บทที่ 1

อัลกอริทึมและโครงสร้างข้อมูลแบบสุ่ม

ในส่วนนี้เราจะพิจารณาอัลกอริทึมและโครงสร้างข้อมูลที่มีการใช้ความสุ่มประกอบการทำงาน ตัวอย่างเช่น อัลกอริทึมการจัดเรียงแบบควิกที่เลือกไพวอตแบบสุ่ม จากที่เราเคยได้พิจารณาแล้วในบทที่ **??** อัลกอริทึมดังกล่าวใช้เวลาทำงานในกรณีที่แย่ที่สุดเป็น $O(n^2)$ อย่างไรก็ตาม อัลกอริทึมพวกนี้มักใช้ความสุ่มทำให้กรณีที่แย่ที่สุดมีโอกาสเกิดขึ้นน้อย

นอกจากนี้ เราจะได้วิเคราะห์ประสิทธิภาพการทำงานของฟังก์ชันแฮชที่เลือกแบบสุ่ม

1.1 พื้นฐานเกี่ยวกับทฤษฎีความน่าจะเป็น

ในการจะวิเคราะห์อัลกอริทึมหรือโครงสร้างข้อมูลที่เกี่ยวข้องกับการสุ่ม เราจำเป็นจะต้องเข้าใจแนวคิดพื้นฐานเกี่ยวกับทฤษฎีความน่าจะเป็นก่อน

เราจะเรียกกิจกรรมที่มีความสุ่มเกี่ยวข้องว่า*การทดลองสุ่ม* ซึ่งเมื่อทดลองเสร็จจะได้*ผลลัพธ์* (outcome) ซึ่งเราอาจจะสังเกตได้หรือไม่ก็ตาม เราเรียกเซตของผลลัพธ์ที่เป็นไปได้ทั้งหมดว่า*ปริภูมิตัวอย่าง* (sample space) ที่มักเขียนแทนด้วย Ω ในที่นี้เราจะสนใจเฉพาะกรณีที่ปริภูมิตัวอย่างมีขนาดจำกัดเท่านั้น

เมื่อเราจะกล่าวถึงความน่าจะเป็น เราจะพิจารณาผลลัพธ์ ω ทุก ๆ ผลลัพธ์ใน Ω และกำหนดค่ามวลความน่าจะเป็น $p(\omega)$ ซึ่งเป็นจำนวนจริงไม่เป็นลบให้กับแต่ละผลลัพธ์ ค่ามวลความน่าจะเป็นนี้จะต้องสอดคล้องกับเงื่อนไข $\sum_{\omega\in\Omega} p(\omega) = 1$ นั่นคือผลรวมของความน่าจะเป็นของผลลัพธ์ทั้งหมดจะต้องเท่ากับ 1

พิจารณาตัวอย่างง่าย ๆ ดังนี้ สมมติว่าเรามีถุงทึบบรรจุลูกบอล 4 ลูก สีขาว สีแดง และสีน้ำเงิน 2 ลูก ถ้าการทดลองสุ่มของเราคือการหยิบลูกบอลออกมาหนึ่งลูก เราจะได้ว่าปริภูมิตัวอย่างคือเซต $\{white, red, blue1, blue2\}$ ในกรณีนี้ถ้าเราทราบว่าความน่าจะเป็นที่จะได้ลูกบอลลูกใด ๆ มีค่าเท่ากัน เราจะได้ว่า $p(white) = p(red) = p(blue1) = p(blue2)$ อย่างไรก็ตามเนื่องจากผลรวมของ มวลความน่าจะเป็นต้องเท่ากับ 1 เราจึงได้ว่า $p(white) = p(red) = p(blue1) = p(blue2) = 1/4$

