การประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่

นายธีรภัทร์ ชุนเดชสัมฤทธิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561

Approximation of Positive Integer In Double-Base Number System with A Single Term

Mr. Teerapat Chundetsumrit

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2018

หัวข้อวิทยานิพนธ์	การประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่
โดย	นายธีรภัทร์ ชุนเดชสัมฤทธิ์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

.... คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ

(รองศาสตราจารย์ ดร.เศรษฐา ปานงาม)

.... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

.... กรรมการภายนอกมหาวิทยาลัย

(รองศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง)

ธีรภัทร์ ชุนเดชสัมฤทธิ์ : การประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่. (Approximation of Positive Integer In Double-Base Number System with A Single Term) อ.ที่ปรึกษาหลัก : ผศ. ดร.อรรถสิทธิ์ สุรฤกษ์

ในระบบจำนวนฐานคู่นั้นเป็นระบบที่ออกแบบมาสำหรับการคำนวณแบบขนานซึ่งเป็นข้อดีในการดำเนินการทางคณิตศาสตร์ต่าง ๆ โดยเฉพาะอย่างยิ่งสำหรับการคูณและการบวก ซึ่งได้มีการนำเสนอคุณสมบัติเอกซ์โพเนนเชียลต่าง ๆ และกฏเกณฑ์เอกลักษณ์ต่าง ๆ เพื่อลดความซับซ้อนของการดำเนินการทางคณิตศาสตร์ต่าง ๆ แต่ทว่าการหารนั้นจะมีข้อด้อยหลาย ๆ ด้านในกรณีที่ตัวหารนั้นมีจำนวนหลายพจน์

โดยงานวิจัยนี้ ผู้วิจัยได้พิจารณาการหารูปแบบแทนจำนวนด้วยพจน์เดี่ยวสำหรับจำนวนใด ๆ ซึ่งผู้วิจัยได้นำเสนออัลกอริทึมการประมาณค่าเพื่อลดจำนวนพจน์โดยเฉพาะสำหรับจำนวนเต็ม รวมถึงยังได้นำเสนออัลกอริทึมการหารทั้งหมดพร้อมบทพิสูจน์ความถูกต้อง และสุดท้ายการใช้ตารางคำนวณค่าล่วงหน้าแสดงให้เห็นว่าสามารถลดเวลาการคำนวณได้ ซึ่งผู้วิจัยคาดหวังว่าแนวคิดในงานวิจัยนี้จะเป็นประโยชน์สำหรับงานวิจัยที่เกี่ยวข้องกับการออกแบบอุปกรณ์ฮาร์ดแวร์ หรือสถาปัตยกรรมทางคอมพิวเตอร์ในอนาคต

สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์	ลายมือชื่อนิสิต ....
ปีการศึกษา	2561	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

5971009621 : MAJOR COMPUTER SCIENCE
KEYWORD: DOUBLE-BASE NUMBER SYSTEM, DIVISION, APPROXIMATION ALGORITHM, OPERATION TIME

Teerapat Chundetsumrit : Approximation of Positive Integer In Double-Base Number System with A Single Term. Advisor: Asst. Prof. Dr. Athasit Surarerks

The double-base number system (DBNS) designed for parallel computing which has an advantage in arithmetic operations especially for multiplication and addition. Many exponential properties and identity rules had been proposed in order to simplify arithmetic operations. Unfortunately, division seem to have many disadvantages in the case that a denominator contains many terms.

In this work, a single term representation for any number is investigated. We proposed an approximation algorithm for reducing a number of terms, especially for an integer. We also proposed a division algorithm together with the proof of correctness. Finally, reduction of an operation time is shown to be realized using a preprocessing table. We expect the idea of this thesis may be useful for future works which are related to some researches in hardware design or computer architecture.

Field of Study: Computer Science		Student's Signature ....
Academic Year: 2018		Advisor's Signature ....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลือของ ผู้ช่วยศาสตราจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษา ซึ่งเป็นผู้ให้แนวทาง คำแนะนำ และข้อคิดเห็นต่างๆ รวมถึงการให้คำปรึกษา อีกทั้งยังช่วยแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการทำวิจัยจนกระทั่งวิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วง ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์ เป็นอย่างสูงที่ให้ความอนุเคราะห์และความเมตตาช่วยเหลือ

ขอขอบพระคุณ รองศาสตราจารย์ ดร. เศรษฐา ปานงาม ประธานกรรมการสอบวิทยานิพนธ์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และ รองศาสตราจารย์ ดร. อานนท์ รุ่งสว่าง กรรมการสอบวิทยานิพนธ์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ที่ได้ให้คำแนะนำในการแก้ไขปรับปรุงและเพิ่มเติมเพื่อให้วิทยานิพนธ์มีคุณภาพดียิ่งขึ้น

สุดท้ายนี้ผู้วิจัยขอขอบพระคุณบิดามารดา ซึ่งเปิดโอกาศให้ได้รับการศึกษาเล่าเรียนรวมถึงเป็นกำลังใจที่สำคัญ และขอขอบคุณเพื่อนกอล์ฟ เพื่อน ๆ พี่ ๆ และน้อง ๆ ทุกคนที่ได้ให้ความช่วยเหลือในทุก ๆ ด้านจนวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

ธีรภัทร์ ชุนเดชสัมฤทธิ์

สารบัญ

หน้า

สารบัญตาราง

สารบัญภาพ

หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในระบบจำนวนฐานคู่ (Double-Base Number System) ซึ่งได้มีการเสนอในงานวิจัยของดิมิทรอฟและคณะ [1] ในปี ค.ศ. 1997 นั้นมีการใช้ประโยชน์ของรูปแบบแทนจำนวนเพื่อประยุกต์สำหรับการเพิ่มประสิทธิภาพการคำนวณในงานวิจัยส่วนใหญ่ 2 ประเภท ได้แก่วิทยาการเข้ารหัสลับ (Cryptography) ซึ่งสามารถติดตามได้ในงานวิจัยดังนี้ [2-12] และงานวิจัยที่เกี่ยวกับการประมวลผลสัญญาณ (Signal processing) โดยในระบบจำนวนฐานคู่นั้นมีข้อดีในเรื่องของการดำเนินการคูณที่มีประสิทธิภาพโดยอาศัยคุณสมบัติของเอกซ์โพเนนเชียล (Exponential property) และคุณสมบัติการกระจาย (Distribution property) ซึ่งทั้งในปี ค.ศ. 2004 งานวิจัยของเบิร์ธและอิมเบิร์ท [13] ได้มีการเสนอวิธีการหารูปแบบแทนจำนวนเพียงพจน์เดียวที่มีค่ามากที่สุดซึ่งไม่เกินกว่าค่าที่กำหนดให้ในระบบจำนวนฐานคู่ ซึ่งเป็นการเพิ่มประสิทธิภาพเชิงเวลาของอัลกอริทึมหารูปแบบแทนจำนวนฐานคู่ แต่อย่างไรก็ตามเมื่อทำการแปลงจำนวนใด ๆ ให้อยู่ในรูปแบบแทนจำนวนฐานคู่จำนวนพจน์ที่เกิดจากการแปลงนั้นจะประกอบไปด้วย 2 กรณีหลัก ๆ ได้แก่พจน์เดียว (Single Digit) หรือหลายพจน์ (Multi-Digit) โดยการดำเนินการคูณในระบบจำนวนฐานคู่จะสามารถดำเนินการได้อย่างมีประสิทธิภาพ ไม่ว่าทั้งสองจำนวนจะอยู่ในรูปแบบพจน์เดียวหรือหลายพจน์ แต่ทว่าสำหรับการดำเนินการหารนั้นจะดำเนินการได้ก็ต่อเมื่อตัวหารอยู่ในรูปแบบพจน์เดียวเท่านั้น โดยในงานวิจัยของมิชราและดิมิทรอฟ [14] ปี ค.ศ. 2007 ได้มีการนำเสนอทฤษฎีกราฟเพื่อเป็นเครื่องมือในการทำความเข้าใจเกี่ยวกับระบบจำนวนฐานคู่ อีกทั้งยังได้กล่าวถึงเงื่อนไขของกระบวนการหารที่ตัวหารนั้นจำเป็นต้องอยู่ในรูปแบบพจน์เดียว นั่นหมายความว่าเราจะต้องหารูปแบบแทนจำนวนของจำนวนใด ๆ ด้วยพจน์เดียว ซึ่งในปี ค.ศ. 1999 งานวิจัยของดิมิทรอฟ, จูเลี่ยนและมิลเลอร์ [15] ได้นิยามความหมายของรูปแบบบัญญัติ (Canonic form) คือจำนวนพจน์ซึ่งไม่ใช่ศูนย์ที่น้อยที่สุด รวมถึงมีการนำเสนอเซตของเอกลักษณ์เพื่อลดจำนวนพจน์ในระบบจำนวนฐานคู่ให้อยู่ในรูปแบบบัญญัติโดยอาศัยความสัมพันธ์ของเลขชี้กำลังพจน์ในสองมิติ แต่อย่างไรก็ตามเอกลักษณ์ที่ได้นำเสนอมานั้น ทำให้เกิดจำนวนพจน์ที่น้อยที่สุดแต่ไม่ใช่จำนวนพจน์เพียงพจน์เดียว ซึ่งในทางปฏิบัติแล้วสำหรับบางจำนวนจะไม่สามารถแสดงให้อยู่รูปพจน์เดียวได้เสมอ ดังนั้นวิธีการแก้ไขปัญหาคือการประมาณค่าจำนวนเต็มบวกใด ๆ ให้อยู่ในรูปพจน์เดียวแทน ทั้งนี้ปัญหาข้างต้นที่ได้กล่าวมานั้น จึงเป็นเหตุผลและแรงบรรดาลใจในงานวิจัยนี้ ซึ่งผู้วิจัยได้นำเสนอทฤษฎีที่รับรองกระบวนการประมาณจำนวนเต็มบวกใด ๆ

ให้อยู่ในรูประบบจำนวนฐานคู่เพียงพจน์เดียวด้วยอัลกอริทึมประมาณค่าซึ่งได้ปรับแต่งจากงานวิจัย [16] ในปี ค.ศ. 2007 ที่ได้เสนออัลกอริทึมการประมาณค่าเพื่อลดการเรียกใช้ตารางเรียกดูค่าซึ่งเป็นข้อด้อยของระบบจำนวนลอการิทึมมิติคู่ (Double Dimensional Logarithmic Number System) โดยผลลัพธ์จากการประมาณค่าจำนวนเต็มบวกใด ๆ จะอยู่ในรูปแบบจำนวนฐานคู่เพียงพจน์เดียว ซึ่งทำให้การดำเนินการเชิงตัวเลขของการหารนั้นมีประสิทธิภาพมากขึ้นและผู้วิจัยยังได้นำเสนออัลกอริทึมการหารจากผลลัพธ์ที่เกิดจากการประมาณค่าด้วยพจน์เดียว รวมถึงนำผลลัพธ์ของการประมาณค่าไปวิเคราะห์เพื่อนำเสนออัลกอริทึมการดำเนินการทางคณิตศาสตร์อื่น ๆ เพิ่มเติม ได้แก่ การคูณและการหาเศษเหลือ นอกจากนี้ยังได้เสนอวิธีลดเวลาการดำเนินการของอัลกอริทึมประมาณค่าด้วยการสร้างตารางค่าไว้ล่วงหน้า โดยให้มีค่าความซับซ้อนเชิงพื้นที่ (Space Complexity) น้อยที่สุดและมีประสิทธิภาพในทางปฏิบัติจริงมากที่สุด

1.2 วัตถุประสงค์ของงานวิจัย

1) นำเสนออัลกอริทึมในการประมาณจำนวนเต็มบวกใด ๆ ด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียว
2) นำเสนออัลกอริทึมการหาร, การคูณและการหาเศษเหลือจากการประมาณค่าจำนวนเต็มบวกด้วยพจน์เดียว
3) เสนอแนวทางในการลดเวลาของการดำเนินการของอัลกอริทึมในหัวข้อที่ 1) ด้วยการใช้ตารางเก็บค่าล่วงหน้า

1.3 ขอบเขตของงานวิจัย

1) ใช้จำนวนเต็มบวกใด ๆ ในการประมาณค่าด้วยรูปแบบจำนวนฐานคู่เพียงพจน์เดียว
2) นำเสนออัลกอริทึมการหาร, การคูณและการหาเศษเหลือ ซึ่งใช้การประมาณค่าด้วยพจน์เดียวพร้อมบทพิสูจน์
3) เสนออัลกอริทึมสำหรับการสร้างตารางเก็บค่าไว้ล่วงหน้าเพื่อลดเวลาการทำงานของอัลกอริทึมประมาณค่า

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1) ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องกับระบบจำนวนฐานคู่
2) ปรับปรุงอัลกอริทึมในงานวิจัยก่อนหน้าให้เหมาะสมกับงานวิจัย
3) นำเสนอทฤษฎีที่ใช้ในการสนับสนุนงานวิจัยพร้อมกับแสดงบทพิสูจน์

4) นำผลลัพธ์จากการประมาณค่ามาออกแบบอัลกอริทึมในการหาร และประยุกต์ในการออกแบบอัลกอริทึมในการคูณและการหาเศษเหลือ
5) ออกแบบการทดลองเพื่อวิเคราะห์เวลาการดำเนินการของอัลกอริทึมในการประมาณค่า
6) สรุปผลการทำงานและปรับปรุงแก้ไข
7) สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1) สามารถประมาณค่าจำนวนเต็มบวกใด ๆ ให้อยู่ในรูปแทนจำนวนฐานคู่เพียงพจน์เดียว
2) ได้วิธีการปรับค่าคลาดเคลื่อนเมื่อนำค่าประมาณเพียงพจน์เดียวไปใช้ในอัลกอริทึมการดำเนินการทางคณิตศาสตร์

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์ในผลงานวิชาการในหัวข้อเรื่องดังนี้

1) "Approximation of Positive Integer in Double-Base Number System with A Single Term" โดย ธีรภัทร์ ชุนเดชสัมฤทธิ์ และอรรถสิทธิ์ สุรฤกษ์ ในงานประชุมวิชาการระดับนานาชาติ 2nd European Conference on Electrical Engineering & Computer Science (EECS 2018)

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

สำหรับทฤษฎีและงานวิจัยที่เกี่ยวข้องในงานวิจัยนี้จะประกอบไปด้วย 2 กลุ่มใหญ่ ๆ คือ ระบบจำนวนฐานคู่ (Double-Base Number System) และระบบจำนวนลอการิทึมมิติคู่แบบขยาย (An Extended Double Dimensional Logarithmic Number System)

2.1 ระบบจำนวนฐานคู่

ในงานวิจัย [1] ปี ค.ศ. 1997 ได้นำเสนอทฤษฎีความสมบูรณ์ของระบบจำนวนฐานคู่ดังทฤษฎีที่ 2.1 พร้อมบทพิสูจน์

ทฤษฎีบทที่ 2.1 สำหรับจำนวนเต็มบวกใด ๆ สามารถแสดงให้อยู่ในรูปแบบ $\sum_{i=1}^{\mathbb{N}} 2^{x_i}3^{y_i}$ ได้เสมอ โดยที่ $x, y \in \mathbb{Z}$

ทั้งนี้การพิสูจน์ทฤษฎีบทที่ 2.1 นั้นผู้วิจัยยังได้นำเสนออัลกอริทึมเชิงละโมบในการคำนวณหารูปแบบจำนวนฐานคู่ได้ดังอัลกอริทึมที่ 2.1

อัลกอริทึมที่ 2.1 อัลกอริทึมในการแปลงจำนวนเต็มบวกใด ๆ ให้อยู่ในระบบจำนวนฐานคู่

Algorithm2.1(n)

input: positive integer n

output: $\sum_{i=1}^{\mathbb{N}} 2^{x_i}3^{y_i}$ that equal to n

(Represented by Array A and Array B ; $x, y \in \mathbb{Z}_0^+$)

Array A: Array of exponents with base 2

Array B: Array of exponents with base 3

begin

1: **if** ($n > 0$) **then do**

2: **Find the largest 2-integer (x,y) that** $2^x 3^y \le n$

3: Insert value a to Array A

4: Insert value b to Array B

5: $n \leftarrow n - 2^x 3^y$

6: Algorithm2.1 (n)

End

ซึ่งจะสังเกตุได้ว่ารูปแบบแทนจำนวนที่เกิดจากอัลกอริทึมที่ 2.1 นั้นจะประกอบไปด้วยพจน์แรกที่มีค่ามากที่สุดและพจน์ถัดไปที่มีค่าลดลงเรื่อย ๆ โดยต่อมาในงานวิจัย [13] ปี ค.ศ. 2004 ได้นำเสนออัลกอริทึมเพื่อคำนวณหารูปแบบแทนจำนวนฐานคู่แบบพจน์เดียวที่มีค่ามากที่สุดซึ่งไม่เกินจำนวนเต็มบวกที่กำหนดมาให้ ซึ่งเป็นการเพิ่มประสิทธิภาพเชิงเวลาของการคำนวณในบรรทัดที่ 2 ของอัลกอริทึมที่ 2.1 และยังเป็นการเพิ่มประสิทธิภาพเชิงเวลาของการคำนวณในอัลกอริทึมหารูปแบบแทนจำนวนฐานคู่ด้วยและในปี ค.ศ. 2007 งานวิจัย [14] ได้นำเสนอทฤษฎีเกี่ยวกับกราฟแบบไม่มีทิศทางและกราฟแบบมีทิศทางเพื่อเป็นเครื่องมือสำหรับวิเคราะห์เกี่ยวกับระบบจำนวนฐานคู่และยังขยายไปจนถึงระบบจำนวนหลายฐาน (Multi-Base Number System) โดยเราสามารถแสดงรูปแทนจำนวนของระบบจำนวนฐานคู่โดยอาศัยกราฟสองส่วน (Bipartite graphs) ซึ่งจะเรียกว่ากราฟระบบจำนวนฐานคู่ (DBNS-graphs)

นิยามที่ 2.1.1 กำหนดให้ n เป็นจำนวนนับใด ซึ่ง $n = 2^{a_1}3^{b_1} + 2^{a_2}3^{b_2} + \ldots + 2^{a_k}3^{b_k}$ เป็นรูปแบบแทนจำนวนในระบบจำนวนฐานคู่

เราสามารถแสดงจำนวน n ด้วยกราฟระบบจำนวนฐานคู่ D_n โดยคุณสมบัติดังนี้

$\bar{a} = \max_{1 \le i \le k}\{a_i\}$ และ $\bar{b} = \max_{1 \le i \le k}\{b_i\}$ จะได้ว่าเซตจุดยอดของ D_n คือ

$V = \{1, 2, \ldots, 2^{\bar{a}}\} \cup \{1, 3, \ldots, 3^{\bar{b}}\}$

และเซตของเส้นเชื่อมคือ $E = \{(2^{a_1}, 3^{b_1}), \ldots, (2^{a_k}, 3^{b_k})\}$

ตัวอย่างที่ 2.1.1 กำหนดให้จำนวนเต็มบวก $n = 2^2 3^7 + 2^7 3^0 + 2^2 3^1$ ซึ่งมีค่าเท่ากับ 8,888

จากนิยามก่อนหน้า จะได้ว่า $\bar{a} = \max_{1 \le i \le k}\{a\} = 7$ และ $\bar{b} = \max_{1 \le i \le k}\{b_i\} = 7$

เซตจุดยอดของ D_n คือ $V = \{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7\} \cup \{1, 3, 3^2, 3^3, 3^4, 3^5, 3^6, 3^7\}$

และเซตของเส้นเชื่อมคือ $E = \{(2^2, 3^7), (2^7, 3^0), (2^2, 3^1)\}$

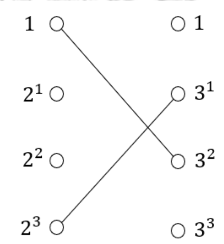
ตัวอย่างที่ 2.1.2 กำหนดให้ $m = 2^6 3^2 + 2^1 3^3 + 2^0 3^2$ ซึ่งมีค่าเท่ากับ 639

จะได้ว่า $\bar{a} = \max_{1 \le i \le k}\{a\} = 6$ และ $\bar{b} = \max_{1 \le i \le k}\{b_i\} = 3$

เซตจุดยอดของ D_m คือ $V = \{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6\} \cup \{1, 3, 3^2, 3^3\}$

และเซตของเส้นเชื่อมคือ $E = \{(2^6, 3^2), (2^1, 3^3), (2^0, 3^2)\}$

สำหรับการแสดงรูปแบบแทนจำนวนที่ทุกพจน์มีค่ามากกว่าศูนย์นั้นสามารถแสดงให้เห็นด้วยกราฟแบบไม่มีทิศทางเพื่อเป็นตัวอย่างดังรูปที่ 1



