บทที่ 2
การวิเคราะห์อัลกอริทึม

การแก้ไขปัญหาหนึ่งๆ ส่วนใหญ่จะมีขั้นตอนวิธีหรือมีอัลกอริทึมที่สามารถแก้ปัญหานั้นได้หลายวิธี โดยในแต่ละวิธีจะใช้เวลาในการแก้ปัญหาแตกต่างกันไป ซึ่งในบางวิธีจะสามารถแก้ปัญหานั้นในบางกรณีได้ดี คืออาจจะมีวิธีหนึ่งที่สามารถแก้ปัญหากรณี (a) ได้ดีกว่าวิธีอื่นๆ แต่วิธีดังกล่าวนี้เมื่อนำไปแก้ปัญหาในกรณี (b) แล้วจะด้อยกว่าวิธีอื่น ดังนั้นในการวิเคราะห์ประสิทธิภาพนั้นจะต้องคำนวณหาเวลาดำเนินการของแต่ละวิธี เพื่อที่จะวิเคราะห์ได้ว่าควรใช้วิธีไหนในการแก้ปัญหานั้นๆ ในปัจจุบันการวิเคราะห์อัลกอริทึมมี 3 แบบด้วยกัน คือ การวิเคราะห์กรณีร้ายแรงที่สุด การวิเคราะห์กรณีเฉลี่ย และการวิเคราะห์ถัวเฉลี่ย ในที่นี้จะขอใช้ปัญหาการนับเลขฐานสองเป็นตัวอย่าง ในการแสดงการวิเคราะห์

พิจารณาการนับเลขฐานสอง 8 บิตที่เริ่มนับจาก 0, 1, 2, ... , n-1 โดยจะใช้แถวลำดับ (array) A [0 ... 7] แทนบิตแต่ละบิต บิตต่ำสุดคือ A[0] และบิตสูงสุดคือ A[7] ดังรูปที่ 2.1 แสดงการนับเลขฐานสอง 8 บิต ที่เริ่มนับจาก 0, 1, ... , 15 แสดงบิตที่ถูกเปลี่ยนค่าในการเพิ่มครั้งต่อไปด้วยแถบสีดำ ที่ซึ่งเวลาในการดำเนินงานเป็นสัดส่วนโดยตรงกับจำนวนการเปลี่ยนค่าของบิต จากบิต 1 เป็นบิต 0 และจากบิต 0 เป็นบิต 1

การทำงานการเพิ่มค่าเลขฐานสองไปอีกหนึ่ง สามารถเขียนเป็นอัลกอริทึมได้ดังนี้[14]

```
i = 0
while i < length[A] and A[i] = 1
      do A[i] = 0
           i = i+1
      end do
if i < length[A]
      then A[i] = 1
```

การวิเคราะห์กรณีร้ายแรงสุด

สำหรับการวิเคราะห์นี้ เวลาที่ใช้ดำเนินงานในกรณีร้ายแรงสุดจะเป็นเวลาที่เป็นขอบเขตบนของการทำงาน กล่าวคือเวลาที่ใช้ในการดำเนินงานนั้น จะไม่มีการใช้เวลาเกินกว่าเวลาในกรณีร้ายแรงสุด แม้ว่าข้อมูลที่อินพุตมาจะมีลักษณะใดก็ตาม

พิจารณาการนับเลขฐานสองในรูปที่ 2.1 ถ้า $i > \lfloor \log_2 n \rfloor$ บิต A[i] จะไม่เปลี่ยนค่า ดังนั้นในกรณีร้ายแรงสุด สำหรับ $i = 0, 1, \ldots , \lfloor \log_2 n \rfloor$ บิต A[i] เปลี่ยนค่า n ครั้งเท่ากันหมด เวลาดำเนินงานรวมกรณีร้ายแรงสุด ($W(n)$) จะเป็น

$$W(n) = \sum_{i=0}^{\log_2 n} n = \sum_{i=1}^{(\log_2 n)+1} n = n((\log_2 n) + 1)$$

$$\approx \quad n \log_2 n$$

ดังนั้นสำหรับการนับเลขฐานสอง เวลารวมที่ใช้ในกรณีร้ายแรงสุดจะเป็นจะมี O($n$log n) หรือเวลาที่ใช้ในกรณีร้ายแรงสุดจะเป็น O(log n)

