ์ดีเอ็นเอ ที่ได้จากขั้นตอนที่ 3.2.1 จะถูกนำมาแปลงให้เป็นคีย์ตามความหมายของต้นไม้แบบบีดีเอส เพื่อใช้ในการค้นหาข้อมูลจากต้นไม้นี้ยกตัวอย่างเช่น ลายพิมพ์ดีเอ็นเอรูปแบบหนึ่งมีจำนวน 5 ลายพิมพ์ มีแถบอ้างอิงที่ตำแหน่งบนมาตราส่วนมาตรฐานเป็น 14.0, 12.8, 11.3 และ 9.3 ตำแหน่งของลายพิมพ์บนมาตราส่วนมาตรฐานเป็น 14.7, 13.2, 12.1, 10.3 และ 8.8 เมื่อผ่านขั้นตอนที่ 3.2.1 ตำแหน่งของลายพิมพ์จะถูกแปลงเป็น 15.1, 12.7, 11.0, 8.3 และ 6.1 ตามลำดับ ซึ่งจะไม่สนใจทศนิยมตำแหน่งที่หนึ่ง (ตัดทศนิยมตำแหน่งที่หนึ่งทิ้ง) จะได้ตำแหน่งลายพิมพ์จะเป็น 15, 12, 11, 8 และ 6 ดังนั้นคีย์ตามความหมายของต้นไม้แบบบีดีเอสในตัวอย่างนี้คือ 00001111 00001100 00001011 00001000 00000110

3.2.4 การเปรียบเทียบหารูปแบบลายพิมพ์ที่ใกล้เคียง

สำหรับในกรณีที่ไม่พบรูปแบบลายพิมพ์ที่ต้องการ และถ้าต้องการจะเปรียบเทียบดูว่า จะมีลายพิมพ์ใดที่มีรูปแบบลักษณะที่คล้ายกัน เพื่อประโยชน์ในการตรวจสอบความเกี่ยวข้องทางสายเลือด ซึ่งจากการศึกษาพบว่าผู้ที่มีความเกี่ยวข้องกันทางสายเลือดจะมีรูปแบบของลายพิมพ์ดีเอ็นเอที่ใกล้เคียงกัน หรือรูปแบบลายพิมพ์ที่ต้องการตรวจรู้อาจจะมีบางตำแหน่งของลายพิมพ์ที่หายไปหรือเกินมา ในกรณีเหล่านี้โปรแกรมตรวจรู้จะสามารถทำการเปรียบเทียบหารูปแบบลายพิมพ์ที่ใกล้เคียงได้ โดยการประยุกต์ใช้วิธีแก้ไขชื่อที่สะกดผิดโดยอัตโนมัติ (Bickel, 1987)

ในกรณีที่ใช้ชื่อของบุคคลเป็นคีย์ในการค้นหาข้อมูล แต่ค้นหาไม่พบซึ่งเป็นไปได้ว่าชื่อนั้นอาจจะสะกดผิด หรือเป็นข้อมูลใหม่จริง ๆ แต่บ่อยครั้งพบว่าการค้นหาข้อมูลจากคีย์ที่ระบุไม่พบนั้น เกิดจากความเผลอเรอของผู้ใช้เอง เช่น สะกดชื่อ Addison เป็น Adison เป็นต้น

วิธีที่จะช่วยแก้ไขชื่อที่สะกดผิด สามารถทำได้โดยการค้นหาชื่อที่ใกล้เคียงที่สุด จากบริเวณข้อมูล (data space) ของชื่อที่มีอยู่ทั้งหมด ซึ่งขั้นตอนวิธีที่ใช้นั้นมีพื้นฐานมาจากเรขาคณิต (geometry) ดังนี้

ตัวอักขระ	ค่าน้ำหนัก
A, E, I, N, O, S, T	3
D, H, L, R, U	4
C, F, G, M, P, W	5
B, V	6
K, Q	7
J, X, Y	8
Z	9

ตารางที่ 3.1 แสดงน้ำหนักของตัวอักขระตามวิธีของ Bickel

ชื่อที่ประกอบด้วยตัวอักขระ 26 แบบ (A ถึง Z) จะถูกถ่วงน้ำหนักด้วยค่าต่าง ๆ แสดงดังตารางที่ 3.1 และจะถูกมองเป็นเวคเตอร์ที่มีทิศออกจากจุดกำเนิด ไปยังมุมต่าง ๆ ของรูปสี่เหลี่ยมผืนผ้าขนาด 26 มิติ (26-dimensional rectangle) โดยความยาวด้านของรูปสี่เหลี่ยมผืนผ้าที่สมนัยกับตัวอักขระใด จะสัมพันธ์กับน้ำหนักที่ถ่วงตัวอักขระนั้น เวคเตอร์แต่ละตัวนั้นจะแทนชื่อต่าง ๆ ที่มีอยู่ทั้งหมดในบริเวณข้อมูล เรียกเวคเตอร์เหล่านี้ว่า เวคเตอร์ชื่อ (name vector) ชื่อที่มีการสะกดคล้าย ๆ กันก็จะมีเวคเตอร์ชื่อที่อยู่ใกล้เคียงกัน โดยอาศัยหลักนี้จะสามารถหาชื่อที่ใกล้เคียงที่สุดได้จากการพิจารณามุมระหว่างเวคเตอร์ชื่อที่สะกดผิด กับเวคเตอร์ชื่ออื่นในบริเวณข้อมูลทั้งหมด ซึ่งมุมระหว่างเวคเตอร์ชื่อคู่ใดที่มีค่าน้อยที่สุด ชื่อนั้นจะเป็นชื่อที่ใกล้เคียงที่สุด