รูปที่ 1 เลขจำนวน $33 = 2^3 3^1 + 2^0 3^2$ ที่ถูกนำเสนอด้วยกราฟแบบไม่มีทิศทางในระบบจำนวนฐานคู่

และสำหรับการแสดงค่าจำนวนที่ติดลบจะนำเสนอด้วยกราฟแบบมีทิศทาง โดยการแสดงรูปแบบเส้นเชื่อมจะใช้เส้นเชื่อมผกผัน ดังตัวอย่างที่ 2.1.3

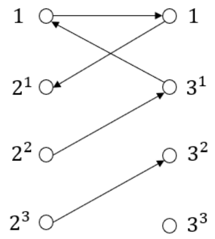
ตัวอย่างที่ 2.1.3 กำหนดให้ $k = 2^6 3^2 - 2^1 3^3 - 2^0 3^2$ ซึ่งมีค่าเท่ากับ 513

จะได้ว่า $\bar{a} = \max_{1 \le i \le k}\{a\} = 6$ และ $\bar{b} = \max_{1 \le i \le k}\{b_i\} = 3$

เซตจุดยอดของ D_k คือ $V = \{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6\} \cup \{1, 3, 3^2, 3^3\}$

และเซตของเส้นเชื่อมคือ $E = \{(2^6, 3^2), (3^3, 2^1), (3^2, 2^0)\}$

โดยสามารถสามารถแสดงให้เห็นด้วยกราฟแบบมีทิศทางเพื่อเป็นตัวอย่างดังรูปที่ 2.2



รูปที่ 2 เลขจำนวน $80 = 2^3 3^2 + 2^2 3^1 - 2^1 3^0 - 2^0 3^1 + 2^0 3^0$ ที่ถูกนำเสนอด้วยกราฟแบบมีทิศทาง ในระบบจำนวนฐานคู่

และผู้วิจัยยังได้เสนอกฎเกณฑ์สำหรับการลดเส้นเชื่อม 2 เส้นที่ซ้ำกันให้เหลือเพียงแค่เส้นเชื่อมเดียวเท่านั้น (D-Rule) โดยผู้วิจัยนั้นได้นำทฤษฎีของกราฟที่นำเสนอก่อนหน้านี้มาเสนอเป็นนิยามของการดำเนินการของจำนวนเต็มภายใต้ระบบจำนวนฐานคู่

การดำเนินการของจำนวนเต็มโดยใช้กราฟระบบจำนวนฐานคู่

กำหนดให้จำนวนเต็ม n_1 และ n_2 เป็นจำนวนที่ถูกแสดงในรูปแบบแทนจำนวน D_{n_1} และ D_{n_2} ด้วยกราฟระบบจำนวนฐานคู่

1) การดำเนินการบวก

นิยามการบวกโดยใช้กราฟระบบจำนวนฐานคู่ สำหรับเซตจุดยอดของ D_{n_1} ซึ่งเท่ากับ $V_1 = \{1, 2, \ldots, 2^{a_1}\} \cup \{1, 3, \ldots, 3^{b_1}\}$ และเซตเส้นเชื่อมเท่ากับ $E_1 = \{(2^{a_{i1}}, 3^{b_{i1}}), \ldots, (2^{a_{ij}}, 3^{b_{ij}})\}$ และสำหรับเซตจุดยอดของ D_{n_2} ซึ่งเท่ากับ $V_2 = \{1, 2, \ldots, 2^{a_2}\} \cup \{1, 3, \ldots, 3^{b_2}\}$ และเซตเส้นเชื่อมเท่ากับ $E_2 = \{(2^{a_{j1}}, 3^{b_{j1}}), \ldots, (2^{a_{jl}}, 3^{b_{jl}})\}$ กำหนดให้ $\max\{a_1, a_2\} = a$ และ $\max\{b_1, b_2\} = b$ จะได้ว่าผลบวกของ D_{n_1} และ D_{n_2} จะสามารถแสดงได้ด้วยกราฟระบบจำนวนฐานคู่ที่มีเซตจุดยอดเป็น $V_1 \cup V_2 = \{1, 2, \ldots, 2^a\} \cup \{1, 3, \ldots, 3^b\}$ และเซตเส้นเชื่อมเท่ากับ $E_1 \cup E_2$ โดยจะขึ้นอยู่กับกฎเกณฑ์ของดี (D-Rule)

2) การดำเนินการลบ

การลบนั้นจะอาศัยกราฟแบบมีทิศทางของระบบจำนวนฐานคู่

นิยามการลบโดยใช้กราฟระบบจำนวนฐานคู่ สำหรับ D_{n_1} และ D_{n_2} สรุปได้ว่า $D_{n_1} - D_{n_2}$ จะเท่ากับ $D_{n_1} + (-D_{n_2})$ (สลับทุกคู่อันดับเส้นเชื่อมใน D_{n_2} เพื่อแสดงถึงค่าที่ติดลบ)

3) การดำเนินการคูณ

นิยาม เครื่องหมายดำเนินการ $SH_{c,d}$ สำหรับจำนวนเต็ม c,d และ $e=(2^a,3^b)$ เป็นเส้นเชื่อม กำหนดให้ $SH_{c,d}(e)=SH_{c,d}((2^a,3^b))=(2^{a+c},3^{b+d})$

นิยามการคูณโดยใช้กราฟระบบจำนวนฐานคู่ สำหรับเซตจุดยอดของ D_{n_1} ซึ่งเท่ากับ $V_1=\{1,2,\ldots,2^{a_1}\}\cup\{1,3,\ldots,3^{b_1}\}$ และเซตเส้นเชื่อมเท่ากับ $E_1=\{(2^{a_{i_1}},3^{b_{i_1}}),\ldots,(2^{a_{i_l}},3^{b_{i_l}})\}$ และสำหรับเซตจุดยอดของ D_{n_2} ซึ่งเท่ากับ $V_2=\{1,2,\ldots,2^{a_2}\}\cup\{1,3,\ldots,3^{b_2}\}$ และเซตเส้นเชื่อมเท่ากับ $E_2=\{(2^{a_{j_1}},3^{b_{j_1}}),\ldots,(2^{a_{j_l}},3^{b_{j_l}})\}$ โดยกำหนดให้ E เป็นเซตของเส้นเชื่อม จะได้ว่า $SH_{c,d}(E)=\{SH_{c,d}(e):e\in E\}$ ดังนั้น $D_{n_1}*D_{n_2}$ จะสามารถแสดงได้ด้วยกราฟระบบจำนวนฐานคู่ที่มีเซตจุดยอดเป็น $V=\{1,2,\ldots,2^{a_1+a_2}\}\cup\{1,3,\ldots,3^{b_1+b_2}\}$ และเซตเส้นเชื่อมเท่ากับ $SH_{a_{i_1},b_{i_1}}(E_2)\cup\ldots\cup SH_{a_{i_l},b_{i_l}}(E_2)$ โดยจะขึ้นอยู่กับกฎเกณฑ์ของดี

4) การดำเนินการหาร

การหารนั้นเป็นการดำเนินการที่จำกัดเนื่องจากสำหรับจำนวนเต็ม n_1 และ n_2 ใด ๆ จะไม่สามารถหาร n_1 ด้วย n_2 ได้เสมอ

โดยเราจะสามารถหาร D_{n_1} ด้วย D_{n_2} ได้ก็ต่อเมื่อสอดคล้องกับเงื่อนไข 2 ข้อด้านล่างนี้

1. D_{n_2} มีเพียงเส้นเชื่อม $E=(2^c,3^d)$ เท่านั้นหรือ n_2 นั้นมีเพียงแค่พจน์ 2^c3^d เดียวในระบบจำนวนฐานคู่
2. D_{n_1} นั้นต้องการถูกแสดงด้วยรูปแบบแทนจำนวนที่ดี ดังนี้

   ถ้าหาก $E_1=\{(2^{a_{i_1}},3^{b_{i_1}}),\ldots,(2^{a_{i_l}},3^{b_{i_l}})\}$

   ดังนั้น $a_{i_k}>c, b_{i_k}>d, \forall k, 1\le k\le l$

หมายเหตุ: สำหรับเงื่อนไขนี้ไม่เป็นจริงสำหรับทุกการหารูปแทนจำนวน n_1 ในกราฟระบบจำนวนฐานคู่ถึงแม้ว่า n_1 จะถูกหารด้วย n_2 ลงตัว

นิยามการหารโดยใช้กราฟระบบจำนวนฐานคู่ ถ้าหากจำนวน n_1 และ n_2 นั้นสอดคล้องกับเงื่อนไขทั้งสองดังข้างบน จะได้ว่า D_{n_1}/D_{n_2} จะสามารถแสดงได้ด้วยกราฟระบบจำนวนฐานคู่ที่มีเซตจุดยอดเป็น $V'=\{1,2,\ldots,2^{a_1-c}\}\cup\{1,3,\ldots,3^{b_1-d}\}$ และเซตเส้นเชื่อมเท่ากับ $SH_{-c,-d}(E_1)$

โดยสำหรับการนำเสนอนิยามของการดำเนินการหารภายใต้ระบบจำนวนฐานคู่จากการวิเคราะห์ด้วยทฤษฎีกราฟนั้นทำให้เราทราบได้ถึงเงื่อนไขของการหารที่ว่าจำนวนรูปแบบแทนจำนวนของตัวหารนั้นควรจะอยู่ในรูปพจน์เดียวเท่านั้น โดยย้อนกลับไปในปี ค.ศ. 1999 งานวิจัย [15] ได้นำเสนอนิยามของรูปแบบบัญญัติ (Canonic form) ดังนิยามที่ 2.1.2

นิยามที่ 2.1.2 รูปแบบบัญญัติ (Canonic form) คือรูปแบบแทนจำนวนซึ่งมีจำนวนพจน์น้อยที่สุดโดยที่แต่ละพจน์ไม่เท่ากับศูนย์

โดยสามารถพิจารณาความเป็นรูปแบบบัญญัติของจำนวนได้ดังตัวอย่างที่ 2.1.4 และ 2.1.5

ตัวอย่างที่ 2.1.4 สำหรับรูปแบบแทนจำนวนฐานคู่ $2^1 3^1 + 2^0 3^1 + 2^0 3^0$ ซึ่งมีค่าเท่ากับ 10 นั้นไม่ใช่รูปแบบบัญญัติเนื่องจากมีรูปแบบแทนจำนวนฐานคู่ $2^3 3^0 + 2^1 3^0$ ที่มีจำนวนพจน์สองพจน์และไม่มีรูปแบบแทนจำนวนแบบอื่นที่จำนวนพจน์น้อยกว่าสองพจน์ได้อีก ดังนั้นรูปแบบบัญญัติของ 10 คือ $2^3 3^0 + 2^1 3^0$

ตัวอย่างที่ 2.1.5 สำหรับรูปแบบแทนจำนวนฐานคู่ $2^5 3^0 + 2^0 3^2$ ซึ่งมีค่าเท่ากับ 41 นั้นเป็นรูปแบบบัญญัติ เนื่องจากเราไม่สามารถนำเสนอจำนวน 41 ด้วยรูปแบบแทนจำนวนเพียงพจน์เดียว

และยังได้นำเสนอเซตของเอกลักษณ์ ดังตัวอย่างที่ 2.1.6 โดยอาศัยความสัมพันธ์ของเลขชี้กำลังบนสองมิติเพื่อลดจำนวนพจน์ของรูปแบบแทนจำนวนฐานคู่ ซึ่งเซตของเอกลักษณ์นั้นจะลดจำนวนพจน์ของรูปแบบแทนจำนวนให้ใกล้เคียงรูปแบบบัญญัติ โดยจำนวนพจน์ที่น้อยที่สุดจะทำให้เวลาการดำเนินการของการคำนวณภายใต้ระบบจำนวนฐานคู่นั้นลดลง

ตัวอย่างที่ 2.1.6 กำหนดให้ i และ j เป็นจำนวนเต็มใด ๆ จะได้ว่าเราสามารถแทนรูปแบบแทนจำนวนหลายพจน์ด้วยจำนวนพจน์ที่น้อยลงเมื่อเลขชี้กำลังของรูปแบบแทนจำนวนนั้นสัมพันธ์กับเอกลักษณ์ดังนี้

เอกลักษณ์ที่ 1) $2^i 3^j + 2^i 3^j = 2^{i+1} 3^j$

เอกลักษณ์ที่ 2) $2^i 3^j + 2^{i+1} 3^j = 2^i 3^{j+1}$

เอกลักษณ์ที่ 3) $2^i 3^j + 2^i 3^{j+1} = 2^{i+2} 3^j$

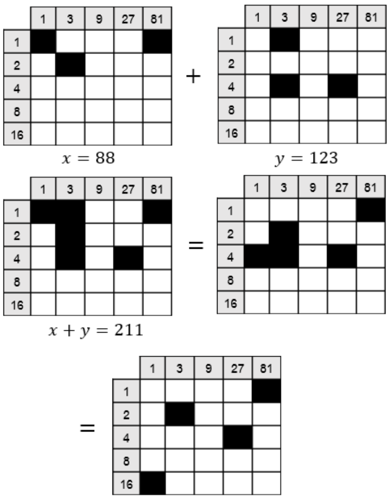
เอกลักษณ์ที่ 4) $2^i 3^j + 2^i 3^{j+1} + 2^{i+1} 3^{j+1} = 2^{i+3} 3^j + 2^{i+1} 3^j$

เอกลักษณ์ที่ 5) $2^{i+1}3^{j} + 2^{i}3^{j+1} + 2^{i+1}3^{j+3} = 2^{i+5}3^{j}$

เอกลักษณ์ที่ 6) $2^{i}3^{j} + 2^{i+4}3^{j} + 2^{i+6}3^{j} = 2^{i}3^{j+4}$

เอกลักษณ์ที่ 7) $2^{i+1}3^{j} + 2^{i}3^{j+2} + 2^{i+4}3^{j} = 2^{i}3^{j+3}$ $\square$

โดยสามารถแสดงให้เห็นถึงตัวอย่างกระบวนการลดจำนวนพจน์หลังจากดำเนินการบวกของสองจำนวนโดยใช้เอกลักษณ์ที่ 3) ได้ดังรูปที่ 3



รูปที่ 3 กระบวนการบวกและการลดจำนวนพจน์ในระบบจำนวนฐานคู่

และนอกเหนือจากงานวิจัยที่เกี่ยวการดำเนินการเชิงตัวเลขภายใต้ระบบจำนวนฐานคู่ก็ยังมีงานวิจัยที่เกี่ยวกับวิทยาการเข้ารหัสลับที่อาศัยประโยชน์ของรูปแบบแทนจำนวนฐานคู่ไปเพิ่มประสิทธิภาพในการคำนวณค่าของจุดคู่อันดับ kP เมื่อ $k \in \mathbb{N}$ ซึ่งถูกแสดงด้วยรูปแบบแทนจำนวนฐานคู่หรือจำนวนสามฐาน และ P คือจุดคู่อับดันดับเริ่มต้นในสมการเส้นโค้งอิลลิปติก (Elliptic Curve) หรือแม้กระทั่งสมการเส้นโค้งแบบอื่น ๆ โดยได้มีงานวิจัยรูปแบบนี้ตั้งแต่ปี ค.ศ. 2005 ในงานวิจัย [11] และมีการพัฒนาอัลกอริทึมให้มีประสิทธิภาพการคำนวณที่มากขึ้นและยังประยุกต์ไปถึงการใช้ระบบจำนวนหลายฐานเข้ามาช่วยการคำนวณ kP ซึ่งสามารถติดตามงานวิจัยได้ดังนี้ [2-10, 12]

2.2 ระบบจำนวนลอการิทึมมิติคู่แบบขยาย

ในงานวิจัย [16] ปี ค.ศ. 2007 ได้มีการเสนอแนวทางแก้ปัญหาในหาค้นหาค่าจากตารางเรียกดูค่าในระบบจำนวนลอกอริทึมมิติคู่เพื่อใช้ในการบวกหรือลบ ดังสมการด่านล่างนี้

$$2^a3^b + 2^c3^d = 2^a3^b(1+2^{c-a}3^{b-d}) \approx 2^a3^b\Phi(c-d,b-d)$$

$$2^a3^b - 2^c3^d = 2^a3^b(1-2^{c-a}3^{b-d}) \approx 2^a3^b\Psi(c-d,b-d)$$

เมื่อกำหนดให้ $\Phi(x,y) = 1+2^x3^y \approx 2^\alpha 3^\beta$ และ $\Psi(x,y) = 1-2^x3^y \approx 2^\gamma 3^\delta$

โดยที่ $x,y,\alpha,\beta,\gamma,\delta \in \mathbb{Z}$

ซึ่งการหาคำตอบของ $\Phi(x,y)$ หรือ $\Psi(x,y)$ จะต้องใช้ค่า x และ y เป็นตัวแปรขาเข้าเพื่อนำไปค้นหาค่าในตารางเรียกดูค่า (Look-up Table) ที่ได้ประมวลผลไว้ โดยตารางเรียกดูค่านั้นต้องมีขนาดใหญ่เพียงพอสำหรับค่า x และ y

ดังนั้นเพื่อหลีกเลี่ยงการใช้ตารางเรียกดูค่า จึงได้มีการนำเสนออัลกอริทึมที่ 2.2 เพื่อประมาณค่าในช่วง $(1,2)$ รวมถึงเสนออัลกอริทึมที่ 2.3 เพื่อปรับปรุงคำตอบของอัลกอริทึมจากประมาณค่าให้คำตอบนั้นมีค่าคลาดเคลื่อนไม่เกินขอบเขตที่เรากำหนดไว้เพื่อที่จะสามารถหาค่าของ $\Phi(x,y)$ หรือ $\Psi(x,y)$ เมื่อ $x,y \in \mathbb{Z}$

อัลกอริทึมที่ 2.2 อัลกอริทึมการประมาณค่าในช่วง (1,2)

Algorithm2.2(x)

input: x where $x \in [1,2)$

output: b,t where $b,t \in \mathbb{Z}$

begin

Define

$a[0] \leftarrow 2^8 3^{-5}$, $a[1] \leftarrow 2^{-3}3^2$, $a[2] \leftarrow 2^5 3^{-3}$,

$a[3] \leftarrow 2^{-6}3^4$, $a[4] \leftarrow 2^2 3^{-1}$, $a[5] \leftarrow 2^{-1}3^1$,

$a[6] \leftarrow 2^7 3^{-4}$, $a[7] \leftarrow 2^{-4}3^3$, $a[8] \leftarrow 2^4 3^{-2}$,

$a[9] \leftarrow 2^{-7}3^5$;

Find the Lower Bound of x

Let s_l be of the form $2^{b_i}3^{t_i}; b,t,i \in \mathbb{Z}$

$s_l \leftarrow a[0]$; $L \leftarrow x - \|s_l\|$; $i \leftarrow 1$

while $L > 0$ **do**

$L \leftarrow x - a[i]$; $i{+}{+}$; $s_l \leftarrow a[i-1]$

enddo

Find the Upper Bound of x

Let s_r be of the form $2^{b_i}3^{t_i};b,t,i \in \mathbb{Z}$

$s_r \leftarrow a[9]$; $R \leftarrow \|s_r\| - x$; $i \leftarrow 8$

while $R > 0$ **do**

$R \leftarrow a[i] - x$; $i--$; $s_l \leftarrow a[i+1]$

enddo

$f_l \leftarrow 2^8 3^{-5}$; $f_r \leftarrow 2^{-8}3^5$

while $|x - \|s_l\|| > |x - \|s_l \times f_l\||$ **do**

$s_l \leftarrow s_l \times f_l$

enddo

while $|x - \|s_r\|| > |x - \|s_r \times f_r\||$ **do**

$s_r \leftarrow s_r \times f_r$

enddo

Select the Closest Estimator s

if $|\|s_l\| - x| < |\|s_r\| - x|$ **then** $b \leftarrow b_l$; $t \leftarrow t_l$

else $b \leftarrow b_r$; $t \leftarrow t_r$

endif

end

อัลกอริทึมที่ 2.3 อัลกอริทึมปรับความแม่นยำคำตอบจากอัลกอริทึมประมาณค่า

input: $x = 2^u 3^v$ where $x \in [1,2)$ and $\|x\| = x$

$y \in [1,2)$

T: terminate condition where $0 < T < 0.5$

output: $z = 2^b 3^t$ the estimation of y

begin

Define initial values

arrays	$2^b 3^t$
a[0]	$2^8 3^{-5}$
a[1]	$2^5 3^{-3}$
a[2]	$2^2 3^{-1}$

a[3]	$2^{7}3^{-4}$
a[4]	$2^{4}3^{-2}$

Let $c = \begin{cases} x & : \ X < y \\ x \times 2^{-8}3^{5} & : \ X > y \end{cases}$