ค่าการนับ	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	รวมเวลาที่ใช้
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	3
3	0	0	0	0	0	0	1	1	4
4	0	0	0	0	0	1	0	0	7
5	0	0	0	0	0	1	0	1	8
6	0	0	0	0	0	1	1	0	10
7	0	0	0	0	0	1	1	1	11
8	0	0	0	0	1	0	0	0	15
9	0	0	0	0	1	0	0	1	16
10	0	0	0	0	1	0	1	0	18
11	0	0	0	0	1	0	1	1	19
12	0	0	0	0	1	1	0	0	22
13	0	0	0	0	1	1	0	1	23
14	0	0	0	0	1	1	1	0	25
15	0	0	0	0	1	1	1	1	26

รูปที่ 2.1 การนับเลขฐานสอง

การวิเคราะห์กรณีเฉลี่ย

สำหรับการวิเคราะห์นี้ เวลาที่ใช้ดำเนินงานจะเป็นค่าคาดหมายของการดำเนินงาน ซึ่งเป็นไปได้ที่จะคาดผิดไป กล่าวคือในการดำเนินงานนั้นจริงๆแล้ว อาจใช้เวลาไปมากกว่าหรือน้อยกว่าเวลาที่ได้คาดไว้ เพราะเวลาที่คาดหมายไว้ เป็นเพียงสมมติฐานของความน่าจะเป็นเท่านั้น

พิจารณาการนับเลขฐานสอง ที่เริ่มนับจาก 0, 1, … , n-1 ถ้า $i > \lfloor \log_2 n \rfloor$ บิต A[i] จะไม่เปลี่ยนค่าเลย สำหรับ $i = 0, 1, \ldots, \lfloor \log_2 n \rfloor$

สมมติให้ $p(I_i)$ เป็น ความน่าจะเป็นที่จะเกิดการเปลี่ยนค่าของบิต A[i]

$t(I_i)$ เป็น เวลาดำเนินงานการเปลี่ยนค่าของบิต A[i]

และ $A(n)$ เป็น เวลาดำเนินงานรวมกรณีเฉลี่ย

แล้ว $$p(I_i)t(I_i) = \sum_{i=0}^{n-1}\frac{1}{n}i = \frac{1}{n}\sum_{i=1}^{n}i = \frac{n+1}{2} \approx \frac{n}{2}$$

$$A(n) = \sum_{i=0}^{\lfloor \log_2 n \rfloor}\frac{n}{2} = \sum_{i=1}^{(\log_2 n)+1}\frac{n}{2} = (\log_2 n + 1)\frac{n}{2} \approx n\log_2 n$$

ดังนั้น เวลารวมที่ใช้ในกรณีเฉลี่ย สำหรับการนับเลขฐานสอง จะเป็น O($n$log n) หรือเวลาที่ใช้ในกรณีเฉลี่ย จะเป็น O(log n)

การวิเคราะห์ถัวเฉลี่ย

สำหรับการวิเคราะห์นี้ จะเป็นการเฉลี่ยเวลาที่เป็นไปได้มากที่สุดในแต่ละการดำเนินงาน ซึ่งในการวิเคราะห์แบบนี้ เปรียบการแทนเวลาที่ใช้จริงเป็นเงินต้นทุนหรือต้นทุนจริง และแทนการเฉลี่ยเวลาที่เป็นไปได้มากที่สุดเป็นต้นทุนถัวเฉลี่ย ถ้าเวลาในแต่ละการดำเนินงานส่วนใหญ่มีต้นทุนจริงถูก การวิเคราะห์แบบนี้จะทำให้มีต้นทุนถัวเฉลี่ยถูก แม้ว่าบางการดำเนินงานจะมีต้นทุนจริงสูงก็ตาม ซึ่งถ้าเทียบกับการวิเคราะห์กรณีร้ายแรงสุดแล้ว เมื่อมีบางการดำเนินงานมีต้นทุนจริงสูงต้นทุนโดยรวมก็ต้องสูง แม้ว่าการดำเนินงานส่วนใหญ่แล้วมีต้นทุนต่ำก็ตาม