โดยปกติแล้วเราจะไม่ได้สนใจความน่าจะเป็นของผลลัพธ์ใด ๆ อย่างเฉพาะเจาะจง แต่เรามักสนใจกลุ่มของผลลัพธ์ที่มีคุณสมบัติที่เราต้องการ เช่น หยิบได้ลูกบอลสีน้ำเงินเป็นต้น เราจะเรียกเซตของผลลัพธ์ว่า*เหตุการณ์* (event) และความน่าจะเป็นของเหตุการณ์ $A \subseteq \Omega$ หรือเขียนแทนว่า $\Pr[A]$ จะมีค่าเท่ากับ

$$\Pr[A] = \sum_{\omega\in A} p(\omega)$$

จากตัวอย่างข้างต้น เราให้เหตุการณ์ B ที่หยิบได้ลูกบอลสีน้ำเงินแทนเซต $\{blue1, blue2\}$ ดังนั้น

$$\Pr[B] = p(blue1) + p(blue2) = 1/2$$

นอกจากที่เราจะสนใจว่าเหตุการณ์ใด ๆ เกิดขึ้นหรือไม่แล้ว หลายครั้งเรายังสนใจ "ค่า" จากผลลัพธ์ของการทดลอง ยกตัวอย่างเช่นถ้าเรามีลูกเต๋าที่มีหน้าเป็นค่าจำนวนเฉพาะ (2,3,5,7,11, และ 13) เมื่อทอยลูกเต๋าเสร็จ เราก็จะสามารถกล่าวถึง

ค่าของหน้าลูกเต๋าที่ออกได้ ค่าที่แปรไปตามผลลัพธ์ของการทดลองนี้ เรียกว่า *ตัวแปรสุ่ม* กล่าวคือ ตัวแปรสุ่มคือฟังก์ชันจากปริภูมิตัวอย่างไปยังจำนวนจริง

ให้ X เป็นตัวแปรสุ่มแทนค่าที่ลูกเต๋าจำนวนเฉพาะออก สังเกตว่าภายใต้นิยามที่เรากล่าวมา $X = 2$ หรือ $X \geq 5$ ก็เป็นเหตุการณ์ โดยที่เหตุการณ์แรกคือเซตของผลลัพธ์ $\{2\}$ ส่วนเหตุการณ์ที่สองคือเซต $\{5, 7, 11, 13\}$

เมื่อให้ตัวแปรสุ่ม X ที่มีค่าเป็นจำนวนเต็ม เรานิยาม*ค่าคาดหวัง* (expected value) ของ X เป็น

$$\mathrm{E}[X] = \sum_{i=-\infty}^{\infty} i \cdot \Pr[X = i]$$

จากตัวอย่างเราจะได้ว่า $\mathrm{E}[X] = 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 7 \cdot \frac{1}{6} + 11 \cdot \frac{1}{6} + 13 \cdot \frac{1}{6} \approx 6.8333$

คุณสมบัติที่สำคัญของค่าคาดหวังก็คือคุณสมบัติเชิงเส้น (linearity of expectations) ที่ระบุว่าสำหรับตัวแปรสุ่ม X และ Y ใด ๆ เราจะได้ว่า

$$\mathrm{E}[X + Y] = \mathrm{E}[X] + \mathrm{E}[Y]$$

และในกรณีที่เรามีตัวแปรสุ่มมากกว่าสองตัว คุณสมบัติดังกล่าวก็ยังเป็นจริงเช่นเดียวกัน กล่าวคือ ให้ $X_1, X_2, \ldots, X_k$ เป็นตัวแปรสุ่ม เราจะได้ว่า

$$\mathrm{E}\left[\sum_{i=1}^{k} X_i\right] = \sum_{i=1}^{k} \mathrm{E}[X_i]$$

1.2 การวิเคราะห์การจัดเรียงข้อมูลแบบควิกที่มีการเลือกไพวอตแบบสุ่ม

การเรียงข้อมูลแบบควิกเป็นอัลกอริทึมแบบแบ่งแยกและเอาชนะ กล่าวคือ ในการแก้ปัญหาการเรียงข้อมูลนั้น อัลกอริทึมเรียงข้อมูลแบบควิกจะหยิบข้อมูลหนึ่งตัวเรียกว่าเป็นไพวอต (pivot) จากนั้นจะแบ่งข้อมูลออกเป็นสองกลุ่ม กลุ่มแรกมีค่าไม่เกินไพวอต กลุ่มที่สองมีค่ามากกว่าไพวอต จากนั้นก็จะเรียกตัวเองเพื่อเรียงข้อมูลแต่ละกลุ่ม เมื่อเรียงข้อมูลเสร็จแล้วก็จะนำผลลัพธ์จากปัญหาย่อยมาต่อกันเป็นรายการของข้อมูลที่เรียงลำดับกันนั่นเอง