Let $C = \|c\|$

if $X < y$ then $e \leftarrow \dfrac{y - X}{X}; e_n \leftarrow e;$

endif

if $X > y$ then $e \leftarrow \dfrac{X - y}{C}; e_n \leftarrow \dfrac{y - C}{C};$

endif

$o \leftarrow 2^{0}3^{0}$

end ← false

while not.end do

  $i \leftarrow 0$

  while $i < 5$ do

    while $a[i] < 2$ do

      $a[i] \leftarrow a[i] \times 2^{8}3^{-5}$

    enddo

    $a[i] \leftarrow a[i]/2$

    $e_0 \leftarrow \|a[i]\| - 1$

    if $|e_0 - e_n| < e$ then $o \leftarrow a[i]; e \leftarrow |e_0 - e_n|;$

    endif

    if $|e \times c| < T$ then $i \leftarrow 5$; end ← true ;

    else $i{+}{+}$

    endif

  enddo

enddo

$z \leftarrow c \times o$

end

บทที่ 3

การประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่

3.1 บทกล่าวนำ

ในระบบจำนวนฐานคู่นั้นมีข้อดีในเรื่องของการดำเนินการคูณ เมื่อจำนวนสองจำนวนนั้นถูกนำเสนอด้วยรูปแบบแทนจำนวนฐานคู่ โดยใช้อัลกอริทึมเชิงละโมบ (Greedy algorithm) หรืออัลกอริทึมที่ 2.1 ที่ได้มีการเสนอในงานวิจัย [1] และยังมีการใช้อัลกอริทึมเชิงละโมบนี้ในอีกหลาย ๆ งานวิจัย ซึ่งในงานวิจัยนี้จะทำการศึกษาการประมาณค่าในระบบจำนวนฐานคู่โดยใช้อัลกอริทึมที่ 2.1 ในการหารูปแบบแทนจำนวนฐานคู่จากจำนวนเต็มบวก

โดยในงานวิจัยนี้จะแบ่งเป็น 3 ส่วนใหญ่ๆดังนี้

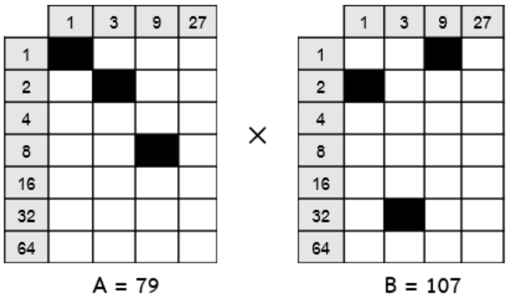
ส่วนที่ 1 การประมาณค่าผลบวกของเทอม 2^x3^y โดยที่ $x, y \in \mathbb{Z}$ ด้วย 2^b3^t เพียงพจน์เดียวเมื่อ $b,t \in \mathbb{Z}$

ส่วนที่ 2 นำเสนออัลกอริทึมการดำเนินการพื้นฐานทางคณิตศาสตร์โดยใช้แนวคิดจากส่วนที่ 1

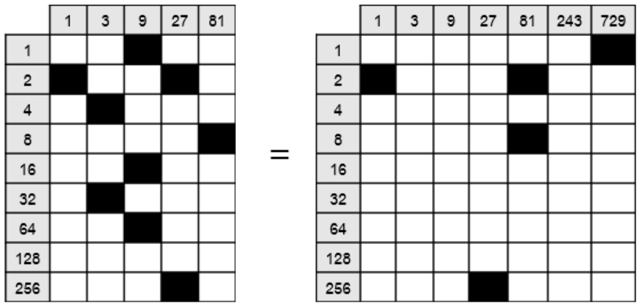
ส่วนที่ 3 ลดเวลาการทำงานในการหาค่าประมาณจากอัลกอริทึมการประมาณค่าโดยการสร้างตารางเก็บค่าไว้ล่วงหน้า

โดยเราขอยกตัวอย่างให้เห็นข้อดีของการคูณและข้อเสียของการหารในระบบจำนวนฐานคู่ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 3.1.1 กำหนดให้ A เป็นจำนวนเต็มบวกและมีค่าเท่ากับ 79 ซึ่งสามารถแทนได้ด้วย $2^3 3^2 + 2^1 3^1 + 2^0 3^0$ และ Bเป็นจำนวนเต็มบวกที่มีค่าเท่ากับ 107 ซึ่งสามารถแทนได้ด้วย $2^5 3^1 + 2^0 3^2 + 2^1 3^0$ และเมื่อนำทั้งสองจำนวนมาดำเนินการคูณกันจะสามารถเห็นได้ดังรูปที่ 4 และรูปที่ 5



รูปที่ 4 ตารางที่แสดงถึงตำแหน่งค่าของแต่ละพจน์ของค่า A และ B



รูปที่ 5 ผลลัพธ์จากการคูณและการลดจำนวนพจน์ของคำตอบ

จากรูปที่ 5 จะเห็นได้ว่าผลลัพธ์การคูณของ A และ B จะได้เป็นรูปแบบแทนจำนวนที่ประกอบไปด้วยจำนวนพจน์ซึ่งเท่ากับจำนวนพจน์คูณจำนวนพจน์ และนอกจากนั้นยังสามารถลดจำนวนพจน์ให้เหลือน้อยลงได้เมื่อเลขชี้กำลังของพจน์นั้นสัมพันธ์กับเอกลักษณ์ซึ่งเป็นข้อดีของการคูณในระบบจำนวนฐานคู่ โดยหลักการในการคูณของระบบจำนวนฐานคู่จะอาศัยสมบัติทางคณิตศาสตร์อยู่ 2 ข้อคือ

1. สมบัติการแจกแจง

$a\times(b+c)=(a\times b)+(a\times c)$ เมื่อ $a,b,c\in\mathbb{R}$

2. สมบัติของเอกซ์โพเนนเชียล

$x^{n}\times x^{m}=x^{n+m}$ เมื่อ $x,n,m\in\mathbb{R}$ □

ในตัวอย่างต่อไปจะเป็นการพิจารณาการดำเนินการหารภายใต้ระบบจำนวนเลขฐานคู่ของการดำเนินการหาร

ตัวอย่างที่ 3.1.2 กำหนดให้ A เป็นจำนวนเต็มบวกและมีค่าเท่ากับ 8,888 ซึ่งสามารถแทนได้ด้วย $2^{2}3^{7}+2^{7}3^{0}+2^{2}3^{1}$ และกำหนดให้ B เป็นจำนวนเต็มบวกที่มีค่าเท่ากับ 2,598 ซึ่งสามารถแทนได้ด้วย $2^{5}3^{4}+2^{1}3^{1}$

เมื่อนำทั้งสองจำนวนมาดำเนินการหารจะได้ว่า $\dfrac{A}{B}=\dfrac{2^{2}3^{7}+2^{7}3^{0}+2^{2}3^{1}}{2^{5}3^{4}+2^{1}3^{1}}$

$$\frac{A}{B}=\frac{2^{2}3^{7}}{2^{5}3^{4}+2^{1}3^{1}}+\frac{2^{7}3^{0}}{2^{5}3^{4}+2^{1}3^{1}}+\frac{2^{2}3^{1}}{2^{5}3^{4}+2^{1}3^{1}}=\frac{1}{2^{3}3^{-3}+2^{-1}3^{-6}}+\frac{1}{2^{-2}3^{4}+2^{-6}3^{1}}+\frac{1}{2^{3}3^{3}+2^{-1}3^{0}}$$

ซึ่งหลักการในการหารของระบบจำนวนฐานคู่จะอาศัยสมบัติทางคณิตศาสตร์อยู่ 2 ข้อคือ

1.สมบัติการแจกแจง

$a\times(b+c)=(a\times b)+(a\times c)$ เมื่อ $a,b,c\in\mathbb{R}$

2.สมบัติของเอกซ์โพเนนเชียล

$$\frac{x^n}{x^m} = \frac{1}{x^{m-n}} \text{ เมื่อ } x,n,m \in \mathbb{R}$$ □

จะเห็นได้อย่างชัดเจนว่าการได้มาซึ่งคำตอบของการดำเนินการหารนั้น ไม่ได้ง่ายเหมือนกับการดำเนินการคูณ ดังนั้นหากว่าเราสามารถหาค่าประมาณของตัวส่วนซึ่งอยู่ในรูปแทนจำนวนฐานคู่ 2^b3^t เมื่อ $b,t \in \mathbb{Z}$ จะทำให้ดำเนินการหารนั้นรวดเร็วมากขึ้น ดังสมการที่ 1 ด้านล่างนี้

$$\frac{A}{B} \approx \frac{\sum_{i=1}^{\mathbb{N}} 2^{x_i}3^{y_i}}{2^b3^t} = \sum_{i=1}^{\mathbb{N}} 2^{x_i-b}3^{y_i-t} \text{ เมื่อ } A,B \in \mathbb{Z}^+ \text{ และ } x_i,y_i,b,t \in \mathbb{Z} \quad (1)$$

โดยใช้สมบัติของเอกซ์โพเนนเชียล $x^n / x^m = 1/x^{m-n}$ เมื่อ $x,n,m \in \mathbb{R}$

3.2 การประมาณค่าผลบวกของเทอม 2^x3^y โดยที่ $x,y \in \mathbb{Z}$ ด้วย 2^b3^t เพียงพจน์เดียวเมื่อ $b,t \in \mathbb{Z}$

สำหรับการประมาณค่าในงานวิจัยนี้ เราจะเริ่มต้นจากการแปลงจำนวนเต็มบวก $\mathbb{N}$ ให้อยู่ในรูปแทนจำนวนฐานคู่โดยใช้อัลกอริทึมที่ 2.1 เนื่องจากอัลกอริทึมมีการทำงานโดยใช้แนวคิดเชิงละโมบ โดยเริ่มจากการหารูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวที่มีค่ามากที่สุดซึ่งมีค่าไม่เกินค่าจำนวนเต็มบวกที่กำหนดให้และนำค่าจำนวนเต็มบวกที่กำหนดให้มาลบออกด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวที่มีค่ามากที่สุดแล้วนำไปเป็นค่าจำนวนเต็มบวกที่กำหนดค่าให้อัลกอริทึมที่ 2.1 และกระทำแบบนี้ซ้ำ ๆ จนกระทั่งค่าจำนวนเต็มบวกที่กำหนดให้อัลกอริทึมเท่ากับศูนย์จึงหยุดการเรียกใช้งานอัลกอริทึมจนท้ายที่สุดก็จะได้เป็น $2^{x_1}3^{y_1} + 2^{x_2}3^{y_2} + \ldots + 2^{x_n}3^{y_n}$ ซึ่งเท่ากับ $\mathbb{N}$ เมื่อ $x,y \in \mathbb{Z}$ และ n เป็นจำนวนเต็มบวก ซึ่งพจน์แรกนั้นมีคุณสมบัติพิเศษคือ $\frac{\mathbb{N}}{2} \le 2^{x_1}3^{y_1} \le \mathbb{N}$ โดยพิสูจน์ได้จากทฤษฎีบทที่ 3.1 โดยทางผู้วิจัยนั้นได้นำเสนอทฤษฎีบท 3 ทฤษฎีซึ่งประกอบไปด้วยทฤษฎีบทที่ 3.1, ทฤษฎีบทที่ 3.2 และทฤษฎีบทที่ 3.3 เพื่อรับรองว่าการประมาณค่าของจำนวนบวกใด ๆ ในงานวิจัยนี้จะสามารถใช้อัลกอริทึมการประมาณค่าในช่วง $(1,2)$ ในงานวิจัย [16] ได้เสมอ โดยตัวอย่างผลลัพธ์ในท้ายที่สุดของการประมาณและปรับค่าด้วยอัลกอริทึมของงานวิจัยนี้คือ $2^{15,162}3^{-9,559}$ มีค่าประมาณเท่ากับ 2,598.4860642336275 ซึ่งเป็นค่าประมาณของ 2,598 (ดังตัวอย่างที่ 3.2.2)

ทฤษฎีบทที่ 3.1) กำหนดให้จำนวนเต็มบวก n จะมีจำนวนเต็ม x และ y ที่ทำให้ $\frac{n}{2} \leq 2^x 3^y \leq n$

พิสูจน์

เริ่มพิจารณาจาก $3^{\log_3 n - 1} = 3^{\log_3 \frac{n}{3}} = \frac{n}{3}$

<u>สมบัติของลอการิทึม</u>

1) $a^{\log_a x} = x$

2) $\log_a \frac{M}{N} = \log_a M - \log_a N$

เมื่อ $a, M, N \in \mathbb{Z}^+$ และ $a \neq 1, k \in \mathbb{R}$

แต่เนื่องจากเลขชี้กำลังของ 3 ต้องเป็นจำนวนเต็มเสมอ จึงได้ว่าเลขชี้กำลังของ 3 คือ $\lfloor \log_3 n \rfloor - 1$

ซึ่ง $\lfloor \log_3 n \rfloor - 1 < \log_3 n - 1$

นั่นคือ $3^{\lfloor \log_3 n \rfloor - 1} \leq \frac{n}{3}$

พิจารณา $\log_3 n - 1 < \lfloor \log_3 n \rfloor$

จะได้ว่า $\log_3 n - 2 < \lfloor \log_3 n \rfloor - 1$

นั่นแสดงว่ามีค่า $\lfloor \log_3 n \rfloor - 1$ ซึ่ง $\frac{n}{3^2} = 3^{\log_3 n - 2} < 3^{\lfloor \log_3 n \rfloor - 1} \leq \frac{n}{3}$

กำหนดให้ j แทนค่า $\lfloor \log_3 n \rfloor - 1$ จะได้ว่า $\frac{n}{9} < 3^j \leq \frac{n}{3}, \frac{2n}{9} < 2 \times 3^j \leq \frac{2n}{3}$

แสดงว่าจำนวน $i = 1, j = \lfloor \log_3 n \rfloor - 1$ จะทำให้ $2^i \times 3^j \in (\frac{2n}{9}, \frac{2n}{3}]$

พิจารณาจะได้ว่า $2^i \times 3^j \in (\frac{2n}{9},\frac{n}{2})$ หรือ $2^i \times 3^j \in [\frac{n}{2},\frac{2n}{3}]$ ซึ่งสำหรับในกรณีนี้จะสรุปได้เลยว่า

จะมีจำนวนเต็ม x และ y ที่ทำให้ $\frac{n}{2} \le 2^x 3^y \le n$

จึงต้องพิสูจน์ในกรณี $2 \times 3^j \in (\frac{2n}{9},\frac{n}{2})$ จะมีค่าจำนวนเต็ม x และ y ที่ทำให้ $\frac{n}{2} \le 2^x 3^y \le n$

พิจารณา $2^2 \times 3^j \in (\frac{4n}{9},1)$ จะสังเกตได้ว่าเมื่อค่านั้นอยู่ในช่วงที่มากกว่าหรือเท่ากับ $\frac{n}{2}$ จะมีค่าจำนวนเต็ม x และ y ที่ทำให้ $\frac{n}{2} \le 2^x 3^y \le n$ แต่หากค่าน้อยกว่า $\frac{n}{2}$ ก็สามารถนำ 2 คูณเพิ่มเข้าไปเพื่อให้ค่าเกิน $\frac{n}{2}$ ได้ ■

ทฤษฎีบทที่ 3.2) กำหนดให้จำนวนเต็มบวก n จะสามารถแปลงให้อยู่ในรูปแบบผลบวกของพจน์ $2^x 3^y$ ได้เมื่อ $x, y \in \mathbb{Z}_0^+$

<u>**พิสูจน์**</u>

จากทฤษฎีบทที่ 3.1 จะได้ว่ามีจำนวนเต็ม x และ y ที่ทำให้ $\frac{n}{2} \le 2^x 3^y \le n$ โดยเลือกค่า x และ y ที่ทำให้ค่า $2^x 3^y$ มากที่สุดและ $2^x 3^y \le n$

หาก $n = 1$ สรุปได้ทันทีว่า $n = 2^0 3^0$

หาก $n > 1$ สรุปได้ทันทีว่าพจน์แรกคือ $n = 2^{x_1} 3^{y_1}$ เมื่อ $x, y \in \mathbb{Z}_0^+$

จากนั้นกำหนดให้ค่าที่เหลือคือ n_1 ซึ่งเท่ากับ $n - 2^{x_1} 3^{y_1}$ และนำค่า n_1 ไปหาพจน์รูปแบบแทนจำนวนฐานคู่ที่มีค่ามากสุดซึ่งไม่เกิน n_1 และหาค่า x และ y ในพจน์ต่อไปจนกระทั่งค่า $n_{k+1} - 2^{x_{k+1}} 3^{y_{k+1}}$ เท่ากับศูนย์ (เมื่อ k เป็นจำนวนเต็มบวก)

จนท้ายที่สุดสามารถเขียนได้เป็น $n = 2^{x_1} 3^{y_1} + 2^{x_2} 3^{y_2} + (n_2 - 2^{x_2} 3^{y_2}) + \ldots + (n_{k+1} - 2^{x_{k+1}} 3^{y_{k+1}})$ ■

ทฤษฎีบทที่ 3.3) กำหนดให้จำนวนเต็มบวก $\mathbb{N}$ ซึ่งแทนด้วย $2^{x_1}3^{y_1}+2^{x_2}3^{y_2}+\ldots+2^{x_k}3^{y_k}$ โดยที่ $x_i, y_i \in \mathbb{Z}$ และ $k \in \mathbb{N}$ ซึ่งแปลงค่าโดยใช้อัลกอริทึมที่ 2.1 เมื่อพิจารณา $2^{x_1}3^{y_1}(1+2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_k-x_1}3^{y_k-y_1})$ จะได้ว่า $1+2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_k-x_1}3^{y_k-y_2} \in (1,2)$

__พิสูจน์__

จากทฤษฎีบทที่ 3.1 กำหนดให้จำนวนเต็มบวก n จะมีจำนวนเต็ม x และ y ที่ทำให้

$$\frac{n}{2} \le 2^x 3^y \le n$$

และทฤษฎีบทที่ 3.2 กำหนดให้จำนวนเต็มบวก n จะสามารถแปลงอยู่ในรูปแบบผลบวกของพจน์ $2^x 3^y$ ได้

($n = 2^{x_1}3^{y_1}+2^{x_2}3^{y_2}+\ldots+2^{x_k}3^{x_k}$ โดยที่ k เป็นจำนวนเต็มบวก) เมื่อ $x, y \in \mathbb{Z}_0^+$ จากทฤษฎีบทที่ 3.1

จะได้ว่า $2^{x_2}3^{y_2}+\ldots+2^{x_k}3^{x_k} \in [0,\frac{n}{2})$ และ $2^{x_1}3^{y_1} \in (\frac{n}{2},n]$

กำหนดให้ $a = 2^{x_2}3^{y_2}+\ldots+2^{x_k}3^{y_k}$

เนื่องจาก $2^{x_1}3^{y_1} \in (\frac{n}{2},n]$ ดังนั้น $\frac{a}{2^{x_1}3^{y_1}} \in [0,1)$

นั่นคือ $2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_k-x_1}3^{y_k-y_2} \in [0,1)$

ดังนั้น $1+\frac{a}{2^{x_1}3^{y_1}} \in (1,2)$ ■

โดยแนวคิดหลักของงานวิจัยนี้คือการประมาณค่าจำนวนเต็มบวก n ซึ่งแทนด้วย $2^{x_1}3^{y_1}+2^{x_2}3^{y_2}+\ldots+2^{x_k}3^{y_k}$ เมื่อ $x, y \in \mathbb{Z}_0^+$ และ k เป็นจำนวนเต็มบวกให้อยู่ในรูปแทนจำนวนฐานคู่ $2^b 3^t$ เพียงพจน์เดียวเมื่อ b และ t เป็นจำนวนเต็มโดยที่ $|2^b 3^t - n| < \alpha$ และ $0 < |\alpha| < 1$ โดยใช้วิธีการพิจารณาค่า n ซึ่งเท่ากับ $2^{x_1}3^{y_1}+2^{x_2}3^{y_2}+\ldots+2^{x_k}3^{y_k}$ หรือ $2^{x_1}3^{y_1}(1+2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_k-x_1}3^{y_k-y_1})$ หลังจากนั้นทำการประมาณค่า $1+2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_k-x_1}3^{y_k-y_1}$ ให้อยู่ในรูปแบบแทนจำนวนฐานคู่ $2^w 3^z$ เมื่อ w และ z เป็นจำนวนเต็ม ดังนั้นเราจะได้รูปแทนจำนวนฐานคู่ $2^b 3^t$ เพียงพจน์เดียวนั่นก็คือ $2^{x_1+w}3^{y_1+z}$ เป็น