สำหรับเทคนิคที่ใช้ในการวิเคราะห์แบบนี้ มี 3 เทคนิควิธี คือ

1. วิธีรวมกลุ่ม (Aggregate method) โดยจะแสดงให้เห็นว่า สำหรับการดำเนินงาน n ลำดับ ต้นทุนเฉลี่ยเป็น $\frac{T(n)}{n}$ โดยที่ $T(n)$ เป็นผลรวมต้นทุนทุกๆการดำเนินงาน

พิจารณาการนับเลขฐานสองในรูปที่ 2.1 เมื่อ n = 16 พบว่าบิต A[1] จะเปลี่ยนค่า 8 ครั้ง หรือ $\left\lfloor \frac{n}{2} \right\rfloor$ ครั้ง บิต A[2] จะเปลี่ยนค่า 4 ครั้ง หรือ $\left\lfloor \frac{n}{4} \right\rfloor$ ครั้ง และถ้า $i > \lfloor \log_2 n \rfloor$ บิต A[i] จะไม่เปลี่ยนค่าเลย ดังนั้นในการดำเนินงานของการนับที่เริ่มจาก 0 ไปจนถึง n-1 สำหรับ i = 0, 1, … , $\lfloor \log_2 n \rfloor$ บิต A[i] จะเปลี่ยนค่า $\left\lfloor \frac{n}{2^i} \right\rfloor$ ครั้ง ผลรวมจำนวนของการเปลี่ยนค่าเป็น [14]

$$\sum_{i=0}^{\lfloor \log_2 n \rfloor}\left\lfloor \frac{n}{2^i} \right\rfloor < n\sum_{i=0}^{\infty}\frac{1}{2^i} = 2n$$

นั่นคือ ต้นทุนถัวเฉลี่ยในการนับเลขฐานสองที่เริ่มนับจาก 0 ถึง n-1 เป็น 2n และต้นทุนถัวเฉลี่ยสำหรับแต่ละการดำเนินงานเป็น $\frac{2n}{n} = 2$ หรือเวลาที่ใช้ในแบบถัวเฉลี่ยในวิธีรวมกลุ่มเป็น O(1)

2. วิธีบัญชี (Accounting method) สำหรับวิธีนี้ จะต้องมีการกำหนดต้นทุนถัวเฉลี่ยในแต่ละการดำเนินงานก่อน โดยต้นทุนถัวเฉลี่ยที่กำหนดอาจมากกว่าหรือน้อยกว่าต้นทุนจริงก็ได้ และถ้าการดำเนินงานใดมีต้นทุนถัวเฉลี่ยมากกว่าต้นทุนจริงแล้ว ความแตกต่างของต้นทุนถัวเฉลี่ยกับต้นทุน

จริงจะถูกบันทึกในโครงสร้างข้อมูลที่เรียกว่า เครดิต ซึ่งเครดิตนี้จะลดลงได้ ถ้าการดำเนินงานใดๆ มีต้นทุนถัวเฉลี่ยน้อยกว่าต้นทุนจริง ดังนั้นในวิธีนี้จะมองต้นทุนถัวเฉลี่ย แยกเป็น 2 ส่วน คือ (1.) ต้นทุนจริง (2.) เครดิต ซึ่งอาจมีเพิ่มขึ้นหรือลดลง ซึ่งจะต่างกับวิธีรวมกลุ่มที่ต้นทุนถัวเฉลี่ยจะไม่แยกเป็น 2 ส่วน ดังเช่นในวิธีนี้ อย่างไรก็ตาม ต้นทุนถัวเฉลี่ยรวมของลำดับการดำเนินงานจะต้องเป็นขอบเขตบนของต้นทุนจริงรวมของลำดับการดำเนินงาน ดังนั้นเครดิตเมื่อสุทธิแล้วต้องไม่ติดลบ เพราะเครดิตสุทธิแสดงถึงจำนวนของต้นทุนถัวเฉลี่ยรวมที่ต้องมากกว่าต้นทุนจริงรวม[14]