ในการวิเคราะห์อัลกอริทึมนี้ เราจะนับจำนวนครั้งที่มีการเปรียบเทียบเกิดขึ้น เพราะนั่นจะระบุเวลาการทำงาน

เราจะวิเคราะห์โดยมองย้อนกลับไปจากผลลัพธ์ เราจะสนใจเหตุการณ์ที่ข้อมูลที่มีอันดับที่ i ในรายการที่เรียงแล้วถูกเปรียบเทียบกับข้อมูลที่มีอันดับ j กล่าวคือให้เหตุการณ์ A_{ij} แทนเหตุการณ์ที่ข้อมูลที่มีอันดับที่ i ถูกเปรียบเทียบกับข้อมูลที่มีอันดับ j

ถ้าเหตุการณ์ A_{ij} เกิดขึ้น แสดงว่ามีการเปรียบเทียบกันเกิดขึ้น 1 ครั้ง เรานิยามตัวแปรสุ่ม X_{ij} ให้มีค่าเป็น 1 ถ้า A_{ij} เกิดขึ้นและให้มีค่าเป็น 0 ถ้าเหตุการณ์ A_{ij} ไม่เกิด

สังเกตว่า

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} X_{ij}$$

เป็นตัวแปรสุ่มที่แทนจำนวนครั้งในการเปรียบเทียบทั้งหมดของอัลกอริทึม เราจะวิเคราะห์ค่าคาดหวังของ X_{ij} แยกทีละตัว จากนั้นจะใช้คุณสมบัติการเป็นเชิงเส้นของค่าคาดหวัง ในการหาจำนวนครั้งเฉลี่ยของจำนวนการเปรียบเทียบทั้งหมด

เพื่อความสะดวก ต่อไปเราจะเรียกข้อมูลที่มีค่าเป็นอันดับที่ i ว่า s_i

เราจะพิจารณาเหตุการณ์ A_{ij} เพื่อให้เข้าใจได้ง่าย จะลองพิจารณา A_{1n} ก่อน นั่นคือเหตุการณ์ที่ข้อมูลที่มากที่สุด s_n และข้อมูลที่น้อยที่สุด s_1 จะถูกเปรียบเทียบกันในการทำงานของการจัดเรียงแบบควิก เมื่ออัลกอริทึมเริ่มทำงาน จะมีการสุ่มเลือกไพวอต สังเกตว่าในการหยิบครั้งแรกนี้เลย ถ้าหยิบไพวอตได้เป็น s_1 หรือ s_n ข้อมูลทั้งสองจะถูกเปรียบเทียบกันในขั้นตอนที่มีการแบ่งข้อมูลเป็นสองส่วนเลย ในทางกลับกันถ้าหยิบไพวอตได้เป็นข้อมูลตัวอื่น ๆ s_1 กับ s_n จะถูกจับแบ่งให้แยกกันอยู่คนละปัญหาย่อย สังเกตว่า เมื่อข้อมูลถูกแบ่งให้อยู่กันคนละปัญหาย่อยแล้ว จะไม่มีทางกลับมาเปรียบเทียบกันได้อีกเลย นั่นคือ ในกรณีหลังนี้ s_1 และ s_n จะไม่ถูกเปรียบเทียบกัน

ดังนั้น ความน่าจะเป็นที่เหตุการณ์ A_{1n} จะเกิดขึ้นคือ $\frac{2}{n}$ เนื่องจากมีไพวอตที่เป็นไปได้ n ตัว แต่ต้องหยิบได้ s_1 หรือ s_n เท่านั้น

เราจะพิจารณากรณีทั่วไป เพื่อความง่าย เราจะพิจารณาเฉพาะกรณีที่ $i < j$ ถ้าเราไล่การทำงานของอัลกอริทึม เรา