ค่าประมาณของจำนวนเต็มบวก n โดยเงื่อนไขที่เราจะยอมรับค่า $2^{x_1+w}3^{y_1+z}$ เป็นคำตอบก็ต่อเมื่อ $\left|2^{x_1+w}3^{y_1+z} - n\right| \in (0,1)$

จากทฤษฎีบททั้ง 3 ที่ได้พิสูจน์ไว้ก่อนหน้า แสดงให้เห็นว่าเราสามารถหาค่าประมาณในรูปแทนจำนวนฐานคู่ 2^w3^z ของ $(1+2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_k-x_1}3^{y_k-y_1})\in(1,2)$ ได้เสมอ เมื่อ $w,z\in\mathbb{Z}$ โดยใช้อัลกอริทึมที่ได้ปรับแต่งจากงานวิจัย [16]

โดยสามารถสรุปเป็นขั้นตอนได้อย่างสั้น ๆ ดังนี้

กระบวนการประมาณค่าจำนวนเต็มบวกใด ๆ ให้อยู่ในรูปแทนจำนวนฐานคู่เพียงพจน์เดียวด้วยค่าขอบเขตความคลาดเคลื่อน $\alpha\in(0,1)$ ที่เรากำหนด

ขั้นตอนที่ 1 กำหนดค่าจำนวนเต็มบวก $\mathbb{N}$ ในรูปแบบ $2^{x_1}3^{y_1}+2^{x_2}3^{y_2}+\ldots+2^{x_n}3^{y_n}$ เมื่อ $x,y\in\mathbb{Z}$ โดยแปลงค่าจากอัลกอริทึมที่ 2.1 ซึ่ง $1+2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_n-x_1}3^{y_n-y_1}\in(1,2)$ และกำหนดค่าขอบเขตความคลาดเคลื่อน $\alpha\in(0,1)$

ขั้นตอนที่ 2 ประมาณค่า $1+2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_n-x_1}3^{y_n-y_1}$ ในรูปแทนจำนวนฐานคู่ 2^w3^z เมื่อ $w,z\in\mathbb{Z}$ ด้วยอัลกอริทึมที่ 3.1

ขั้นตอนที่ 3 ปรับปรุงค่า 2^w3^z ให้มีความแม่นยำตามเงื่อนไข $\left|2^{x_1+w}3^{y_1+z}-\mathbb{N}\right|\le\alpha$ เมื่อ $\alpha\in(0,1)$ ด้วยอัลกอริทึมที่ 3.2

กระบวนทำงานของอัลกอริทึมที่ 3.1

ขั้นตอนที่ 1 กำหนดค่าในช่วง $(1,2)$

ขั้นตอนที่ 2 หาค่าขอบเขตล่าง (Lower) และค่าขอบเขตบน (Upper) จากค่าในตารางเริ่มต้นที่ได้กำหนดค่าไว้แล้ว

ขั้นตอนที่ 3 ปรับค่า Lower และ Upper ให้มีความแม่นยำมากที่สุด

ขั้นตอนที่ 4 เลือกค่า Lower หรือ Upper ที่เข้าใกล้ค่าจริงมากที่สุด

อัลกอริทึมที่ 3.1 อัลกอริทึมการประมาณค่าในช่วง $(1,2)$ ในรูปแทนจำนวนฐานคู่เพียงพจน์เดียว

Algorithm3.1(x)

input: x where $x \in [1,2)$

output: w,z where $w,z \in \mathbb{Z}$

begin

Define: Initial Array A

Index	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
Item	$2^{8}3^{-5}$	$2^{-3}3^{2}$	$2^{5}3^{-3}$	$2^{-6}3^{4}$	$2^{2}3^{-1}$	$2^{-1}3$	$2^{7}3^{-4}$	$2^{-4}3^{3}$	$2^{4}3^{-2}$	$2^{-7}3^{5}$

Find the Lower bound of x

L ← A[0]

if (x > L) **then**

  for (i=1 to 9) **do**

    if (A[i] ≥ x) **then** L ← A[i-1] ; **Break**

      endif

endif

Find the Upper bound of x

U ← A[9]

if (x < U) **then**

  for (i=8 to 0) **do**

if (A[i] $\leq$ x) **then** U = A[i+1] ; **Break**

endif

endif

$F_l \leftarrow 2^8 3^{-5}, F_r \leftarrow 2^{-8} 3^5$

while ($|x-L| > |x-F_l \times L|$) **do** $L \leftarrow L \times F_l$

enddo

while ($|x-U| > |x-F_r \times U|$) **do** $U \leftarrow U \times F_r$

enddo

Select the closest value

if ($|L-x| < |U-x|$) **then**

w $\leftarrow$ exponent of Lower bound from base 2

z $\leftarrow$ exponent of Lower bound from base 3

else

w $\leftarrow$ exponent of Upper bound from base 2

z $\leftarrow$ exponent of Upper bound from base 3

endif

end

ตัวอย่างที่ 3.2.1 จงหาค่าประมาณของ 2,598 ด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียว

วิธีทำ 2,598 เมื่อแปลงให้อยู่ในรูปแทนจำนวนฐานคู่โดยอัลกอริทึมที่ 2.1 จะได้เป็น $2^5 3^4 + 2^1 3^1$

นั่นคือ $2^{x_1} 3^{y_1} = 2^5 3^4$ โดยจะทำการประมาณค่า $1 + \frac{2^1 3^1}{2^5 3^4}$ ซึ่งเท่ากับ $1 + 2^{-4} 3^{-3}$ หรือ 1.002314815

ขั้นตอนที่ 1

เริ่มต้นกำหนดค่า L เริ่มต้นคือ $2^8 3^{-5} = 1.0534979$ ซึ่ง $L = 2^8 3^{-5} > 1.002314815$ จึงไม่ต้องหาค่าในตารางที่เข้าใกล้ค่าจริงมากกว่านั้นแล้ว

เริ่มต้นกำหนดค่า U เริ่มต้นคือ $2^{-7} 3^5 = 1.8984375$ ซึ่ง $1.002314815 < 2^{-7} 3^5 = U$ จึงไม่ต้องหาค่าในตารางที่เข้าใกล้ค่าจริงมากกว่านั้นแล้ว

กำหนดค่า $F_l = 2^8 3^{-5}$ และ $F_r = 2^{-8} 3^5$

ขั้นตอนที่ 2

พิจารณา $|x - L| > |x - F_l \times L|$

$$\left|1.002314815 - 2^8 3^{-5}\right| > \left|1.002314815 - F_l \times 2^8 3^{-5}\right|$$

จะได้ว่า $0.05118 > 0.10754$ ซึ่งไม่เป็นจริง

นั่นคือเราไม่ต้องปรับค่า L ด้วยค่า F_l ดังนั้น $L = 2^8 3^{-5}$

ขั้นตอนที่ 3

พิจารณา $|x - U| > |x - F_r \times U|$

$$\left|1.002314815 - 2^{-7} 3^5\right| > \left|1.002314815 - F_r \times 2^{-7} 3^5\right|$$

จะได้ว่า $0.89612 > 0.79972$ ซึ่งเป็นจริง จึงต้องทำการปรับค่า U ด้วย F_r

นั่นคือ $U = F_r \times 2^{-7} 3^5 = 2^{-15} 3^{10}$ และกลับไปทำขั้นตอนที่ 3 ซ้ำ

จนกระทั่ง $U = 2^{-103}3^{65} = 1.0157621$

ซึ่งเมื่อพิจารณา $|x - U| > |x - F_r \times U|$

$$\left|1.002314815 - 2^{-103}3^{65}\right| > \left|1.002314815 - F_r \times 2^{-103}3^{65}\right|$$

จะได้ว่า 0.13445 > 0.03813 ซึ่งไม่เป็นจริง นั่นคือหยุดการปรับค่า U

ดังนั้น $U = 2^{-103}3^{65}$

ขั้นตอนที่ 4

ทำการเปรียบเทียบ $\left|1.002314815 - 2^{8}3^{-5}\right| = 0.0511$ กับ $\left|1.002314815 - 2^{-103}3^{65}\right| = 0.1344$

จึงเลือก $2^{-103}3^{65}$ เป็นคำตอบของการประมาณค่าเนื่องจากมีค่าคลาดเคลื่อนที่น้อยที่สุด

สรุปได้ว่าค่าประมาณของ 2,598 ในรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียว

คือ $2^{x_1}3^{y_1} \times 2^{-103}3^{65} = 2^{5}3^{4} \times 2^{-103}3^{65} = 2^{-98}3^{69} \approx 2,632.9$

แต่เนื่องจากค่าประมาณจากอัลกอริทึมที่ 3.1 นั้นยังมีความแม่นยำไม่เพียงพอสำหรับในงานวิจัยนี้ จึงต้องมีการปรับค่าให้แม่นยำมากขึ้นตามที่เราต้องการโดยที่เรากำหนดค่าคลาดเคลื่อนไม่ให้คำตอบนั้นคลาดเคลื่อนเกินกว่าขอบเขตที่เราต้องการ ตามอัลกอริทึมปรับปรุงค่าที่เราได้ปรังแต่งจากอัลกอริทึมในงานวิจัย [16]

โดยสามารถอธิบายกระบวนการปรับปรุงค่าจากอัลกอริทึมที่ 3.1 ตามค่าคลาดเคลื่อน $\alpha \in (0,1)$ ที่เรากำหนดได้ดังนี้

ขั้นตอนที่ 1 กำหนดค่าจำนวนเต็มบวก $\mathbb{N}$ ให้กับอัลกอริทึมที่ 3.1 และค่าความคลาดเคลื่อน $\alpha \in (0,1)$

ซึ่งจะได้ค่า $2^{x_1+w}3^{y_1+z}$ เป็นคำตอบจากการประมาณ เมื่อ $x, y, w, z \in \mathbb{Z}$

ขั้นตอนที่ 2 ตรวจสอบค่าที่ได้ตามเงื่อนไข $\left|2^{x_1+w}3^{y_1+z} - \mathbb{N}\right| \le \alpha$ โดยที่ $\alpha \in (0,1)$

ถ้าเงื่อนไขเป็นจริงส่งค่า $2^{x_1+w}3^{y_1+z}$ เป็นคำตอบแล้วสิ้นสุดการทำงาน

แต่หากไม่เป็นไปตามเงื่อนไขให้ทำงานต่อในขั้นตอนที่ 3

ขั้นตอนที่ 3 ปรับค่าประมาณ $2^{w}3^{z}$ จากอัลกอริทึมที่ 3.1 ให้มีค่าน้อยกว่า

$$1+2^{x_2-x_1}3^{y_2-y_1}+\ldots+2^{x_n-x_1}3^{y_n-y_1}$$

ขั้นตอนที่ 4 ปรับค่าในตารางเริ่มต้นทั้งหมด

ขั้นตอนที่ 5 กำหนดให้ A เท่ากับค่าประมาณจากอัลกอริทึมที่ 3.1 คูณกับค่าในตารางที่ดีที่สุด

ขั้นตอนที่ 6 ตรวจสอบค่าที่ได้ตามเงื่อนไข $|A-\mathbb{N}|\le\alpha$

ถ้าเงื่อนไขเป็นจริงส่งค่า A เป็นคำตอบแล้วสิ้นสุดการทำงาน

แต่หากไม่เป็นไปตามเงื่อนไขให้ย้อนกลับไปทำงานในขั้นตอนที่ 4

ซึ่งสามารถแสดงได้เห็นได้ดังอัลกอริทึมที่ 3.2

อัลกอริทึมที่ 3.2 อัลกอริทึมในการประมาณจำนวนเต็มบวก $\mathbb{N}$ ในรูปพจน์ $2^{b}3^{t}$ เมื่อ $b,t\in\mathbb{Z}$ โดยแปลงค่าจากอัลกอริทึมที่ 2.1 ซึ่ง $|\mathbb{N}-2^{b}3^{t}|\le\alpha$ และค่า $\alpha\in(0,1)$ คือค่าที่เรากำหนดให้

Algorithm3.2($\mathbb{N}$)

input : $\mathbb{N}=\sum_{i=1}^{n}2^{x_i}3^{y_i}$ and $\alpha\in(0,1)$

output : b,t ; $b,t\in\mathbb{Z}$ where $|\mathbb{N}-2^{b}3^{t}|\le\alpha$

begin

First Approximate

Let $q\leftarrow\dfrac{\mathbb{N}}{2^{x_1}3^{y_1}}$

$w,z\leftarrow$ Algorithm3.1(q)

Check Answer

if ($|2^{x_1+w}3^{y_1+z}-\mathbb{N}|\le\alpha$) **then** $b\leftarrow x_1+w$; $t\leftarrow y_1+z$; **exit**

endif

if ($2^w 3^z < q$ **) then** $C \leftarrow 2^w 3^z$; $E, E_n \leftarrow (q - 2^w 3^z) / 2^w 3^z$; $O \leftarrow 2^0 3^0$

else $C \leftarrow 2^{w-8} 3^{z+5}$; $E \leftarrow (2^w 3^z - q) / C$; $E_n \leftarrow (q - C) / C$; $O \leftarrow 2^0 3^0$

end ← False

Define : Initial Array A

Index	A[0]	A[1]	A[2]	A[3]	A[4]
Item	$2^8 3^{-5}$	$2^{-3} 3^2$	$2^5 3^{-3}$	$2^{-6} 3^4$	$2^2 3^{-1}$

Update All Value in Array

for (i = 0 to 4) **do**

$n \leftarrow \lceil \log_{2^8 3^{-5}} (2 / A[i]) \rceil$

$A[i] \leftarrow A[i] \times 2^{8n-1} 3^{-5n}$

while (not.end) **do**

for (i = 0 to 4) **do**

e_0 ← A[i]-1

if ($|e_0 - E_n| < E$)

$O \leftarrow A[i] ; E \leftarrow |e_0 - E_n|$

endif

if ($|2^{x_1} 3^{y_1} \times E \times C| \le \alpha$)

end ← True ; $b \leftarrow x_1 + o_2 + c_2$; $t \leftarrow y_1 + o_3 + c_3$

endif

Update All Value in Array

for (i = 0 to 4)

$$n \leftarrow \left\lceil \log_{2^8 3^{-5}} (2 / A[i]) \right\rceil$$

$$A[i] \leftarrow A[i] \times 2^{8n-1} 3^{-5n}$$

enddo

end

หมายเหตุ : o_2 คือเลขชี้กำลังฐานสองของค่า O , o_3 คือเลขชี้กำลังฐานสามของค่า O

c_2 คือเลขชี้กำลังฐานสองของค่า C , c_3 คือเลขชี้กำลังฐานสามของค่า C

ตัวอย่างที่ 3.2.2 จงหาค่าประมาณของ 2,598 ในรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวเมื่อกำหนดค่า $\alpha = 0.5$

วิธีทำ 2,598 เมื่อแปลงให้อยู่ในรูปแบบแทนจำนวนฐานคู่โดยอัลกอริทึมที่ 2.1 จะได้เป็น

$2^5 3^4 + 2^1 3^1$

และจะได้ว่า $2^{x_1} 3^{y_1} = 2^5 3^4$

เริ่มต้นกำหนดค่าให้ $q = \dfrac{2,598}{2^5 3^4}$

ขั้นตอนที่ 1 ประมาณค่า q จากอัลกอริทึมที่ 3.1 ซึ่งได้ว่า $w = -103$ และ $z = 65$

ขั้นตอนที่ 2 ตรวจสอบเงื่อนไข $\left| 2^{x_1} 3^{y_1} \times 2^w 3^z - \mathbb{N} \right| \leq \alpha$

แทนค่าเพื่อพิจารณาเงื่อนไข $\left| 2^5 3^4 \times 2^{-103} 3^{65} - 2,598 \right| \leq 0.5$

$34.9 \leq 0.5$ ซึ่งไม่เป็นจริง

จึงต้องทำงานต่อในขั้นตอนที่ 3

ถ้าหากเงื่อนไขเป็นจริง ให้ส่งค่า $2^{x_1+w}3^{y_1+z}$ เป็นคำตอบและหยุดการทำงาน

ขั้นตอนที่ 3 ตรวจสอบค่าประมาณจากอัลกอริทึมประมาณค่าว่ามีค่ามากกว่าหรือน้อยกว่าค่าจริง

ตามเงื่อนไขดังนี้ $2^{w}3^{z} < q$

$2^{-103}3^{65} < q$ ซึ่งไม่เป็นจริง

กำหนดค่า $C = 2^{w-8}3^{z+5}$

$$E = \frac{2^{w}3^{z} - q}{C}$$

$$E_n = \frac{q - C}{C}$$

$$O = 2^{0}3^{0}$$

แทนค่าจะได้เป็น $C = 2^{-103-8}3^{65+5} = 2^{-111}3^{70}$

$$E = \frac{2^{-103}3^{65} - q}{2^{-111}3^{70}}$$

$$E_n = \frac{q - 2^{-111}3^{70}}{2^{-111}3^{70}}$$

$$O = 2^{0}3^{0}$$

ขั้นตอนที่ 4 ทำการปรับทุกค่าในตาราง

จึงได้ค่าในตารางใหม่ดังนี้

$A[0] = 2^{111}3^{-70} = 1.0371502785538707$

$A[1] = 2^{92}3^{-58} = 1.0513003944109913$

$A[2] = 2^{65}3^{-41} = 1.0115288518086079$

$A[3] = 2^{46}3^{-29} = 1.0253294077568396$

$A[4]=2^{27}3^{-17}=1.0393182483438532$

กำหนดค่า $End=False$

ขั้นตอนที่ 5 พิจารณาเงื่อนไข $\sim End=False$ หากเป็นจริงให้ทำต่อในขั้นตอนที่ 6

พิจารณาทุกค่าในตารางตามเงื่อนไขดังนี้

e_0 = A[i]-1 และหากเข้าเงื่อนไข ($\left|e_0-E_n\right|<E$) จะได้ว่า O = A[i]

และ E = $\left|e_0-E_n\right|$

โดยจากการการพิจารณาทุกค่าในตาราง จึงได้ว่า

$O=1.0393182483438532$

$E=0.039551089383856756$

และพิจารณาเงื่อนไข $\left|2^{x_1}3^{y_1}\times E\times C\right|\leq\alpha$

ซึ่ง $\left|2^5 3^4\times E\times 2^{-111}3^{70}\right|\leq 0.5$ ไม่เป็นจริง

จึงต้องทำการปรับทุกค่าในตารางได้ดังนี้

$A[0]=2^{214}3^{-135}=1.0210562897420383$

$A[1]=2^{195}3^{-123}=1.0349868310485855$

$A[2]=2^{176}3^{-111}=1.0491074304185746$

$A[3]=2^{149}3^{-94}=1.0094188494143352$

$A[4]=2^{130}3^{-82}=1.0231906180412405$

แล้วย้อนกลับไปทำซ้ำขั้นตอนที่ 5

จนกระทั่งเงื่อนไข $\left|2^{x_1}3^{y_1}\times E\times C\right|\leq\alpha$ เป็นจริง จึงกำหนดให้

$End=True$

ในตัวอย่างนี้ค่า $E = 0.00019449137937555677$

คือค่าที่ทำให้เงื่อนไข $\left|2^{x_1}3^{y_1} \times E \times C\right| \leq \alpha$ เป็นจริง

ขั้นตอนที่ 6 ส่งค่า $2^{x_1}3^{y_1} \times O \times C$ เป็นคำตอบในรูปของเลขชี้กำลัง ซึ่ง

$O = 2^{15,268}3^{-9,633}$

ดังนั้นคำตอบในประมาณค่า 2,598 ในรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวคือ $2^5 3^4 \times 2^{15,268}3^{-9,633} \times 2^{-111}3^{70} = 2^{15,162}3^{-9,559}$

ซึ่งมีค่าประมาณ 2,598.4860642336275

โดยที่มีค่าคลาดเคลื่อนไม่เกิน 0.5

หลังจากที่ได้ปรับแต่งอัลกอริทึมในการประมาณค่าเพื่อความเหมาะสมสำหรับงานวิจัยนี้ ดังอัลกอริทึมที่ 3.2 เราลองมายกตัวอย่างเพื่อพิจารณาการดำเนินการหารโดยใช้อัลกอริทึมที่ 3.2 ในการประมาณค่าของตัวหารที่เหลือเพียงพจน์เดียวเท่านั้น

ตัวอย่างที่ 3.2.3 กำหนดให้ A เป็นจำนวนเต็มบวกและมีค่าเท่ากับ 8,888 ซึ่งสามารถแทนได้ด้วย $2^2 3^7 + 2^7 3^0 + 2^2 3^1$ และกำหนดให้ B เป็นจำนวนเต็มบวกที่มีค่าเท่ากับ 2,598 ซึ่งสามารถแทนได้ด้วย $2^5 3^4 + 2^1 3^1$ และนำค่า B เป็นค่ากำหนดให้อัลกอริทึมที่ 3.2 เพื่อประมาณค่าในรูปแทนจำนวนฐานคู่เพียงพจน์เดียว โดยกำหนดค่าคลาดเคลื่อน $\alpha = 0.5$ จะได้ว่า $2^{15,162}3^{-9,559}$ เป็นค่าประมาณเพียงพจน์เดียว