พิจารณาการนับเลขฐานสอง ดังรูปที่ 2.2 กำหนดให้ต้นทุนจริงในการเปลี่ยนบิตเป็น 1 บาท และให้ต้นทุนถัวเฉลี่ยของการเซตบิตให้เป็น 1 เป็น 2 บาท โดยใช้ 1 บาทเพื่อจ่ายสำหรับการเซตบิตให้เป็น 1 และอีก 1 บาท ถูกจัดเก็บไว้เป็นเครดิตของบิตนั้นๆ นั่นคือที่บิต 1 ทุกๆบิต จะมีเครดิตอยู่ 1 บาท เพื่อใช้เป็นค่าใช้จ่ายในการเซตบิต 1 กลับเป็นบิต 0 ในอนาคต จากต้นทุนถัวเฉลี่ย 2 บาทที่ใช้ในการนับแต่ละครั้งที่กำหนดให้นั้น พบว่าไม่มีเครดิตที่บิตใดติดลบเสมอระหว่างนับ (นั่นคือ ต้นทุนถัวเฉลี่ยรวมไม่เคยน้อยกว่าต้นทุนจริงรวม) ดังนั้นต้นทุนถัวเฉลี่ยที่กำหนดไว้ 2 บาท ใช้ได้ จึงแสดงว่าเวลาการทำงานถัวเฉลี่ยของการนับแต่ละครั้งจึงเป็น 2 หรือเวลาที่ใช้ในกรณีถัวเฉลี่ยในวิธีบัญชีเป็น O(1) [14]

ค่าการนับ	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	ต้นทุนจริง	ต้นทุนจริงรวม	ต้นทุนถัวเฉลี่ยรวม
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1(1)	1	1	2
2	0	0	0	0	0	0	1(1)	0	2	3	4
3	0	0	0	0	0	0	1(1)	1(1)	1	4	6
4	0	0	0	0	0	1(1)	0	0	3	7	8
5	0	0	0	0	0	1(1)	0	1(1)	1	8	10
6	0	0	0	0	0	1(1)	1(1)	0	2	10	12
7	0	0	0	0	0	1(1)	1(1)	1(1)	1	11	14
8	0	0	0	0	1(1)	0	0	0	4	15	16
9	0	0	0	0	1(1)	0	0	1(1)	1	16	18
10	0	0	0	0	1(1)	0	1(1)	0	2	18	20
11	0	0	0	0	1(1)	0	1(1)	1(1)	1	19	22
12	0	0	0	0	1(1)	1(1)	0	0	3	22	24
13	0	0	0	0	1(1)	1(1)	0	1(1)	1	23	26
14	0	0	0	0	1(1)	1(1)	1(1)	0	2	25	28
15	0	0	0	0	1(1)	1(1)	1(1)	1(1)	1	26	30