มุมระหว่างเวคเตอร์สามารถหาได้จาก สูตรการหาอินเนอร์โปรดักส์ (inner product) ของเวคเตอร์ ซึ่งมีค่าเท่ากับ

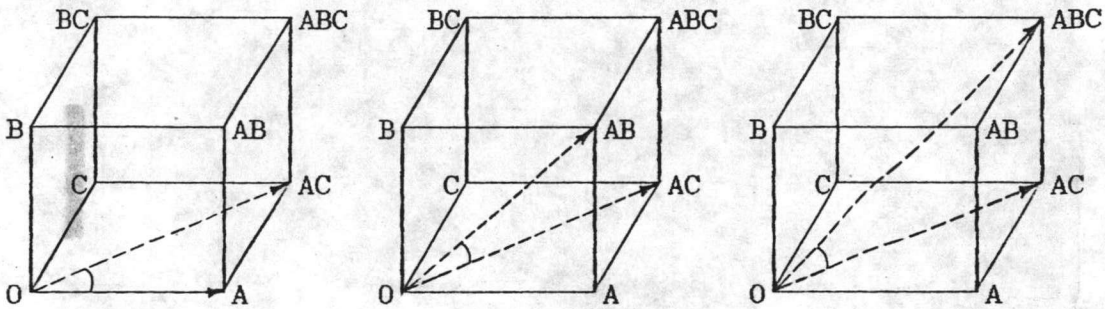
$$U.V = |U||V| \cos(a)$$

เมื่อ a เป็นมุมระหว่างเวคเตอร์ U, V

$|U|$ และ $|V|$ เป็นขนาดของเวคเตอร์ U, V ตามลำดับ

ดังนั้น $a = \arccos \left[\frac{U.V}{|U||V|} \right]$

เพื่อความเข้าใจจะยกตัวอย่างในกรณีที่ง่ายดังนี้ สมมติว่ามีตัวอักขระอยู่เพียง 3 แบบคือ A, B และ C และแต่ละตัวมีน้ำหนักที่ถ่วงเป็น 1 ซึ่งจะทำให้ปัญหาที่สนใจลดรูปเป็น 3 มิติ รูปสี่เหลี่ยมผืนผ้าก็จะเป็นรูปสี่เสี่ยมลูกบาศก์, ถ้าให้ AC เป็นชื่อที่สะกดผิด และบริเวณข้อมูลประกอบด้วยชื่อ A, AB และ ABC ดังนั้นคู่ของเวคเตอร์ที่ต้องพิจารณาคือ (AC, A), (AC, AB) และ (AC, ABC) พิจารณารูปที่ 3.7 ซึ่งแสดงการหามุมระหว่างเวคเตอร์คู่ต่าง ๆ จะเห็นว่าชื่อที่ใกล้เคียงควรจะเป็น ABC ทั้งนี้เพราะว่าให้ค่ามุมระหว่างเวคเตอร์ที่น้อยที่สุด



มุม(AC,A)=45 มุม(AC,AB)=60 มุม(AC,ABC)≅35

รูปที่ 3.7 แสดงมุมระหว่างเวคเตอร์ชื่อคู่ต่าง ๆ

สำหรับการประยุกต์วิธีนี้เพื่อใช้ในงานวิจัยได้ทำดังนี้คือ ตำแหน่งของลายพิมพ์ดีเอ็นเอที่ได้หลังจากการทำมาตราส่วนให้เป็นมาตรฐาน และตัดทศนิยมตำแหน่งที่หนึ่งทิ้งไปแล้วนั้น จะถูกมองเป็นตัวอักขระที่สมนัยกับตำแหน่งนั้นเช่น ตำแหน่งที่มีค่า 1 ก็สมนัยกับตัวอักขระ A, ตำแหน่งที่มีค่า 2 ก็สมนัยกับตัวอักขระ B, ตำแหน่งที่มีค่า 3 ก็สมนัยกับตัวอักขระ C ตำแหน่งที่มีค่าอื่น ๆ ก็เป็นไปในทำนองเดียวกัน เช่นจากตัวอย่างในข้อ 3.2.3.3 ได้ตำแหน่งของลายพิมพ์เป็น 15, 12, 11, 8 และ 6 ก็จะเป็นชุดของตัวอักขระ OLKHF ดังนี้เป็นต้น