ซึ่งคำตอบที่ได้มานั้นมีเลขชี้กำลังที่มากซึ่งยากลำบากต่อการคำนวณโดยตรง จึงได้มีแนวทางแก้ไขปัญหาดังนี้

สำหรับการคำนวณ $2^b 3^t$ เมื่อ $b,t \in \mathbb{Z}$ กำหนดให้ $2^x = 3^t$ และ $x = t\log_2 3$ จะได้ว่า $2^b 3^t = 2^b 2^{t\log_2 3} = 2^{b + t\log_2 3}$

ดังนั้นจากวิธีการคำนวณที่เราได้เสนอในข้างต้น จะได้ว่า $2^{15,162}3^{-9,559} = 2,598.4860642336275$

พิจารณา $\frac{A}{B} \approx \frac{2^2 3^7 + 2^7 3^0 + 2^2 3^1}{2^{15,162}3^{-9,559}} = 2^{-15,160}3^{9,566} + 2^{-15,155}3^{9,559} + 2^{-15,160}3^{9,560} = 3.4204532101762863$

โดยหากเราลองคำนวณผลหารของทั้งสองจำนวนบนระบบจำนวนจริง

นั่นคือ $\frac{A}{B} = \frac{8,888}{2,598} = 3.421093148575828$ ซึ่งจะเห็นได้ว่าการดำเนินการหารที่ใช้การประมาณค่าตัวหารด้วยพจน์เดียวนั้นมีคำตอบที่คลาดเคลื่อนไปจากค่าจริง ดังนั้นเราจึงต้องมีวิธีการแก้ไขค่าความคลาดเคลื่อนในการดำเนินการหารซึ่งใช้การประมาณตัวหารด้วยพจน์เดียวโดยจะพูดถึงในหัวข้อ 3.3

3.3 นำเสนออัลกอริทึมการดำเนินการพื้นฐานทางคณิตศาสตร์โดยใช้การประมาณค่าเพียงพจน์เดียว

ในหัวข้อนี้จะแบ่งเป็นหัวข้อย่อยได้ 3 หัวข้อ ประกอบไปด้วย

1. การนำค่าประมาณจำนวนเต็มบวกด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวมานำเสนออัลกอริทึมการหารในระบบจำนวนฐานคู่ ซึ่งเป็นปัญหาหลักที่เราต้องการแก้ไข
2. การนำค่าประมาณจำนวนเต็มบวกด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวมานำเสนออัลกอริทึมการคูณในระบบจำนวนฐานคู่และวิเคราะห์ค่าความซับซ้อนเชิงเวลา
3. การนำค่าประมาณจำนวนเต็มบวกด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวมานำเสนออัลกอริทึมการหาเศษเหลือในระบบจำนวนฐานคู่

3.3.1 การนำค่าประมาณจำนวนเต็มบวกด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวมานำเสนออัลกอริทึมการหารในระบบจำนวนฐานคู่

เมื่อเราประมาณค่าส่วนในรูปพจน์ 2^b3^t โดยที่ $b,t \in \mathbb{Z}$ เพียงพจน์เดียวนั้น จะเห็นได้ถึงข้อดีของการใช้คุณสมบัติเอกซ์โพเนนเชียล $\frac{2^x3^y}{2^b3^t} = 2^{x-b}3^{y-t}$ เมื่อ $x,y,b,t \in \mathbb{R}$ สำหรับการดำเนินการหารในจำนวนฐานคู่นั้นมีความรวดเร็วแต่จะมีข้อเสียในเรื่องของคำตอบที่ได้นั้นจะมีค่าคลาดเคลื่อนไปจากค่าจริง ทางผู้วิจัยจึงได้ตั้งสมมติฐานที่จะแก้ไขปัญหาการดำเนินการหารให้มีค่าที่แม่นยำหรือผิดพลาดจากค่าจริงน้อยที่สุดจากตัวแปรที่เราทราบก่อนหน้าจากอัลกอริทึมการประมาณค่าและปรับปรุงค่า (อัลกอริทึมที่ 3.2) ซึ่งก็คือค่าความคลาดเคลื่อนจากการประมาณจำนวนเต็มบวก โดยยังคงคุณสมบัติเอกซ์โพเนนเชียล $\frac{2^x3^y}{2^b3^t} = 2^{x-b}3^{y-t}$ เอาไว้

เริ่มต้นการแก้ปัญหาจากสิ่งที่กำหนดให้คือจำนวนเต็มบวก A และ B ซึ่งอยู่ในรูปแบบแทนจำนวน $\sum_{i=1}^{\mathbb{N}} 2^{x_i}3^{y_i}$ เมื่อ $x_i, y_i \in \mathbb{Z}$ โดยที่ B สามารถประมาณได้ด้วย 2^b3^t และกำหนดให้ $\omega = 2^b3^t$ เมื่อ $b,t \in \mathbb{Z}$ และค่าความคลาดเคลื่อน ε คือค่าคลาดเคลื่อนที่เกิดจากการประมาณค่าซึ่งคำนวณได้จาก $B-\omega$ จะได้ว่า $B = \omega + \varepsilon$

เราต้องการที่จะหาค่า $\frac{A}{B}$ หรือ $\frac{A}{\omega+\varepsilon}$

และลองพิจารณา $$\frac{A}{\omega+\varepsilon}=\frac{A}{\omega}-\frac{A\varepsilon}{(\omega+\varepsilon)\omega} \qquad (2)$$

กำหนดให้ $C=\frac{A}{\omega+\varepsilon}$ และแทนค่า C ลงในสมการที่ 2

จะได้เป็น $C=\frac{A}{\omega}-\frac{\varepsilon C}{\omega}$, $C+C\frac{\varepsilon}{\omega}=\frac{A}{\omega}$, $C(1+\frac{\varepsilon}{\omega})=\frac{A}{\omega}$, $C=\frac{\frac{A}{\omega}}{(1+\frac{\varepsilon}{\omega})}$

$$\therefore \frac{A}{B}=\frac{\frac{A}{\omega}}{(1+\frac{\varepsilon}{\omega})} \qquad (3)$$

ดังนั้น $\frac{A}{B}=\frac{\sum_{i=1}^{\mathbb{N}} 2^{x_i-b}3^{y_i-t}}{(1+\frac{\varepsilon}{2^b 3^t})}$ ซึ่งเห็นได้ว่าเรายังคงคุณสมบัติของเอกซ์โพเนนเชียลในเทอม $\sum_{i=1}^{\mathbb{N}} 2^{x_i-b}3^{y_i-t}$ เพียงแค่ทำการเพิ่มการคำนวณค่าของเทอมส่วน $1+\frac{\varepsilon}{2^b 3^t}$ จะทำให้คำตอบของการดำเนินการหารจากการประมาณค่าตัวส่วนด้วยพจน์เดียวตามสมการที่ 3 ที่ได้เสนอนั้นมีความแม่นยำมากกว่าการคำนวณด้วยค่าประมาณเพียงอย่างเดียว

ลองพิจารณานำสมการที่ 3 ที่ได้เสนอก่อนหน้านี้มาแสดงเป็นตัวอย่างการดำเนินการหาร ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 3.3.1 กำหนดให้จำนวนเต็มบวก A ซึ่งแทนได้ด้วย $2^2 3^7+2^7 3^0+2^2 3^1$ และจำนวนเต็มบวก B ซึ่งแทนได้ด้วย $2^5 3^4+2^1 3^1$ และ $2^{15,162}3^{-9,559}$ ซึ่งเท่ากับ 2,598.486064233675305 และเป็นค่าประมาณจากอัลกอริทึมที่ 3.2 โดยกำหนดค่าความคลาดเคลื่อนไว้ไม่เกิน 0.5 และจากคำตอบที่ได้จากการประมาณจะมีค่าความคลาดเคลื่อน $\varepsilon=-0.4860642336275305$

โดยค่าที่ได้คำนวณไว้ในตัวอย่างที่ 3.2.3

คือ $\frac{A}{B}\approx\frac{2^2 3^7+2^7 3^0+2^2 3^1}{2^{15,162}3^{-9,559}}=2^{-15,160}3^{9,566}+2^{-15,155}3^{9,559}+2^{-15,160}3^{9,560}=3.4204532101762863$

หลังจากนั้นทำการนำคำตอบที่เกิดจากการหารโดยใช้ค่าประมาณตัวส่วนเพียงพจน์เดียวมาคำนวนกับค่าตัวส่วนที่เราได้เพิ่มขึ้นมาดังสมการที่ 3 จะได้ว่า

$$\frac{A}{B}=\frac{\frac{2^2 3^7+2^7 3^0+2^2 3^1}{2^{15,162}3^{-9,559}}}{1+(\frac{-0.4860642336275305}{2^{15,162}3^{-9,559}})}=\frac{3.4204532101762863}{1+(\frac{-0.4860642336275305}{2^{15,162}3^{-9,559}})}=3.4210931485782354$$

ซึ่งคำตอบนั้นมีความแม่นยำมากขึ้นเมื่อเทียบกับคำตอบจริงซึ่งเท่ากับ 3.4210931485758276 □

หลังจากที่เราได้เสนอวิธีการปรับค่าความคลาดเคลื่อนจากสมการที่ 3 เราจะทำการเปรียบเทียบค่าความซับซ้อนเชิงเวลาของอัลกอริทึมในการหารแบบเดิมเทียบกับอัลกอริทึมในการหารที่ได้ใช้สมการที่ 3

โดยเริ่มจากตัวอย่างการหารในระบบจำนวนฐานคู่ ดังที่ได้แสดงให้เห็นในตัวอย่างที่ 3.1.2 ซึ่งอาศัยคุณสมบัติของเอกซ์โพเนนเชียลมาใช้ในการหาร จะสามารถแปลงเป็นอัลกอริทึมได้ดังนี้

อัลกอริทึมที่ 3.3.1 อัลกอริทึมในการหารแบบธรรมดาโดยใช้คุณสมบัติของเอกซ์โพเนนเชียล

input: $A = \sum_{i=1}^{\text{length of } A} 2^{x_i} 3^{y_i}$ and $B = \sum_{i=1}^{\text{length of } B} 2^{x_i} 3^{y_i}$

(Represented by ArrayA2 , ArrayA3 , ArrayB2 , ArrayB3)

Array A2: Array of exponents with base 2 of value A

Array A3: Array of exponents with base 3 of value A

Array B2: Array of exponents with base 2 of value B

Array B3: Array of exponents with base 3 of value B

output: w where $w = \frac{A}{B}$

begin

$w \leftarrow 0$

for (i=0 to length of A) **do**

Denominator $\leftarrow 0$

for (j=0 to length of B) **do**

Denominator $\leftarrow$ Denominator $+ 2^{ArrayB2[j]-ArrayA2[i]} \times 3^{ArrayB3[j]-ArrayA3[i]}$

$w \leftarrow w$ + (1/Denominator)

end

โดยค่าความซับซ้อนเชิงเวลาของอัลกอริทึมที่ 3.3.1 คือ $\Theta(l_A \times l_B)$ เมื่อ l_A คือจำนวนพจน์ของจำนวนเต็มบวก A และ l_B คือจำนวนพจน์ของจำนวนเต็มบวก B ซึ่งจะเห็นได้ว่าค่าความซับซ้อนจะอยู่ในรูปผลคูณของจำนวนพจน์ทั้งสองจำนวนในระบบจำนวนฐานคู่ หลังจากนั้นเราลองพิจารณาการหารโดยไม่ต้องอาศัยคุณสมบัติของเอกซ์โพเนนเชียล นั่นคือการหารแบบตรง ๆ โดยการรวมค่าตัวเศษและรวมค่าตัวส่วนแล้วนำค่าผลรวมทั้งสองค่ามาหารกัน ซึ่งสามารถแปลงเป็นอัลกอริทึมได้ดังนี้

อัลกอริทึมที่ 3.3.2 อัลกอริทึมในการหารแบบธรรมดาโดยตรง

input: $A = \sum_{i=1}^{length\ of\ A} 2^{x_i} 3^{y_i}$ and $B = \sum_{i=1}^{length\ of\ B} 2^{x_i} 3^{y_i}$

(Represented by Array A2, Array A3, Array B2, Array B3)

Array A2: Array of exponents with base 2 of value A

Array A3: Array of exponents with base 3 of value A

Array B2: Array of exponents with base 2 of value B

Array B3: Array of exponents with base 3 of value B

output: w where $w = \frac{A}{B}$

begin

Denominator $\leftarrow 0$

for (i=0 to length of B) **do**

Denominator $\leftarrow$ Denominator $+ 2^{ArrayB2[i]} \times 3^{ArrayB3[i]}$

Numerator $\leftarrow 0$

for (i=0 to length of A) **do**

Numerator $\longleftarrow$ Numerator $+ 2^{ArrayA2[i]} \times 3^{ArrayA3[i]}$

$w \longleftarrow$ Numerator /Denominator

end

โดยค่าความซับซ้อนเชิงเวลาของอัลกอริทึมที่ 3.3.2 คือ $\Theta(l_A + l_B)$ เมื่อ l_A คือจำนวนพจน์ของจำนวนเต็มบวก A และ l_B คือจำนวนพจน์ของจำนวนเต็มบวก B ซึ่งจะเห็นได้ว่าค่าความซับซ้อนเชิงเวลาอยู่ในรูปบวกของจำนวนพจน์ทั้งสองจำนวนในระบบจำนวนฐานคู่ ซึ่งเป็นข้อดีในเรื่องของค่าความซับซ้อนเชิงเวลาที่น้อยกว่าอัลกอริทึมที่ 3.3.1 ซึ่งใช้คุณสมบัติของเอกซ์โพเน็นเชียล แต่วิธีการหารแบบตรง ๆ ดังอัลกอริทึมที่ 3.3.2 นั้น ในทางปฏิบัติจริงอาจจะใช้งานได้ไม่ค่อยดีนัก เนื่องจากเราไม่ได้ใช้ข้อดีของรูปแบบแทนจำนวนฐานคู่เพื่อช่วยให้การหารนั้นมีประสิทธิภาพมากขึ้นเลย

หลังจากได้วิเคราะห์ค่าความซับซ้อนเชิงเวลาของการหารโดยใช้คุณสมบัติของเอกซ์โพเนนเชียลและการหารแบบตรง ๆโดยไม่อาศัยคุณสมบัติของเอกซ์โพเนนเชียล เราลองมาวิเคราะห์ค่าความซับซ้อนเชิงเวลาของอัลกอริทึมในการหารโดยการประมาณตัวส่วนและใช้คุณสมบัติของเอกซ์โพเนนเชียลที่อาศัยหลักการจากสมการที่ 3 ซึ่งแปลงเป็นอัลกอริทึมได้ดังนี้

อัลกอริทึมที่ 3.3.3 อัลกอริทึมในการหารโดยการประมาณค่าและใช้คุณสมบัติของเอกซ์โพเน็นเชียล

Algorithm3.3.3(Array A2, Array A3, b, t, ε)

input : $A = \sum_{i=1}^{\text{length of } A} 2^{x_i} 3^{y_i}$, $B \approx 2^b 3^t$ and $\varepsilon \in (0,1)$ where ε = error of

approximate B

(Represented by ArrayA2 , ArrayA3 , b , t , ε)

ArrayA2 : Array of exponents with base 2 of value A

ArrayA3 : Array of exponents with base 3 of value A

output : w where $w = \frac{A}{B}$

begin

Numerator $\leftarrow$ 0

for (i=0 to length of A) **do**

    p $\leftarrow$ ArrayA2[i] - b

    q $\leftarrow$ ArrayA3[i] - t

    Numerator $\leftarrow$ Numerator + $2^{p} \times 3^{q}$

Denominator $\leftarrow$ $1+(\varepsilon / 2^{b}3^{t})$

w $\leftarrow$ Numerator/Denominator

end

โดยค่าความซับซ้อนเชิงเวลาของอัลกอริทึมที่ 3.3.3 คือ $\Theta(l_A)$ เมื่อ l_A คือจำนวนพจน์ของจำนวนเต็มบวก A ซึ่งเห็นได้ว่าค่าความซับซ้อนเชิงเวลาของอัลกอริทึมที่ได้ใช้แนวคิดหลักจากสมการที่ 3 นั้นขึ้นอยู่กับจำนวนพจน์ของตัวเศษเพียงค่าเดียวเท่านั้นและยังคงไว้ซึ่งคุณสมบัติของเอกซ์โพเน็นเชียล ซึ่งแตกต่างจากอัลกอริทึมที่ 3.3.1 ที่ขึ้นอยู่กับจำนวนพจน์ของค่าตัวเศษคูณกับจำนวนพจน์ของค่าตัวส่วน ซึ่งใช้คุณสมบัติของเอกซ์โพเน็นเชียลและอัลกอริทึมที่ 3.3.2 ที่ขึ้นอยู่กับจำนวนพจน์ของค่าตัวเศษบวกกับจำนวนพจน์ของค่าตัวส่วน แต่ไม่ได้อาศัยคุณสมบัติของเอกซ์โพเนนเชียล สรุปว่าอัลกอริทึมที่ 3.3.3 นั้นใช้ค่าความซับซ้อนเชิงเวลาน้อยที่สุดเพื่อคำนวนคำตอบในการหารของสองจำนวนภายใต้ระบบจำนวนฐานคู่รวมถึงยังคงไว้ซึ่งคุณสมบัติของเอกซ์โพเน็นเชียล

3.3.2 การนำค่าประมาณจำนวนเต็มบวกด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวมานำเสนออัลกอริทึมการคูณในระบบจำนวนฐานคู่และวิเคราะห์ค่าความซับซ้อนเชิงเวลา

ลองมาวิเคราะห์อัลกอริทึมในการประมาณค่าไปประยุกต์ใช้กับอัลกอริทึมการดำเนินการคูณจะได้ดังนี้

อัลกอริทึมที่ 3.3.4 อัลกอริทึมในการคูณโดยประมาณค่าและใช้คุณสมบัติของเอกซ์โพเนนเชียล

input: $A \approx 2^h 3^g$, $B \approx 2^b 3^t$, ε_A = error of approximate A

, ε_B = error of approximate B

(Represented by h , g , b , t , ε_A , ε_B)

output: r where $r = A \times B$

begin

$r \leftarrow (2^{h+b} 3^{g+t}) + (2^b 3^t \times \varepsilon_B) + (2^h 3^g \times \varepsilon_A) + \varepsilon_A \varepsilon_B$

end

โดยค่าความซับซ้อนเชิงเวลาของอัลกอริทึมที่ 3.3.4 คือ $\Theta(1)$ ซึ่งเราจะเปรียบเทียบกับอัลกอริทึมการคูณแบบตรง ๆ โดยอาศัยคุณสมบัติของเอกซ์โพเน็นเชียลที่ได้มีการใช้อยู่ในหลาย ๆ งานวิจัย ซึ่งสามารถแสดงเป็นอัลกอริทึมได้ดังนี้

อัลกอริทึมที่ 3.3.5 อัลกอริทึมในการคูณแบบธรรมดาโดยใช้คุณสมบัติของเอกซ์โพเนนเชียล

input: $A = \sum_{i=1}^{length\ of\ A} 2^{x_i} 3^{y_i}$ and $B = \sum_{i=1}^{length\ of\ B} 2^{x_i} 3^{y_i}$

(Represented by Array A2, Array A3, Array B2, Array B3)