รูปที่ 2.2 การนับเลขฐานสอง กับการวิเคราะห์ถัวเฉลี่ยในวิธีบัญชี

3. วิธีศักย์ (Potential method) สำหรับในวิธีนี้ จะมองในแง่ของการสะสมของฟังก์ชันศักย์ กล่าวคือ ถ้าการดำเนินงานใดมีต้นทุนถัวเฉลี่ยมากกว่าต้นทุนจริงแล้ว จะเป็นการสะสมฟังก์ชันศักย์ โดยสะสมเป็นจำนวนเท่ากับผลต่างของต้นทุนถัวเฉลี่ยกับต้นทุนจริง ทำนองเดียวกัน ถ้าการดำเนินงานใดมีต้นทุนถัวเฉลี่ยน้อยกว่าต้นทุนจริงแล้ว จะเป็นการลดฟังก์ชันศักย์เป็นจำนวนเท่ากับผลต่างของต้นทุนถัวเฉลี่ยกับต้นทุนจริง ซึ่งฟังก์ชันศักย์ที่สะสมนี้ จะเป็นส่วนที่สะสมของทั้งระบบ ต่างกับวิธีบัญชีที่เครดิตนั้น จะเป็นเงินที่กำกับอยู่กับข้อมูลต่างๆในระบบ

สำหรับการดำเนินงาน n ลำดับ ที่ $i = 1, 2, \ldots, n$

ให้ C_i เป็น ต้นทุนจริงของการดำเนินงานที่ i

Φ_i เป็น ฟังก์ชันศักย์ของการดำเนินงานที่ i

Φ_{i-1} เป็น ฟังก์ชันศักย์ของการดำเนินงานที่ i-1

$\hat{C}_i$ เป็น ต้นทุนถัวเฉลี่ยของการดำเนินงานที่ i

เพราะฉะนั้น [14]

$$\hat{C}_i = C_i + \Phi_i - \Phi_{i-1}$$

พิจารณา การนับเลขฐานสอง ดังรูปที่ 2.3 ต้นทุนจริงเป็นภาระการเปลี่ยนบิต และให้ฟังก์ชันศักย์เป็นจำนวนบิต 1

กำหนดให้ b_i คือ จำนวนบิตที่เป็น 1 หลังการดำเนินงานที่ i

t_i คือ จำนวนของบิตที่ถูกเซตให้เป็น 0 ในการดำเนินงานที่ i

ดังนั้น $C_i \le t_i + 1$ เนื่องจากต้นทุนจริงอย่างมากคือภาระการเปลี่ยนบิตจาก 1 เป็น 0 จำนวน t_i บิต และการเปลี่ยนบิต 0 เป็น 1 อีกอย่างมาก 1 บิต สำหรับจำนวนบิต 1 หลังการดำเนินงาน i (b_i) อย่างน้อยเป็นจำนวนบิต 1 ในการดำเนินงานที่ i (b_{i-1}) หักออกตามจำนวนบิต 0 ในการดำเนินงานที่ i (t_i) เนื่องจาก บิต 1 ในการดำเนินงานที่ i ไม่จำเป็นต้องถูกเซตเป็น 1 และบวกอีก 1 จากการเปลี่ยนบิต 0 เป็นบิต 1 อย่างมาก 1 บิตในทุกๆการดำเนินงาน

ดังนั้น $$b_i \le b_{i-1} - t_i + 1$$

$$\Phi_i = b_i$$

$$\Phi_i - \Phi_{i-1} = b_i - b_{i-1} \le (b_{i-1} - t_i + 1) - b_{i-1} = 1 - t_i$$