จะพบว่าข้อมูล s_i และ s_j จะอยู่ในปัญหาย่อยเดียวกันไปตลอด จนกว่าจะมีการหยิบไพวอตในเซต $s_i, s_{i+1}, s_{i+2}, \ldots, s_j$ เพราะว่าไพวอตอื่น ๆ ก็จะแบ่งข้อมูลและทำให้ s_i และ s_j อยู่ในปัญหาย่อยเดียวกันเสมอ ดังนั้นเหตุการณ์ที่มีผลต่อการเปรียบเทียบกันของ s_i และ s_j คือเหตุการณ์ที่มีการหยิบ $s_i, \ldots, s_j$ ตัวใดตัวหนึ่งเป็นไพวอต

เนื่องจากเราเลือกไพวอตแบบสุ่ม ข้อมูล $j-i+1$ ตัวเหล่านั้นล้วนมีความน่าจะเป็นเท่ากันที่จะเป็นไพวอตทั้งหมด ถ้าเราเลือก $s_{i+1}, \ldots, s_{j-1}$ เราจะได้ว่า s_i และ s_j จะไม่ถูกเปรียบเทียบกัน นั่นคือมีแค่ 2 กรณีเท่านั้นที่จะมีการเปรียบเทียบ s_i กับ s_j

ดังนั้น ความน่าจะเป็นที่ระหว่างการทำงาน s_i จะถูกเปรียบเทียบกับ s_j คือ

$$\Pr[A_{ij}] = \frac{2}{j-i+1}$$

จากนิยามของค่าคาดหวัง เราจะได้ว่า

$$\mathrm{E}[X_{ij}] = 0 \cdot \Pr[X_{ij}=0] + 1 \cdot \Pr[X_{ij}=1] = 1 \cdot \Pr[A_{ij}] = \frac{2}{j-i+1}$$

เมื่อเรานำไปแทนค่าในนิพจน์ตอนบน เราจะได้ว่าค่าคาดหวังของจำนวนครั้งการเปรียบเทียบคือ

$$\mathrm{E}\left[\sum_{i=1}^{n}\sum_{j=i+1}^{n} X_{ij}\right] = \sum_{i=1}^{n}\sum_{j=i+1}^{n}\mathrm{E}[X_{ij}] = \sum_{i=1}^{n}\sum_{j=i+1}^{n}\frac{2}{j-i+1} \le 2n \cdot \left(\frac{1}{1}+\frac{1}{2}+\cdots+\frac{1}{n-1}+\frac{1}{n}\right)$$

ซึ่งมีค่าเป็น $O(n \log n)$

1.3 การใช้การสุ่มในการสร้างฟังก์ชันแฮช

ในส่วนนี้เราจะวิเคราะห์การสร้างฟังก์ชันแฮชที่กล่าวถึงในส่วนที่ **??**

เราจะให้ m แทนจำนวนช่องในตาราง มีหมายเลขเป็น $0, 1, \ldots, m-1$ ให้ n แทนจำนวนกุญแจที่เป็นไปได้ทั้งหมด โดยกุญแจมีค่าในเซต $\{0, 1, \ldots, n-1\}$ ให้ p เป็นจำนวนเฉพาะที่มีค่าไม่น้อยกว่า n

เราจะสุ่มเลือกจำนวนเต็ม a จากเซต $\{1, 2, \ldots, p-1\}$ และ b จากเซต $\{0, 1, 2, \ldots, p-1\}$ จากนั้นฟังก์ชันแฮชที่เราใช้คือ

$$h(k) = ((a \cdot k + b) \bmod p) \bmod m$$

ฟังก์ชันแฮชที่ดีควรเป็นอย่างไร? จากที่เราได้เคยพิจารณาไปแล้ว เราพอสรุปได้คร่าว ๆ ว่าจะต้องเป็นฟังก์ชันที่กระจายข้อมูลไปให้ทั่วตารางแฮช หรือในอีกมุมหนึ่งก็คือเป็นฟังก์ชันที่มีลักษณะของการชนกันของข้อมูลเหมือนฟังก์ชันสุ่ม

▷ **คำถาม 1.1** สุ่มโยน
ถ้าเราโยนข้อมูลลงไปในตารางแฮชที่มี m ช่อง ช่องหนึ่งแบบสุ่ม โดยที่ทุกช่องมีความน่าจะเป็นที่จะมีข้อมูลโยนลงไปเท่ากัน ความน่าจะเป็นที่โยนข้อมูลสองตัวแล้วตกลงไปในช่องเดียวกันเป็นเท่าใด? ◁