Array A2: Array of exponents with base 2 of value A

Array A3: Array of exponents with base 3 of value A

Array B2: Array of exponents with base 2 of value B

Array B3: Array of exponents with base 3 of value B

output: r where $r = A \times B$

begin

$r \leftarrow 0$

for (i=0 to length of A) **do**

for (j=0 to length of B) **do**

$$r \longleftarrow r+2^{ArrayA2[i]+ArrayB2[j]} \times 3^{ArrayA3[i]+ArrayB3[j]}$$

end

โดยค่าความซับซ้อนเชิงเวลาของอัลกอริทึมที่ 3.3.5 คือ $\theta(l_A \times l_B)$ เมื่อ l_A คือจำนวนพจน์ของจำนวนเต็มบวก A และ l_B คือจำนวนพจน์ของจำนวนเต็มบวก B หากเปรียบเทียบค่าความซับซ้อนเชิงเวลาของอัลกอริทึมที่ 3.3.4 กับอัลกอริทึมที่ 3.3.5 จะเห็นได้ว่าการประมาณด้วยพจน์เดียวนั้นทำให้ค่าความซับซ้อนเชิงเวลาอยู่ในรูปค่าคงที่และยังคงสามารถใช้คุณสมบัติของเอกซ์โพเนนเชียลได้อยู่ แต่ในขณะเดียวกันผลคูณที่เกิดจากการประมาณด้วยพจน์เดียวจะมีอีก 3 พจน์ที่ไม่ได้อยู่ในรูปแบบแทนจำนวนฐานคู่ซึ่งจะทำให้ไม่สามารถใช้คุณสมบัติของเอกซ์โพเนนเชียลได้ในทันที แต่ถ้าหากเราสามารถแปลงค่าพจน์ที่ไม่ได้อยู่ในรูปแบบแทนจำนวนฐานคู่ให้อยู่ในรูปแบบแทนจำนวนฐานคู่ได้จะทำให้อัลกอริทึมในการคูณโดยประมาณค่าสามารถใช้คุณสมบัติของเอกซ์โพเน็นเชียลได้อย่างสมบูรณ์ภายใต้ค่าความซับซ้อนเชิงเวลาเท่ากับ $\theta(1)$ หรือดีกว่า $\theta(l_A \times l_B)$

3.3.3 การนำค่าประมาณจำนวนเต็มบวกด้วยรูปแบบแทนจำนวนฐานคู่เพียงพจน์เดียวมานำเสนออัลกอริทึมการหาเศษเหลือในระบบจำนวนฐานคู่

สำหรับการหาเศษเหลือ ลองพิจารณาตัวอย่างดังนี้

ตัวอย่างที่ 3.3.2 กำหนดให้จำนวนเต็มบวก A มีค่าเท่ากับ 189,735,964

ซึ่งแทนได้ด้วย $2^{10}3^{11} + 2^{15}3^{5} + 2^{9}3^{6} + 2^{9}3^{1} + 2^{0}3^{3} + 2^{0}3^{0}$

และกำหนดให้จำนวนเต็มบวก B มีค่าเท่ากับ 31 ซึ่งแทนได้ด้วย $2^{0}3^{3} + 2^{2}3^{0}$

เมื่อเรานำค่า B ไปประมาณค่าด้วยอัลกอริทึมที่ 3.2 และกำหนดขอบเขตค่าความคลาดเคลื่อนเท่ากับ 0.5 จะได้ว่าค่าประมาณในรูปแบบจำนวนเพียงพจน์เดียวของ B คือ $2^{43}3^{-24}$ ซึ่งเท่ากับ 31.14438076061395 และค่าคลาดเคลื่อนของการประมาณค่าคือ ε ซึ่งมีค่าเท่ากับ

-0.14438076061395 โดยเราต้องการหาค่า A mod B

เริ่มพิจารณาจากนิยามของเศษเหลือ

นิยามของเศษเหลือ สำหรับ $a,b \in \mathbb{Z}^{+}$ และ r คือเศษเหลือ เมื่อ r เท่ากับ $a \bmod b$ สามารถเขียนให้อยู่ในรูปสมการได้ว่า $a = bx + r$ เมื่อ $x \in \mathbb{Z}_{0}^{+}$ และ $0 \leq r < |b|$

ดังนั้น $A = B\left\lfloor \frac{A}{\omega} \right\rfloor + r$ โดยที่ $\left\lfloor \frac{A}{\omega} \right\rfloor \in \mathbb{Z}_{0}^{+}$ เมื่อ ω คือค่าประมาณของ B และค่า $\frac{A}{\omega}$ สามารถคำนวณได้จากอัลกอริทึมที่ 3.3.3

โดยเราสามารถแปลงค่า $\left\lfloor \frac{A}{\omega} \right\rfloor$ ให้อยู่ในเทอม $\sum_{i=1}^{\mathbb{N}} 2^{x_i} 3^{y_i}$ เมื่อ $x,y \in \mathbb{Z}$ เพื่อให้ง่ายต่อคำนวณ

$$A = B \times \sum_{i=1}^{\mathbb{N}} 2^{x_i} 3^{y_i} + r$$

$$A = (2^0 3^3 + 2^2 3^0) \times \sum_{i=1}^{\mathbb{N}} 2^{x_i} 3^{y_i} + r$$

$$\therefore\ r = A - (2^0 3^3 + 2^2 3^0) \times \sum_{i=1}^{\mathbb{N}} 2^{x_i} 3^{y_i}$$ $\square$

จากตัวอย่างที่ 3.3.2 สามารถนำเสนออัลกอริทึมในการหาเศษเหลือได้ดังนี้

อัลกอริทึมที่ 3.3.6 อัลกอริทึมหาเศษเหลือโดยการประมาณค่า

Algorithm3.2.6(Array A2, Array A3, Array B2, Array B3)

input: $A = \sum_{i=1}^{length\,of\,A} 2^{x_i} 3^{y_i}$ and $B = \sum_{i=1}^{length\,of\,B} 2^{x_i} 3^{y_i}$ where $x,y \in \mathbb{Z}$

(Represented by ArrayA2, ArrayA3, ArrayB2, ArrayB3)

Array A2: Array of exponents with base 2 of value A

Array A3: Array of exponents with base 3 of value A

Array B2: Array of exponents with base 2 of value B

Array B3: Array of exponents with base 3 of value B

output: r where r = A mod B

begin

Approximate value B

$p,q \leftarrow$ Algorithm3.2 (B , $\alpha = 0.5$)

error $\leftarrow \left| B - 2^p 3^q \right|$

Divided by New Proposed Algorithm

$d \leftarrow$ Algorithm3.2.3(A2, A3, p , q , error)

Convert Integer d to DBNS

$v \leftarrow$ Algorithm2.1($\lfloor d \rfloor$)

$r \leftarrow A - B \times v$

end

3.4 ลดเวลาการทำงานในการหาค่าประมาณจากอัลกอริทึมการประมาณค่าโดยการสร้างตารางเก็บค่าไว้ล่วงหน้า

จากหัวข้อที่ 3.3 จะเห็นได้ว่าการที่เราสามารถประมาณจำนวนเต็มบวกด้วย $2^b 3^t$ เพียงพจน์เดียวเมื่อ $b,t \in \mathbb{Z}$ นั้นมีข้อดีในเรื่องของการดำเนินการทางคณิตศาสตร์ในระบบจำนวนฐานคู่ แต่อีกสิ่งหนึ่งที่ควรคำนึงถึงคือด้านเวลาของอัลกอริทึมที่ใช้ในการประมาณค่าซึ่งถ้าวิเคราะห์เชิงเวลาจะสามารถเขียนให้อยู่ในรูปอสมการได้นี้

$$T_{Approx} + T_{NewOperate} < T_{Operate}$$

T_{Approx} : เวลาที่ใช้ในการประมาณค่าจำนวนเต็มบวกด้วยเมื่อ $2^b 3^t$ เมื่อ $b,t \in \mathbb{Z}$

$T_{NewOperate}$: เวลาที่ใช้ในการคำนวนการดำเนินการทางคณิตศาสตร์ด้วยอัลกอริทึมที่เราได้เสนอ

$T_{Operate}$: เวลาที่ใช้ในการคำนวนการดำเนินการทางคณิตศาสตร์ด้วยอัลกอริทึมแบบธรรมดา

นั่นหมายความเราต้องหาวิธีลดเวลาในการดำเนินการของอัลกอริทึมประมาณค่า โดยแนวทางที่เรานำเสนอคือการสร้างตารางเก็บค่าไว้ล่วงหน้าเพื่อลดการดำเนินการของอัลกอริทึมประมาณค่าจำนวนเต็มบวก

พิจารณาการสร้างตารางเก็บค่าจากกระบวนการประมาณค่าและปรับปรุงค่าที่ได้นำเสนอ จะวิเคราะห์ได้ 2 กรณี

1. กรณีที่ไม่ได้สร้างค่าไว้ล่วงหน้า

จำนวนค่าเริ่มต้นที่ต้องใช้ใน 2 ส่วนหลัก ๆ ประกอบไปด้วย

- ค่าเริ่มต้น 10 ค่าสำหรับอัลกอริทึมการประมาณค่าในช่วง $(1,2)$ ในรูปแทนจำนวนฐานคู่เพียงพจน์เดียว (สำหรับอัลกอริทึมที่ 3.1) ดังตารางที่ 1

ตารางที่ 1 ค่าเริ่มต้นสำหรับอัลกอริทึมที่ 3.1 เพื่อคำนวณหาค่าประมาณที่แม่นยำที่สุด

Index	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
Item	$2^{8}3^{-5}$	$2^{-3}3^{2}$	$2^{5}3^{-3}$	$2^{-6}3^{4}$	$2^{2}3^{-1}$	$2^{-1}3$	$2^{7}3^{-4}$	$2^{-4}3^{3}$	$2^{4}3^{-2}$	$2^{-7}3^{5}$

- ค่าเริ่มต้น 5 ค่าสำหรับอัลกอริทึมในการประมาณจำนวนเต็มบวก $\mathbb{N}$ ในรูปพจน์ $2^{w}3^{z}$ เมื่อ $w,z \in \mathbb{Z}$ (สำหรับอัลกอริทึมที่ 3.2) ดังตารางที่ 2

ตารางที่ 2 ค่าเริ่มต้นสำหรับอัลกอริทึมที่ 3.2 เพื่อสร้างค่าในช่วง $(1,2^{8}3^{-5})$ ในรูปพจน์ $2^{w}3^{z}$ เมื่อ $w,z \in \mathbb{Z}$

Index	A[0]	A[1]	A[2]	A[3]	A[4]
Item	$2^{8}3^{-5}$	$2^{-3}3^{2}$	$2^{5}3^{-3}$	$2^{-6}3^{4}$	$2^{2}3^{-1}$

2. กรณีที่สร้างค่าไว้ล่วงหน้า

จากหัวข้อที่ 3.1 ในกระบวนการประมาณค่าจำนวนเต็มบวกจะเห็นได้ว่าคำตอบของการประมาณค่านั้นจะได้มาจากผลลัพธ์ของอัลกอริทึมที่ 3.1 คูณกับค่าที่เกิดจากตารางของอัลกอริทึมที่ 3.2

ดังนั้นเราจึงสามารถแบ่งวิธีการสร้างตารางเก็บค่าได้เป็น 2 กรณีย่อยได้ดังนี้

2.1 สร้างตาราง 2 ตารางสำหรับ 2 อัลกอริทึม (อัลกอริทึมที่ 3.1 และอัลกอริทึมที่ 3.2)

โดยอัลกอริทึมที่ 3.1 จะต้องเก็บค่าเริ่มต้นไว้ทั้งหมด k ค่า เมื่อ k เป็นจำนวนเต็มบวก เนื่องจากในบทวิเคราะห์ในงานวิจัย [16] ได้เสนอว่าเราสามารถหาทุกค่าในอัลกอริทึมที่ 3.1 ได้ด้วยจำนวนที่จำกัด นั่นคือเราสามารถสร้างตารางเก็บค่าทั้งหมด k ค่า ซึ่งค่า k มีค่าประมาณ 80

ซึ่งอัลกอริทึมที่ 3.2 จะต้องเก็บค่าเริ่มต้นไว้ทั้งหมด m ค่า โดยที่ $5|m$ เมื่อ $m\in\mathbb{N}$ เนื่องจากทุกครั้งที่เราทำการเพิ่มค่าที่เกิดจากการปรับค่าในตารางที่ 3.2 เราจะทำการเพิ่มครั้งละ 5 ค่าสำหรับการเรียกใช้อัลกอริทึมปรับค่าในหนึ่งครั้ง

ดังนั้นจะได้ว่าค่าความซับซ้อนเชิงพื้นที่สำหรับกรณีนี้คือ $\theta(k+m)$

2.2 สร้างตารางเพียงตารางเดียวสำหรับเพื่อหาค่าในช่วง (1,2) ซึ่งพร้อมสำหรับการค้นหาคำตอบได้ทันที

จะได้ว่าตารางเก็บค่าต้องมีขนาดเท่ากับขนาดของตารางเก็บค่าของอัลกอริทึมที่ 3.1 คูณกับขนาดของตารางเก็บค่าของอัลกอริทึมที่ 3.2

ดังนั้นจะได้ว่าค่าความซับซ้อนเชิงพื้นที่สำหรับกรณีนี้คือ $\theta(km)$

เมื่อเราทำการเปรียบเทียบค่าความซับซ้อนเชิงพื้นที่ของกรณี 2.1 และกรณี 2.2 จะได้เป็น

$\frac{k+m}{km}$ ซึ่ง $\frac{k+m}{km}=\frac{k}{km}+\frac{m}{km}=\frac{1}{m}+\frac{1}{k}<0.2125$ เมื่อ $m\geq 5$ และ $k\approx 80$

นั่นแสดงว่าสำหรับวิธีการตารางเก็บค่าไว้ล่วงหน้าในกรณี 2.1 นั้นใช้พื้นที่เพียงไม่ถึง 21.25 เปอร์เซ็นต์ของกรณี 2.2 จึงเป็นแนวทางที่ค่อนข้างหน้าสนใจสำหรับการประมวลผลไว้ล่วงหน้าด้วยกรณี 2.1

จากการพิจารณาวิธีสร้างตารางเก็บค่าที่ได้กล่าวไว้ก่อนหน้า เราจะเลือกวิธีการสร้างตาราง 2 ตารางสำหรับ 2 อัลกอริทึม (อัลกอริทึมที่ 3.1 และอัลกอริทึมที่ 3.2) โดยตารางเก็บค่าสำหรับอัลกอริทึมที่ 3.1 นั้นจะมีขนาดเป็นค่าคงที่ แต่สำหรับตารางเก็บค่าสำหรับอัลกอริทึมที่ 3.2 นั้นอาจจะไม่สามารถสร้างได้ด้วยขนาดที่เป็นค่าคงที่ เนื่องจากสำหรับอัลกอริทึมที่ 3.2 นั้นอาศัยการเปลี่ยนแปลงค่าในตารางเริ่มต้นจนกระทั่งได้ค่าที่สอดคล้องกับเงื่อนไขที่กำหนดจึงหยุดการปรับค่า

โดยในหัวข้อนี้เราได้วิเคราะห์กระบวนการปรับค่าในตาราง ของอัลกอริทึมที่ 3.2 และได้นำเสนออัลกอริทึมปรับค่าในตารางที่มีประสิทธิภาพด้านเวลาดีกว่าอัลกอริทึมปรับค่าในงานวิจัย [16] รวมถึงนำอัลกอริทึมในการสร้างตารางเก็บค่าสำหรับการดำเนินการของอัลกอริทึมที่ 3.2

โดยหลักการของอัลกอริทึมปรับค่าคือการหาค่าในช่วง $(1, 2^8 3^{-5})$ ซึ่งเราสามารถหาคำตอบได้จากการเปลี่ยนแปลงค่าในตารางที่ 3.2 ด้วยฟังก์ชันที่ได้เสนอในงานวิจัย [16] ดังด้านล่างนี้

$F(p)$ คือฟังก์ชันในการปรับค่า p เมื่อ p คือค่าในตารางที่ 3.2

$$F(p) = p \times 2^{8n-1} 3^{-5n} \quad (4)$$

เมื่อค่า n คือค่าจำนวนเต็มบวกที่น้อยที่สุดซึ่งสอดคล้องกับเงื่อนไข $p \times (2^8 3^{-5})^n > 2$

โดยหลักการเปลี่ยนแปลงค่านี้สามารถแปลงเป็นอัลกอริทึมได้ดังนี้

อัลกอริทึมที่ 3.4.1 อัลกอริทึมปรับค่าเพื่อหาค่าในช่วง $(1, 2^8 3^{-5})$

input: $p \in (1, 2)$

output: $q \in (1, 2^8 3^{-5})$

begin

while ($p < 2$) **do**

$p \leftarrow p \times 2^8 3^{-5}$

enddo

$q \leftarrow \frac{p}{2}$

end

จากหลักการเปลี่ยนแปลงค่าที่เห็นได้อย่างชัดเจนคือการคูณค่า p ด้วยค่า $2^8 3^{-5}$ ไปจนกระทั่งค่า p มีค่ามากกว่า 2 แล้วหารด้วย 2 ซึ่งสามารถพิจารณาปัญหาในรูปอสมการได้ดังนี้

$$p \times (2^8 3^{-5})^n > 2$$

$$(2^8 3^{-5})^n > \frac{2}{p}$$

$$n > \log_{(2^8 3^{-5})} \frac{2}{p}$$

$$n = \left\lceil \log_{(2^8 3^{-5})} \frac{2}{p} \right\rceil$$

$$p = p \times 2^{8n-1} 3^{-5n}$$

เมื่อนำวิธีการเปลี่ยนแปลงค่าดังอสมการข้างต้นมานำเสนอเป็นอัลกอริทึมจะได้ดังนี้

อัลกอริทึมที่ 3.4.2 อัลกอริทึมเพื่อปรับค่าเพื่อหาค่าในช่วง $(1, 2^8 3^{-5})$ (ปรับแต่งจากอัลกอริทึม 3.4.1)

input: $p \in (1, 2)$

output: $q \in (1, 2^8 3^{-5})$

begin

$n \leftarrow \left\lceil \log_{(2^8 3^{-5})} \frac{2}{p} \right\rceil$

$q \leftarrow p \times 2^{8n-1} 3^{-5n}$

end

โดยค่าความซับซ้อนเชิงเวลาของอัลกอริทึมปรับค่าที่เราได้เสนอ (อัลกอริทึมที่ 3.3.2) จะเท่ากับ $\theta(1)$ เมื่อเปรียบเทียบกับอัลกอริทึมปรับค่าในงานวิจัย [16] (อัลกอริทึมที่ 3.4.1) ซึ่งเท่ากับ $\theta(\left\lceil \log_{(2^8 3^{-5})} \frac{2}{p} \right\rceil)$

โดยเราได้เสนอหลักการสร้างตารางค่า ซึ่งอาศัยหลักการของอัลกอริทึมเชิงละโมบ เนื่องจากว่าเราไม่สามารถทราบได้ว่าในตารางควรมีค่าใดบ้าง เราจึงทำการเพิ่มค่าเข้าไปทีละ 5 ค่าแบบเรียงลำดับค่า จากการเรียกใช้หนึ่งครั้งของอัลกอริทึมสำหรับสร้างค่าซึ่งแปลงเป็นอัลกอริทึมจะได้ดังนี้

อัลกอริทึมที่ 3.4.3 อัลกอริทึมในการสร้างตารางค่าสำหรับอัลกอริทึมที่ 3.2

input: Table ; For collect value

Array A ; Value for Update

output : None

begin

for (i = 0 to 4) **do**

Update Value in Initial Table

$q \leftarrow$ Algorithm3.4.2(A[i])

Insert Value q to Table by Sequence

end

โดยในทางปฏิบัติแล้วเราสามารถเตรียมตารางค่าสำหรับอัลกอริทึมที่ 3.2 ได้ด้วยการกำหนดค่าจำนวนรอบเพื่อเรียกใช้อัลกอริทึมที่ 3.4.3 ซึ่งในครั้งแรกของการเรียกใช้อัลกอริทึมที่ 3.4.3 ต้องใช้ค่าในตารางที่ 3.2 เพื่อเป็นตัวแปรขาเข้าสำหรับการสร้างค่าในตารางและเมื่อมีการเรียกใช้อัลกอริทึมที่ 3.4.3 ในครั้งถัดไปเราจะใช้ค่าในตารางที่ 3.2 ที่ได้ปรับค่าล่าสุดมาเป็นตัวแปรขาเข้า

บทที่ 4

บทวิเคราะห์การประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่

ในบทนี้จะกล่าวถึงบทวิเคราะห์ในด้านต่าง ๆ ของการประมาณค่าจำนวนเต็มบวกใด ๆ ด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่ โดยจะเน้นที่การวิเคราะห์เวลาในการดำเนินการและค่าความซับซ้อนเชิงเวลาของอัลกอริทึมพร้อมทั้งเปรียบเทียบ

4.1 เวลาในการดำเนินการของอัลกอริทึมประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยว

ทางผู้วิจัยได้ออกแบบการทดลองเพื่อวิเคราะห์เวลาการดำเนินการของอัลกอริทึมประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยว

โดยผู้วิจัยได้แบ่งการทดลองออกเป็น 4 ส่วนดังนี้

ส่วนที่ 1 ทำการศึกษาเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง

(1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5

ส่วนที่ 2 ทำการศึกษาเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง

(1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3

ส่วนที่ 3 ทำการศึกษาเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง

(1,000,000-100,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 1

ส่วนที่ 4 ทำการเปรียบเทียบเวลาที่ในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง

(1,000,000-100,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนคือ 0.3 กับ 0.5