ดังนั้นต้นทุนถัวเฉลี่ย เป็น

$$\hat{C}_i = C_i + \Phi_i - \Phi_{i-1}$$

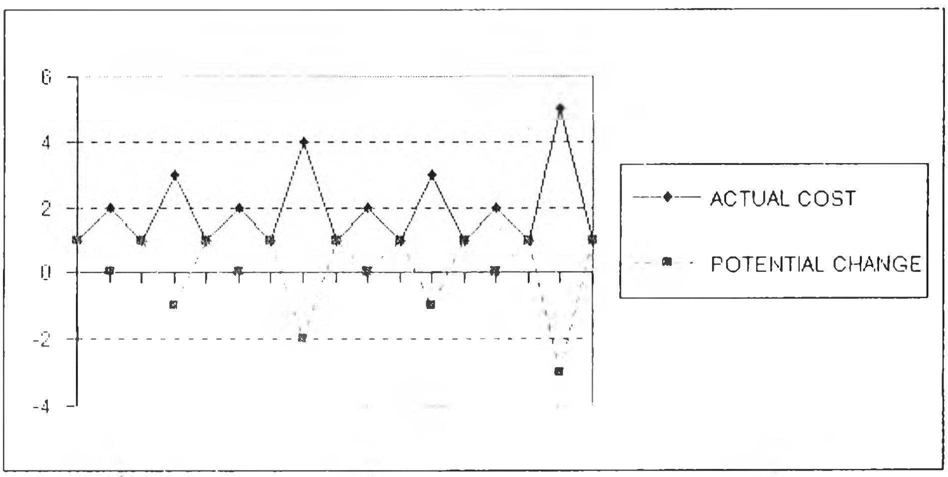
$$\le (t_i + 1) + (1 - t_i) = 2$$

และ $$\sum_{i=1}^{n} \hat{C}_i = 2n$$

ถ้านำต้นทุนจริง และความแตกต่างฟังก์ชันศักย์ของการนับเลขฐานสอง ที่เริ่มนับจาก 1 ไปจนถึง 17 มาพล็อตกราฟ จะได้เป็นดังรูปที่ 2.4

ค่าการนับ	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	ต้นทุนจริง	ต้นทุนจริงรวม	ฟังก์ชันศักย์	ต้นทุนถัวเฉลี่ย	ต้นทุนถัวเฉลี่ยรวม
0	0	0	0	0	0	0	0	0	0	0	0	0+0=0	0
1	0	0	0	0	0	0	0	1	1	1	1	1+1=2	2
2	0	0	0	0	0	0	1	0	2	3	1	2+0=2	4
3	0	0	0	0	0	0	1	1	1	4	2	1+1=2	6
4	0	0	0	0	0	1	0	0	3	7	1	3-1=2	8
5	0	0	0	0	0	1	0	1	1	8	2	1+1=2	10
6	0	0	0	0	0	1	1	0	2	10	2	2+0=2	12
7	0	0	0	0	0	1	1	1	1	11	3	1+1=2	14
8	0	0	0	0	1	0	0	0	4	15	1	4-2=2	16
9	0	0	0	0	1	0	0	1	1	16	2	1+1=2	18
10	0	0	0	0	1	0	1	0	2	18	2	2+0=2	20
11	0	0	0	0	1	0	1	1	1	19	3	1+1=2	22
12	0	0	0	0	1	1	0	0	3	22	2	3-1=2	24
13	0	0	0	0	1	1	0	1	1	23	3	1+1=2	26
14	0	0	0	0	1	1	1	0	2	25	3	2+0=2	28
15	0	0	0	0	1	1	1	1	1	26	4	1+1=2	30

รูปที่ 2.3 การนับเลขฐานสอง กับการวิเคราะห์ถัวเฉลี่ย ในวิธีศักย์



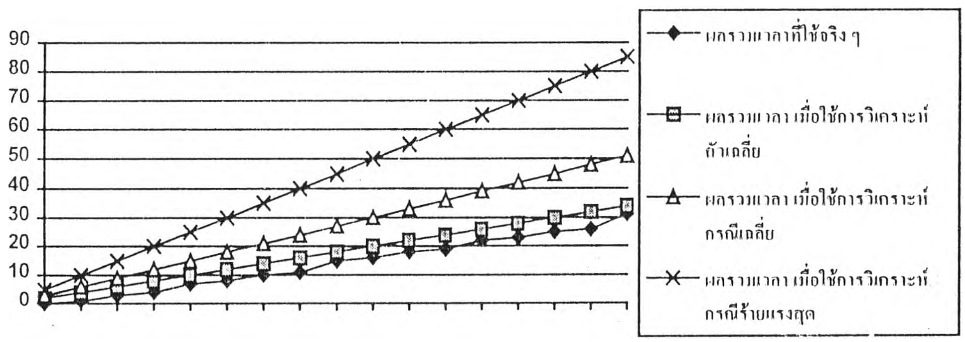
รูปที่ 2.4 กราฟแสดงความสัมพันธ์ ระหว่างต้นทุนจริง และความแตกต่างของฟังก์ชันศักย์