สำหรับในกรณีของเรา ฟังก์ชันที่เราใช้นี้ เป็นผลลัพธ์จากการทดลองสุ่ม ที่หยิบเลือก a และ b มา เราต้องการพิสูจน์ว่า สำหรับกุญแจ x และ y ใด ๆ ที่ไม่เท่ากัน ความน่าจะเป็นที่ฟังก์ชันแฮชจะจับให้อยู่ในตารางช่องเดียวกันเป็น

$$\Pr[h(x) = h(y)] = \frac{1}{m}$$

เพื่อความสะดวก ต่อไปเราจะเขียน $[w]$ แทนเซต $\{0, 1, 2, \ldots, w-1\}$
ให้ฟังก์ชัน

$$g(k) = (a \cdot k + b) \bmod p$$

เราจะเขียนฟังก์ชัน h ใหม่ได้เป็น

$$h(k) = g(k) \bmod m$$

เนื่องจากฟังก์ชัน h ที่ได้มีการคำนวณสองส่วนคือการคำนวณฟังก์ชัน g และการนำผลลัพธ์ที่ได้ไปหารเอาเศษด้วย m เราจะแบ่งการวิเคราะห์เป็นสองส่วนเช่นเดียวกัน

กุญแจ x และ y จะถูก h พาไปอยู่ช่องเดียวกันได้ก็ต่อเมื่อ

$$g(x) \bmod m = g(y) \bmod m$$

เราจะนับคู่ของจำนวนเต็ม w, z ที่ $w \neq z$ ในเซต $[n]$ ที่ $w \bmod m = z \bmod m$ พิจารณาจำนวนเต็ม $w \in [n]$ ใด ๆ จำนวนเต็ม z ที่สอดคล้องกับเงื่อนไขดังกล่าวจะมีไม่เกิน $\lceil p/m \rceil - 1$ จำนวน ดังนั้นจำนวนคู่ที่สอดคล้องกับเงื่อนไขคือ $p(\lceil p/n \rceil - 1) \leq p(p-1)/m$

ในขั้นต่อไป เราจะใช้คุณสมบัติทางทฤษฎีจำนวน สำหรับจำนวนเต็มสี่จำนวน $w, x, y, z \in [n]$ ใด ๆ พิจารณาสมการทั้งสองนี้

$$(a \cdot x + b) \bmod p = w$$

$$(a \cdot y + b) \bmod p = z$$

เราสามารถพิสูจน์ได้ว่าถ้า p เป็นจำนวนเฉพาะ จะมีจำนวนเต็ม a และ b ที่อยู่ในเซต $[n]$ เพียงแค่คู่เดียวที่เป็นคำตอบของสมการทั้งคู่

จากที่ได้กล่าวมา เราจะสามารถสรุปได้ว่ามีคู่ของจำนวนเต็ม a และ b เพียงแค่ $p(p-1)/m$ คู่เท่านั้นที่ทำให้

$$h(x) = h(y)$$

สำหรับกุญแจ x และ y ที่ $x \neq y$

สังเกตว่าเราสามารถเลือก a ได้ $p-1$ แบบ และ b ได้อีก p รวมคู่ของ a และ b ที่เป็นไปได้คือ $p(p-1)$ คู่ ดังนั้นความน่าจะเป็นที่หยิบได้คู่ที่ทำให้ $h(x) = h(y)$ คือ

$$\frac{p(p-1)/m}{p(p-1)} = \frac{1}{m}$$

ตามต้องการ

กลุ่มของฟังก์ชันแฮชทั้งหมดที่เราสร้างได้นี้มีคุณสมบัติที่ความน่าจะเป็นที่กุญแจสองกุญแจใด ๆ ที่ไม่เท่ากัน จะถูกจับให้อยู่ในตารางช่องเดียวกันเท่ากับ $1/m$ เมื่อ m เป็นขนาดของตารางแฮะ เราจะเรียกเซตของฟังก์ชันแฮชใด ๆ ที่มีคุณสมบัติดังกล่าวว่าเป็นตระกูลแฮชสากล (universal hash family)