ส่วนที่ 1 ทำการศึกษาเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง

(1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5

โดยการทดลองมีขั้นตอนดังนี้

1) กำหนดให้ค่า n เท่ากับ 1,000,000 และบวกด้วยจำนวนเต็มบวกที่เกิดจากการสุ่ม

ในช่วง (1,000,000-2,000,000)

2) ประมาณค่าจำนวนเต็มบวก n ด้วยพจน์เดี่ยวโดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5

3) บวกค่า n ด้วยจำนวนเต็มบวกที่เกิดจากการสุ่มในช่วง (1,000,000-2,000,000)

4) พิจารณาค่า n ว่ามีค่าเกิน 50,000,000 หรือไม่

หากไม่เป็นจริง ให้กลับไปทำซ้ำขั้นตอนที่ 2

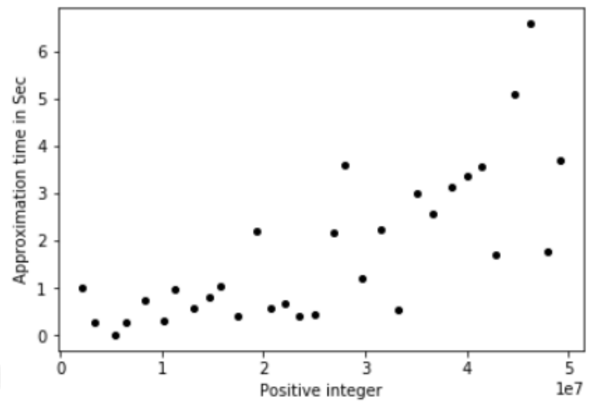
หากเป็นจริง ให้ทำต่อในขั้นตอนที่ 5

5) นำทุกคู่อันดับ (x,y) มาแสดงด้วยกราฟ

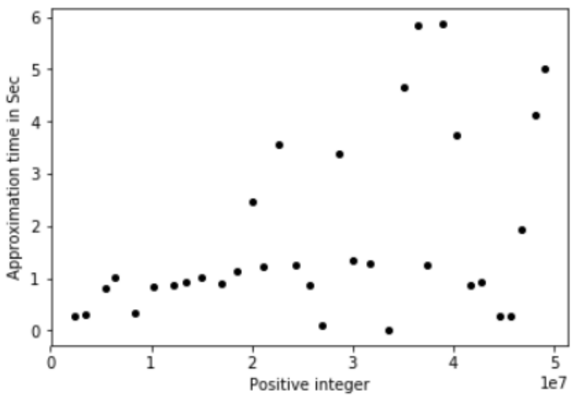
โดยแสดงเป็นพิกัดดังนี้ x : จำนวนเต็มบวก n

y : เวลาในการดำเนินการของอัลกอริทึมประมาณค่า หน่วยคือวินาที

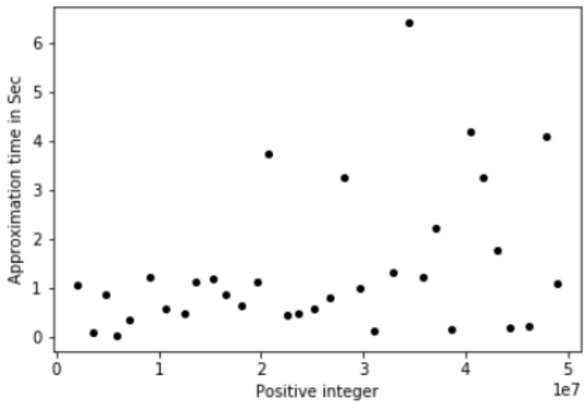
โดยผลลัพธ์จากการทดลองซ้ำ 3 ครั้งสามารถดูได้ดังรูปที่ 6-8



รูปที่ 6 ผลการทดลองครั้งที่ 1 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5



รูปที่ 7 ผลการทดลองครั้งที่ 2 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5



รูปที่ 8 ผลการทดลองครั้งที่ 3 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5

ส่วนที่ 2 ทำการศึกษาเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3

โดยการทดลองมีขั้นตอนดังนี้

1) กำหนดให้ค่า n เท่ากับ 1,000,000 และบวกด้วยจำนวนเต็มบวกที่เกิดจากการสุ่ม ในช่วง (1,000,000-2,000,000)

2) ประมาณค่าจำนวนเต็มบวก n ด้วยพจน์เดี่ยวโดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3

3) บวกค่า n ด้วยจำนวนเต็มบวกที่เกิดจากการสุ่มในช่วง (1,000,000-2,000,000)

4) พิจารณาค่า n ว่ามีค่าเกิน 50,000,000 หรือไม่

หากไม่เป็นจริง ให้กลับไปทำซ้ำขั้นตอนที่ 2

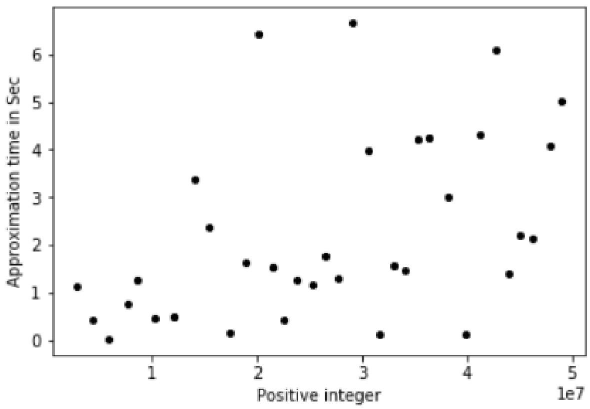
หากเป็นจริง ให้ทำต่อในขั้นตอนที่ 5

5) นำทุกคู่อันดับ (x,y) มาแสดงด้วยกราฟ

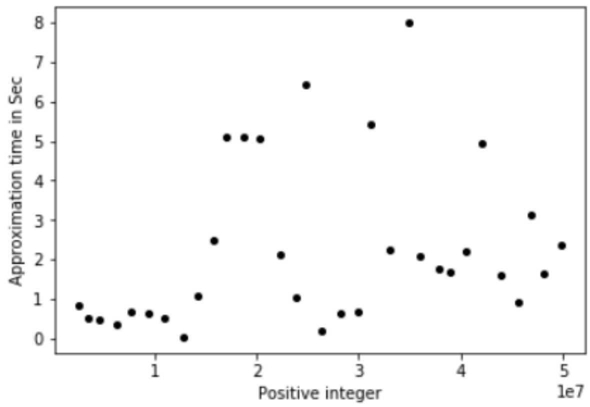
โดยแสดงเป็นพิกัดดังนี้ x : จำนวนเต็มบวก n

y : เวลาในการดำเนินการของอัลกอริทึมประมาณค่า หน่วยคือวินาที

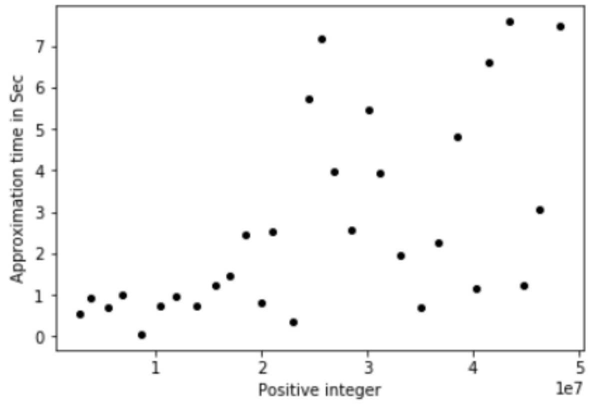
โดยผลลัพธ์จากการทดลองซ้ำ 3 ครั้งสามารถดูได้ดังรูปที่ 9-11



รูปที่ 9 ผลการทดลองครั้งที่ 1 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3



รูปที่ 10 ผลการทดลองครั้งที่ 2 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3



รูปที่ 11 ผลการทดลองครั้งที่ 3 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3

ส่วนที่ 3 ทำการศึกษาเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-100,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 1

โดยการทดลองมีขั้นตอนดังนี้

1) กำหนดให้ค่า n เท่ากับ 1,000,000 และบวกด้วยจำนวนเต็มบวกที่เกิดจากการสุ่ม ในช่วง (1,000,000-2,000,000)

2) ประมาณค่าจำนวนเต็มบวก n ด้วยพจน์เดี่ยวโดยกำหนดค่าขอบเขตความผิดพลาดเท่ากับ 1

3) บวกค่า n ด้วยจำนวนเต็มบวกที่เกิดจากการสุ่มในช่วง (1,000,000-2,000,000)

4) พิจารณาค่า n ว่ามีค่าเกิน 100,000,000 หรือไม่

หากไม่เป็นจริง ให้กลับไปทำซ้ำขั้นตอนที่ 2

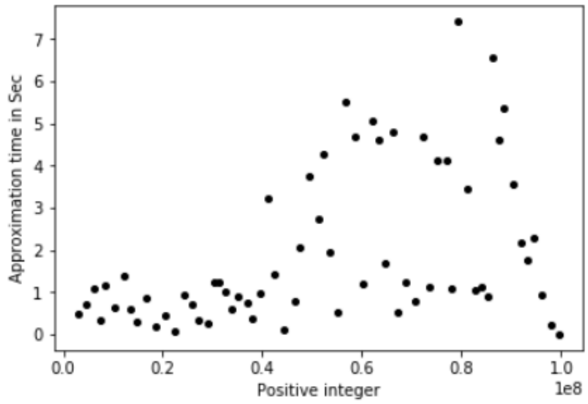
หากเป็นจริง ให้ทำต่อในขั้นตอนที่ 5

5) นำทุกคู่อันดับ (x,y) มาแสดงด้วยกราฟ

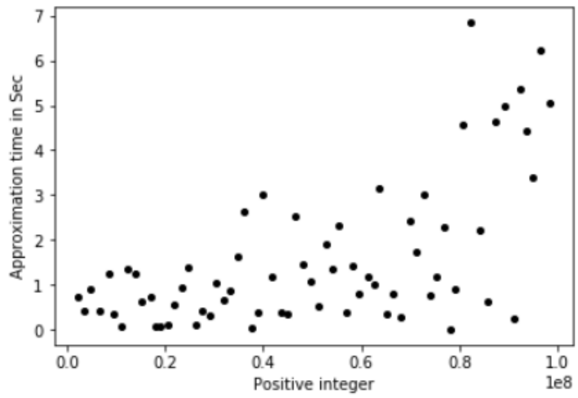
โดยแสดงเป็นพิกัดดังนี้ x : จำนวนเต็มบวก n

y : เวลาในการดำเนินการของอัลกอริทึมประมาณค่า หน่วยคือวินาที

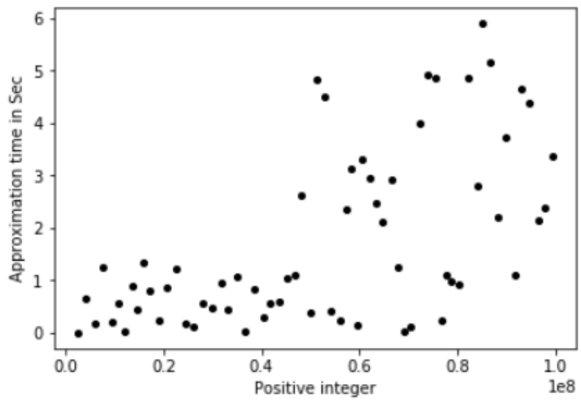
โดยผลลัพธ์จากการทดลองซ้ำ 3 ครั้งสามารถดูได้ดังรูปที่ 12-14



รูปที่ 12 ผลการทดลองครั้งที่ 1 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-100,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 1



รูปที่ 13 ผลการทดลองครั้งที่ 2 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-100,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 1



รูปที่ 14 ผลการทดลองครั้งที่ 3 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-100,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 1

ส่วนที่ 4 ทำการเปรียบเทียบเวลาที่ในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนคือ 0.3 กับ 0.5

โดยการทดลองมีขั้นตอนดังนี้

1) กำหนดให้ค่า n เท่ากับ 1,000,000 และบวกด้วยจำนวนเต็มบวกที่เกิดจากการสุ่ม ในช่วง (1,000,000-2,000,000)
2) ประมาณค่าจำนวนเต็มบวก n ด้วยพจน์เดี่ยวโดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3 และประมาณค่าจำนวนเต็มบวก n ด้วยพจน์เดี่ยวโดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5
3) บวกค่า n ด้วยจำนวนเต็มบวกที่เกิดจากการสุ่มในช่วง (1,000,000-2,000,000)
4) พิจารณาค่า n ว่ามีค่าเกิน 50,000,000 หรือไม่

   หากไม่เป็นจริง ให้กลับไปทำซ้ำขั้นตอนที่ 2

   หากเป็นจริง ให้ทำต่อในขั้นตอนที่ 5
5) นำทุกคู่อันดับ (x,y) และ (x,z) มาแสดงด้วยกราฟ

   โดยแสดงเป็นพิกัดดังนี้ x : จำนวนเต็มบวก n

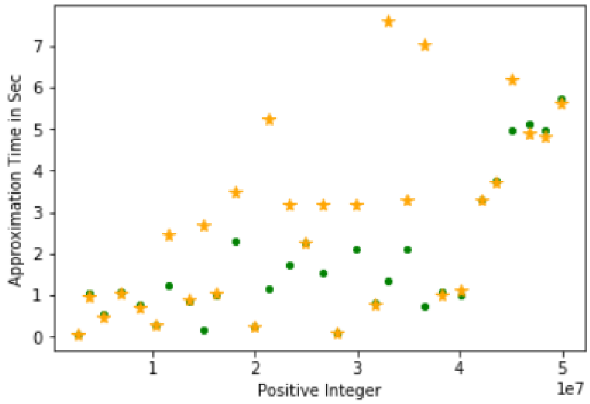
   y : เวลาในการดำเนินการของอัลกอริทึมประมาณค่าด้วยค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3 หน่วยเวลาคือวินาที

   z : เวลาในการดำเนินการของอัลกอริทึมประมาณค่าด้วยค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5 หน่วยเวลาคือวินาที

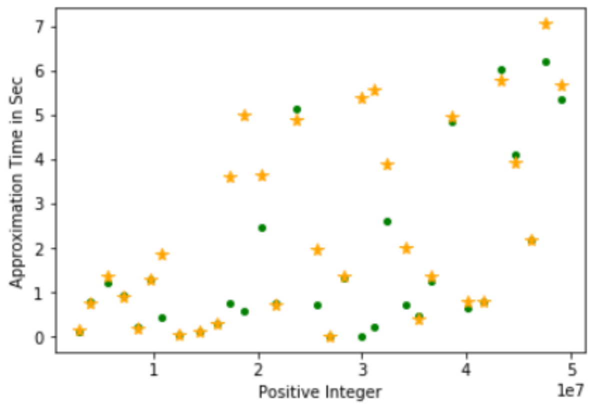
หมายเหตุ กำหนดให้จุดรูปทรงดาวแทนเวลาในการดำเนินการของอัลกอริทึมประมาณค่า ซึ่งกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.3

และกำหนดให้จุดวงกลมแทนเวลาในการดำเนินการของอัลกอริทึมประมาณค่า ซึ่งกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5

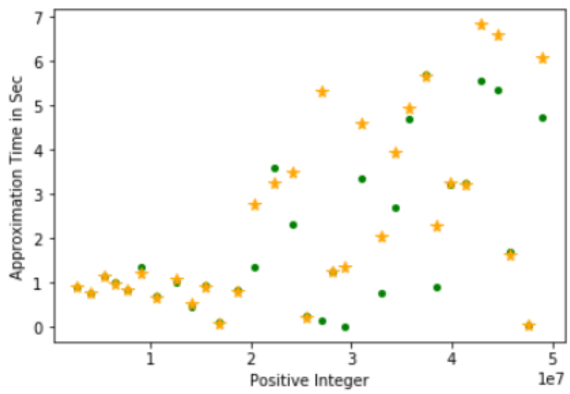
โดยผลลัพธ์จากการทดลองซ้ำ 3 ครั้งสามารถดูได้ดังรูปที่ 15-17



รูปที่ 15 ผลการทดลองครั้งที่ 1 เปรียบเทียบเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนที่ต่างกันคือ 0.3 กับ 0.5



รูปที่ 16 ผลการทดลองครั้งที่ 2 เปรียบเทียบเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนที่ต่างกันคือ 0.3 กับ 0.5



รูปที่ 17 ผลการทดลองครั้งที่ 3 เปรียบเทียบเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (1,000,000-50,000,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนที่ต่างกันคือ 0.3 กับ 0.5

จากการทดลองเพื่อสังเกตุเวลาในการดำเนินการของอัลกอริทึมประมาณค่านั้น ผู้วิจัยพบว่าเวลาในการประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่นั้นอยู่ในรูปเอกซ์โพเนนเชียล โดยเวลาการดำเนินการของอัลกอริทึมประมาณค่านั้นจะแปรผันตรงกับจำนวนเต็มบวกที่เป็นตัวแปรขาเข้า หรือเวลาการดำเนินการของอัลกอริทึมประมาณค่านั้นจะแปรผกผันกับค่าขอบเขตความคลาดเคลื่อนที่เป็นตัวแปรขาเข้า

4.2 ค่าความซับซ้อนเชิงเวลาของอัลกอริทึมการดำเนินการทางคณิตศาสตร์

สำหรับอัลกอริทึมประมาณค่านั้นเป็นอัลกอริทึมที่ทำงานโดยวิธีค้นเฉพาะที่ (local search) ซึ่งจะหยุดการทำงานก็เมื่อเจอค่าประมาณที่ให้ความคลาดเคลื่อนไม่เกินค่าขอบเขตความผิดพลาดที่กำหนดให้ ซึ่งสามารถเขียนให้อยู่ในรูปค่าความซับซ้อนเชิงเวลาได้เป็น $\mathrm{O}(\alpha^{-1})$

สำหรับค่าความซับซ้อนเชิงเวลาของอัลกอริทึมการหารที่ผู้วิจัยได้เสนอจะอยู่ในรูป $\theta(l_A)$ เมื่อ l_A คือจำนวนพจน์ในระบบจำนวนฐานคู่ของจำนวนเต็มบวก A ซึ่งอาศัยคุณสมบัติของเอกซ์โพเน็นเชียลในการดำเนินการหาร ซึ่งน้อยที่สุดเมื่อเปรียบเทียบกับอัลกอริทึมการหารที่ไม่ได้อาศัยการประมาณค่าตัวส่วนทั้ง 2 อัลกอริทึม ได้แก่ อัลกอริทึมการหารแบบอาศัยคุณสมบัติของเอกซ์โพเน็นเชียล ค่าความซับซ้อนเชิงเวลาจะอยู่ในรูป $\theta(l_A \times l_B)$ และอัลกอริทึมการหารโดยตรง ค่าความซับซ้อนเชิงเวลาจะอยู่ในรูป $\theta(l_A + l_B)$ แต่ทั้งนี้กระบวนการหารโดยใช้อัลกอริทึมที่ผู้วิจัยเสนอนั้นอาจจะช้าในทางปฏิบัติ เนื่องด้วยการคำนวณเลขชี้กำลังของพจน์ส่วนที่เป็นจำนวนเต็มซึ่งมีค่ามาก

สำหรับอัลกอริทึมการคูณที่อาศัยการประมาณค่ามาประยุกต์โดยการประมาณค่าจำนวนเต็มที่ดำเนินคูณทั้งสองจำนวนให้อยู่ในรูปพจน์เดี่ยวและใช้ค่าคลาดเคลื่อนจากการประมาณมาเป็นตัวแปรขาเข้า ดังที่เสนอไว้ในบทที่ 3 ค่าความซับซ้อนเชิงเวลาจะอยู่ในรูปค่าคงคงที่หรือ $\theta(1)$ เมื่อเปรียบเทียบกับการอัลกอริทึมการคูณที่ไม่ได้อาศัยการประมาณค่าคือ $\theta(l_A \times l_B)$

4.3 การสร้างตารางเก็บค่าไว้ล่วงหน้าเพื่อลดเวลาการดำเนินการของอัลกอริทึมประมาณค่า

ผู้วิจัยได้ทำการออกแบบการทดลองเพื่อเปรียบเทียบความแตกต่างด้านเวลาในการดำเนินการหาค่าประมาณจำนวนเต็มบวกด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่ ระหว่างกรณีที่ไม่ได้เตรียมตารางเรียกดูค่าไว้ล่วงหน้ากับกรณีที่มีการเตรียมตารางเรียกดูค่าซึ่งมีคำตอบของการประมาณค่าอยู่ในตารางนั้นแล้ว