จากรูปที่ 2.3 และ 2.4 พบว่า ถ้าการดำเนินงานใดมีต้นทุนจริงสูงกว่าต้นทุนถัวเฉลี่ย จะเป็นการไปลดการสะสมของฟังก์ชันศักย์ เปรียบเสมือนว่าถ้าการดำเนินงานใดมีต้นทุนจริงสูง จนกระทั่งต้นทุนถัวเฉลี่ยมีไม่พอ ก็ต้องไปเอาฟังก์ชันศักย์ที่สะสมไว้มาใช้แทน ซึ่งจะเอามาใช้มากหรือน้อยขึ้นอยู่กับว่าเราต้องการอีกเท่าใด หรือก็คือยิ่งต้นทุนจริงสูงกว่าต้นทุนถัวเฉลี่ยมากเท่าไรก็

ยิ่งต้องเอาฟังก์ชันศักย์ที่สะสมไว้มาใช้มากขึ้นเท่านั้น และเช่นกันถ้าการดำเนินงานใดมีต้นทุนจริงต่ำกว่าต้นทุนถัวเฉลี่ยจะเป็นการไปเพิ่มการสะสมของฟังก์ชันศักย์ จุดที่ต้องสังเกตคือ ณ ขณะใดขณะหนึ่งระหว่างการดำเนินงาน ฟังก์ชันศักย์ที่สะสมอยู่จะน้อยกว่าฟังก์ชันศักย์ตอนเริ่มต้นไม่ได้ เพราะหากเป็นเช่นนั้น แสดงว่าฟังก์ชันศักย์ที่ตั้งขึ้นนั้นผิด

ดังนั้น เวลาที่ใช้แบบถัวเฉลี่ยสำหรับการนับเลขฐานสอง ที่เริ่มนับตั้งแต่ 0 ถึง n จะเป็น $2n$ $O(n)$ และ เวลาที่ใช้แบบถัวเฉลี่ยของแต่ละการดำเนินงาน จะเป็น $\frac{O(n)}{n} = O(1)$

พิจารณา การนับเลขฐานสองกับการวิเคราะห์กรณีร้ายแรงสุด การวิเคราะห์กรณีเฉลี่ย และการวิเคราะห์แบบถัวเฉลี่ย ถ้าใช้การวิเคราะห์กรณีร้ายแรงสุด หรือใช้การวิเคราะห์กรณีเฉลี่ย เวลาที่ใช้โดยรวมจะมีค่าบิกโอเป็น $O(n\log n)$ แต่ถ้าใช้การวิเคราะห์ถัวเฉลี่ย เวลาที่ใช้โดยรวมจะดีขึ้น คือใช้เวลาที่มีค่าบิกโอเป็น $O(n)$ ดังแสดงเป็นกราฟ ในรูปที่ 2.5 ซึ่งจะเห็นได้ชัดว่า ยิ่งข้อมูลมีมากขึ้นเท่าไร เส้นกราฟของการวิเคราะห์กรณีร้ายแรงสุด และเส้นกราฟของการวิเคราะห์กรณีเฉลี่ย จะยิ่งห่างจากเส้นกราฟของเวลาที่ใช้จริง ๆ ไปมากขึ้น ซึ่งต่างกับเส้นกราฟของการวิเคราะห์ถัวเฉลี่ย ที่เมื่อข้อมูลมีมากขึ้นเท่าไร เส้นกราฟก็ยังคงใกล้เคียงกับเส้นกราฟของเวลาที่ใช้จริง ๆ [11]



รูปที่ 2.5 กราฟแสดงความสัมพันธ์ระหว่าง การวิเคราะห์ถัวเฉลี่ย การวิเคราะห์กรณีเฉลี่ย และการวิเคราะห์กรณีร้ายแรงสุด