โดยขั้นตอนการทดลองมีดังนี้

1) กำหนดตารางค้นหาค่าเริ่มต้นที่เป็นอาเรย์ว่างขึ้นมา

2) กำหนดให้ค่า n เท่ากับ 10,000 และบวกด้วยจำนวนเต็มบวกที่เกิดจากการสุ่ม ในช่วง (5,000-10,000)

3) ประมาณค่าจำนวนเต็มบวก n ด้วยพจน์เดี่ยวโดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5 โดยเริ่มจากหาคำตอบในตารางค้นหาค่า

-หากในตารางค้นหาค่ามีคำตอบ ให้ส่งคำตอบของการประมาณค่ากลับไปในรูปแบบของเลขชี้กำลัง

-หากไม่พบคำตอบ ให้ทำการเปลี่ยนแปลงค่าในตารางพร้อมกับใส่ค่าไปในตารางเรียกดูค่าตามลำดับจากน้อยไปมากและทำกระทำแบบนี้ซ้ำ ๆ ไปจนกระทั่งพบคำตอบ

หากในตารางคำตอบค้นหาค่ามีคำตอบอยู่แล้วให้ทำการจับเวลาในการเรียกดูค่า

และหากในตารางคำตอบค้นหาค่ายังไม่มีคำตอบให้ทำการเปลี่ยนแปลงค่าในตารางพร้อมกับใส่ค่าในตารางเรียกดูค่าจนกว่าจะเจอคำตอบ หลังจากนั้นจึงทำการจับเวลาในการเรียกดูค่าในตารางที่มีคำตอบแล้ว

4) บวกค่า n ด้วยจำนวนเต็มบวกที่เกิดจากการสุ่มในช่วง (5,000-10,000)

5) พิจารณาค่า n ว่ามีค่าเกิน 500,000 หรือไม่

หากไม่เป็นจริง ให้กลับไปทำซ้ำขั้นตอนที่ 3

หากเป็นจริง ให้ทำต่อในขั้นตอนที่ 6

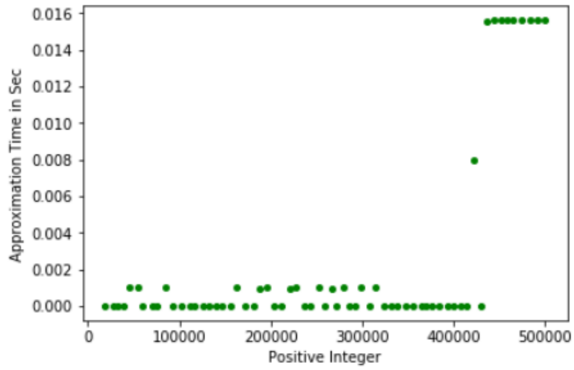
6) นำทุกคู่อันดับ (x,y) และ (x,z) มาแสดงด้วยกราฟ

โดยแสดงเป็นพิกัดดังนี้ x : จำนวนเต็มบวก n

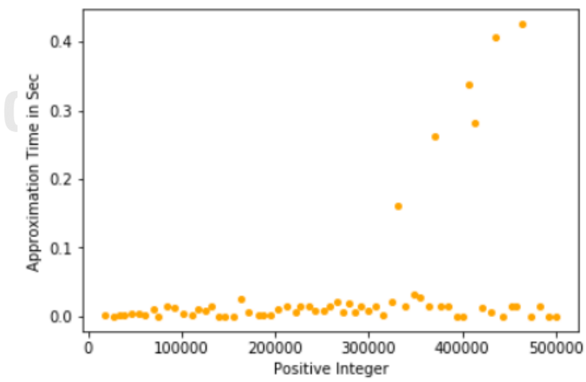
y : เวลาในการค้นหาค่าในตารางค้นหาค่า กรณีที่ในตารางค้นหาค่านั้นมีคำตอบอยู่หน้าแล้ว หน่วยเวลาคือวินาที

z : เวลาในการดำเนินการของอัลกอริทึมประมาณค่า หน่วยเวลาคือวินาที

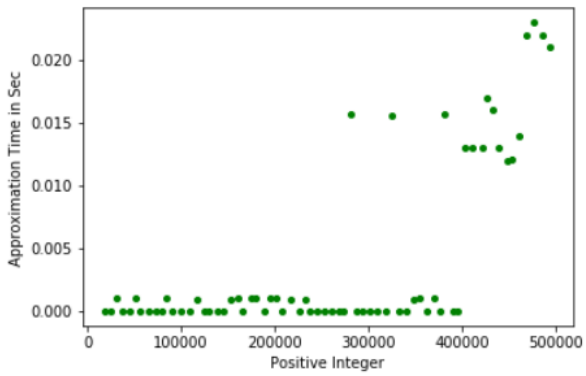
โดยผลลัพธ์จากการทดลองซ้ำ 3 ครั้งสามารถดูได้ดังรูปที่ 18-23



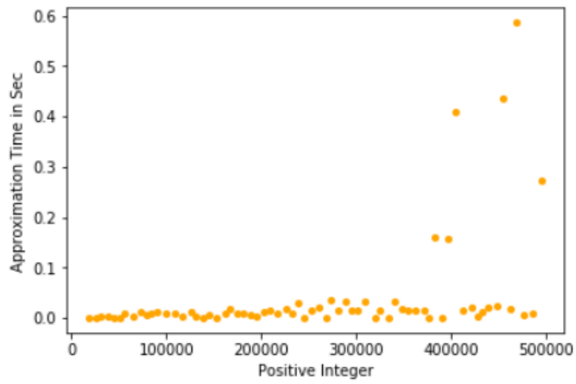
รูปที่ 18 ผลการทดลองครั้งที่ 1 แสดงเวลาในการค้นหาค่าในตารางเก็บค่าของการประมาณค่าในช่วง (10,000-500,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5 ในกรณีที่ตารางนั้นมีคำตอบของการประมาณอยู่แล้ว



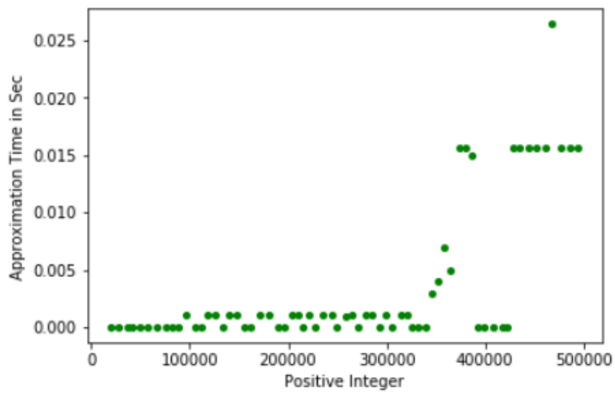
รูปที่ 19 ผลการทดลองครั้งที่ 1 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (10,000-500,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5



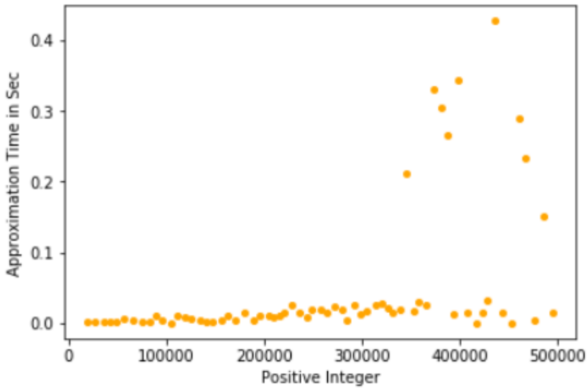
รูปที่ 20 ผลการทดลองครั้งที่ 2 แสดงเวลาในการค้นหาค่าในตารางเก็บค่าของการประมาณค่าในช่วง (10,000-500,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5 ในกรณีที่ตารางนั้นมีคำตอบของการประมาณอยู่แล้ว



รูปที่ 21 ผลการทดลองครั้งที่ 2 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (10,000-500,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5



รูปที่ 22 ผลการทดลองครั้งที่ 3 แสดงเวลาในการค้นหาค่าในตารางเก็บค่าของการประมาณค่าในช่วง (10,000-500,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5 ในกรณีที่ตารางนั้นมีคำตอบของการประมาณอยู่แล้ว



รูปที่ 23 ผลการทดลองครั้งที่ 3 แสดงเวลาในการดำเนินการของอัลกอริทึมประมาณค่าในช่วง (10,000-500,000) โดยกำหนดค่าขอบเขตความคลาดเคลื่อนเท่ากับ 0.5

จากการทดลองจะเห็นว่าถ้ามีการเรียกใช้อัลกอริทึมประมาณค่าหลาย ๆ ครั้ง หากเราเก็บคำตอบที่ได้สร้างไว้จากการเรียกใช้อัลกอริทึมประมาณค่าครั้งล่าสุด เมื่อมีการเรียกใช้อัลกอริทึมประมาณค่าในครั้งไป ในกรณีที่ตารางเก็บค่ามีคำตอบของการประมาณ เราจะสามารถเรียกดูค่านั้นในตารางได้ทันที และในกรณีทีไม่พบคำตอบเราสามารถนำค่าที่ปรับในตารางที่ปรับล่าสุดมาปรับต่อจนกระทั่งเจอคำตอบของการประมาณได้ แต่ทั้งนี้กระบวนการสร้างตารางเก็บค่าไว้ก็มีข้อด้อยคือ ในแต่ละครั้งของการใส่คำตอบลงในตารางเก็บค่านั้นมีความล่าช้าซึ่งเกิดจากการใส่ค่าตามลำดับค่าจากน้อยไปมากและตารางเก็บค่านั้นจะต้องมีขนาดที่ใหญ่เพื่อแลกกับความเร็วในการหาคำตอบ ของการประมาณค่าด้วยพจน์เดียว

บทที่ 5

สรุปผลงานวิจัยและข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

งานวิจัยนี้นำเสนออัลกอริทึมประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยวซึ่งพัฒนามาจากอัลกอริทึมประมาณค่าในระบบจำนวนลอการิทึมมิติคู่แบบขยาย ซึ่งรูปแบบแทนจำนวนในระบบจำนวนลอการิทึมมิติคู่แบบขยายนั้นมีรูปแบบที่คล้ายคลึงกับระบบจำนวนฐานคู่ โดยผลลัพธ์จากการประมาณค่าจำนวนเต็มบวกด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่จากอัลกอริทึมประมาณค่าที่วิจัยนี้ได้เสนอนั้น จะนำไปแก้ไขปัญหาการดำเนินการหารในระบบจำนวนฐานคู่ที่เงื่อนไขของการหารคือตัวหารนั้นอยู่ในรูปแบบแทนจำนวนเพียงพจน์เดียวเท่านั้น ซึ่งเป็นวัตถุประสงค์หลักของงานวิจัยนี้ โดยงานวิจัยนี้ยังได้เสนออัลกอริทึมการหารจากการใช้พจน์ที่เกิดจากการประมาณค่าและวิธีปรับปรุงคำตอบของการหารให้มีความแม่นยำมากกว่าการใช้การหารด้วยพจน์ประมาณเพียงอย่างเดียว

ผลลัพธ์จากงานวิจัยนี้ทำให้สามารถสรุปประเด็นที่น่าสนใจได้ดังนี้

1) ในมุมมองสำหรับการจัดเก็บค่าของจำนวนเต็มบวกที่ถูกประมาณค่าด้วยพจน์เดี่ยว เราสามารถเก็บเลขชี้กำลังของฐานสองและฐานสามในหน่วยความจำได้ทันที เนื่องจากหากเราใช้ค่าขอบเขตความคลาดเคลื่อนเท่า 0.5 เราจะสามารถบอกได้ว่าค่าประมาณที่เราเก็บไว้ในหน่วยความจำนั้นคือจำนวนเต็มบวกใดเมื่อเราทำการคำนวณย้อนกลับมาด้วยวิธีการปัดเศษขึ้นหรือปัดเศษลง

2) สำหรับการประมาณค่าจำนวนเต็มบวกใด ๆ เพื่อใช้เป็นตัวหารนั้นเราไม่จำเป็นต้องใช้ค่าขอบเขตความเคลื่อนที่น้อยจนลู่เข้าใกล้ศูนย์เสมอไป เนื่องด้วยเหตุผลสองประการดังนี้

เหตุผลข้อที่ 1 หากเรากำหนดค่าขอบเขตความคลาดเคลื่อนที่น้อยจนเกินไป อัลกอริทึมการประมาณค่านั้นย่อมใช้เวลาในการดำเนินการที่มากขึ้น

เหตุผลข้อที่ 2 หากพิจารณาเทอมส่วนที่เพิ่มเข้าไปเพื่อทำให้คำตอบของการหารแม่นยำมากขึ้นในอัลกอริทึมการหารที่วิจัยนี้เสนอ จะพบว่าค่าของเทอมส่วนนั้นจะมีค่าที่ยิ่งน้อยลง ซึ่งจะส่งผลให้คำตอบของการหารนั้นคลาดเคลื่อนไป

และเรายังพบว่าเราไม่สามารถคำนวณได้ว่าอัลกอริทึมประมาณค่านั้นจะหยุดเมื่อไหร่จากตัวแปรขาเข้า ซึ่งประกอบไปด้วยจำนวนเต็มบวกและค่าขอบเขตความคลาดเคลื่อน ซึ่งอัลกอริทึมจะหยุดก็ต่อเมื่อเจอคำตอบที่สอดคล้องกับเงื่อนไขที่เรายอมรับนั่นคือเมื่อคำตอบนั้นให้ค่าคลาดเคลื่อนน้อยกว่าค่าที่เรากำหนดให้อัลกอริทึม แต่เราสามารถเพิ่มประสิทธิภาพด้านความเร็วของในการหาคำตอบของอัลกอริทึมประมาณค่าในกรณีที่เรามีการเรียกใช้อัลกอริทึมประมาณค่าหลาย ๆ ครั้ง โดยการเก็บคำตอบที่เราเคยสร้างไว้และเก็บค่าที่ปรับเปลี่ยนไว้ล่าสุดเพื่อนำมาปรับค่าต่อทันทีในกรณีที่ตารางเก็บค่าไม่พบคำตอบที่เราต้องการ

โดยในบทวิเคราะห์นั้นผู้วิจัยได้แสดงให้เห็นถึงทิศทางของเวลาในการดำเนินการของอัลกอริทึมประมาณค่าและทำการทดลองซ้ำ ๆ หลาย ๆ ครั้งโดยปรับเปลี่ยนพารามิเตอร์ของอัลกอริทึมประมาณค่าที่แตกต่างกันออกไป การศึกษาพบว่าการที่เราจะได้มาซึ่งคำตอบของการประมาณค่าที่มีค่าขอบเขตความคลาดเคลื่อนที่น้อยมาก ๆ หรือเป็นจำนวนเต็มบวกที่ใหญ่มาก ๆ นั้น เราย่อมต้องแลกด้วยเวลาในการดำเนินการที่นานมากขึ้นเท่านั้น

5.2 ข้อเสนอแนะ

ข้อเสนอแนะจากงานวิจัยนี้คือ สำหรับในอัลกอริทึมประมาณค่า (อัลกอริทึมที่ 3.2) ซึ่งมีกระบวนการหาคำตอบจากการเปลี่ยนแปลงค่าในตารางเริ่มต้นซึ่งเป็นฎู่าที่อยู่ในช่วง $(1, 2^8 3^{-5})$ ไปจนกระทั่งพบคำตอบที่ต้องการ หากเปลี่ยนค่าเริ่มต้นใหม่เป็นค่าอื่นจะส่งผลให้เราสามารถลดเวลาในการเข้าถึงคำตอบได้มากขึ้นเท่านั้น และพจน์เดี่ยวที่เกิดจากการประมาณค่านั้นจะมีเลขชี้กำลังซึ่งเป็นจำนวนที่มีค่ามาก ดังนั้นหากเราอาศัยทฤษฎีของพีชคณิตเชิงเส้นในการแปลงเลขชี้กำลังโดยการแปลงเชิงเส้นให้เป็นจำนวนที่น้อยลงเพื่อความสะดวกในการคำนวณและจัดเก็บค่าในหน่วยความจำ จะส่งผลให้กระบวนการหารโดยการประมาณค่าตัวส่วนด้วยพจน์เดี่ยวในระบบจำนวนฐานคู่นั้นมีประสิทธิภาพดียิ่งขึ้น

บรรณานุกรม

1. Dimitrov, V.S., G.A. Jullien, and W.C. Miller. *Theory and applications for a double-base number system*. in *Proceedings 13th IEEE Sympsoium on Computer Arithmetic*. 1997.
2. Adikari, J., V. S. Dimitrov, and R. Cintra, *A New Algorithm for Double Scalar Multiplication over Koblitz Curves*. 2011. 709-712.
3. Avanzi, R., et al. *Extending Scalar Multiplication Using Double Bases*. in *Advances in Cryptology – ASIACRYPT 2006*. 2006. Berlin, Heidelberg: Springer Berlin Heidelberg.
4. Barua, R., S.K. Pandey, and R. Pankaj. *Efficient Window-Based Scalar Multiplication on Elliptic Curves Using Double-Base Number System*. in *Progress in Cryptology – INDOCRYPT 2007*. 2007. Berlin, Heidelberg: Springer Berlin Heidelberg.
5. Bernstein, D.J., et al. *Optimizing Double-Base Elliptic-Curve Single-Scalar Multiplication*. in *Progress in Cryptology – INDOCRYPT 2007*. 2007. Berlin, Heidelberg: Springer Berlin Heidelberg.
6. Dimitrov, V., L. Imbert, and P.K. Mishra. *Efficient and Secure Elliptic Curve Point Multiplication Using Double-Base Chains*. in *Advances in Cryptology - ASIACRYPT 2005*. 2005. Berlin, Heidelberg: Springer Berlin Heidelberg.
7. Doche, C. and L. Imbert. *Extended Double-Base Number System with Applications to Elliptic Curve Cryptography*. in *Progress in Cryptology - INDOCRYPT 2006*. 2006. Berlin, Heidelberg: Springer Berlin Heidelberg.
8. Doche, C., D.R. Kohel, and F. Sica. *Double-Base Number System for Multi-scalar Multiplications*. in *Advances in Cryptology - EUROCRYPT 2009*. 2009. Berlin, Heidelberg: Springer Berlin Heidelberg.
9. Doche, C. and D. Sutantyo, *New and Improved Methods to Analyze and Compute Double-Scalar Multiplications.* IEEE Transactions on Computers, 2014. **63**(1): p. 230-242.
10. Meloni, N. and M. Anwar Hasan, *Elliptic Curve Point Scalar Multiplication*

Combining Yao's Algorithm and Double Bases. Vol. 5747. 2009. 304-316.

11. S. Dimitrov, V., L. Imbert, and P. K. Mishra, *Fast Elliptic Curve Point Multiplication using Double-Base Chains*. Vol. 2005. 2005. 69.
12. Yu, W., et al. *Triple-Base Number System for Scalar Multiplication*. in *Progress in Cryptology – AFRICACRYPT 2013*. 2013. Berlin, Heidelberg: Springer Berlin Heidelberg.
13. Berthé, V. and L. Imbert, *On Converting Numbers to the Double-Base Number System*. Vol. 5559. 2004.
14. Mishra, P.K. and V. Dimitrov. *A Graph Theoretic Analysis of Double Base Number Systems*. in *Progress in Cryptology – INDOCRYPT 2007*. 2007. Berlin, Heidelberg: Springer Berlin Heidelberg.
15. Dimitrov, V.S., G.A. Jullien, and W.C. Miller, *Theory and applications of the double-base number system.* IEEE Transactions on Computers, 1999. **48**(10): p. 1098-1106.
16. Pichan Pratummal, *An extended double dimensional logarithmic number system*, in *Computer Engineering*. 2007, Chulalongkorn University: Chulalongkorn University.

ประวัติผู้เขียน

ชื่อ-สกุล	ธีรภัทร์ ชุนเดชสัมฤทธิ์
วัน เดือน ปี เกิด	วันที่ 9 พฤศจิกายน พ.ศ. 2536
สถานที่เกิด	จังหวัดนนทบุรี
วุฒิการศึกษา	วิทยาศาสตรบัณฑิต (วท.บ.) คณะวิทยาศาสตร์และเทคโนโลยี สาขาวิชาคณิตศาสตร์ ภาควิชาคณิตศาสตร์และสถิติ มหาวิทยาลัยธรรมศาสตร์
ที่อยู่ปัจจุบัน	บ้านเลขที่ 37/13 หมู่ที่ 3 ถนนเทศบาล 6 ตำบลโสนลอย อำเภอบางบัวทอง จังหวัดนนทบุรี รหัสไปรษณีย์ 11110
ผลงานตีพิมพ์	T. Chundetsumrit and A. Surarerks, "Approximation of Positive Integer in Double-Base Number System with A Single Term", 2nd European Conference on Electrical Engineering & Computer Science (EECS 2018), December 20-22, 2018, Bern, Switzerland.