

C07 - Croissaria Number - Editorial by พี่ที่ออกโจทย์ง่ายที่สุด

สังเกตว่าขอบเขตของตัวแปร L, R มีค่าสูงมาก ๆ ดังนั้นการที่เราเช็คตัวเลขทุกตัวตั้งแต่ $[L, R]$ จึงเป็นไปได้เพราะจะใช้เวลานานเกินไป* ให้ $C(l, r)$ แทนจำนวนของตัวเลข Croissaria ในช่วง $[l, r]$ สังเกตว่า

$$C(L, R) = C(10, R) - C(10, L - 1)$$

จึงเพียงพอที่จะหาค่าของ $C(10, x)$ โดยที่ x เป็นจำนวนเต็มบวก

สมมติว่า x เป็นจำนวนเต็มบวกที่มี $n \geq 2$ หลัก จะได้ว่าสำหรับจำนวนเต็ม Croissaria ใด ๆ ที่มี k หลัก ซึ่ง $k < n$ เราสามารถให้เลขโดดซ้ายเป็นเลขใดก็ได้ (เพราะมีค่าไม่เกิน x แน่نون) สมมติว่าเป็น d จะได้ว่าเลขโดดใน $k - 1$ หลักที่เหลือจะมีค่าที่เป็นไปได้ทั้งหมด d ค่า $(0, 1, \dots, d - 1)$ ดังนั้นจำนวนของตัวเลข Croissaria ที่มี k หลัก และเลขโดดหน้าสุดคือ d จะมีทั้งหมด d^{k-1} ตัว

เนื่องจาก d สามารถเป็นได้ทั้งหมด 9 ค่า คือ $1, 2, \dots, 9$ จะให้จำนวนของตัวเลข Croissaria ทั้งหมดที่มี

จำนวนหลักน้อยกว่า n มีทั้งหมด $\sum_{k=2}^{n-1} \sum_{d=1}^9 d^{k-1}$ ตัว ซึ่งผลรวมดังกล่าวสามารถหาได้ด้วย Loop

ต่อมาเราจะพิจารณากลุ่มจำนวนเต็มที่มี n หลัก และมีค่าไม่เกิน x ซึ่งเราสามารถแบ่งกลุ่มของจำนวนเต็มทั้งหมดเป็น 3 กลุ่มย่อย ดังนี้ (ให้ D_i คือเลขโดดหลักที่ i ของ x ซึ่งนับจากทางซ้าย)

- กลุ่ม 1 : หลักแรกน้อยกว่า D_1
- กลุ่ม 2 : เลขโดดใน k หลักแรกตรงกับ x และเลขโดดในหลักที่ $k + 1$ น้อยกว่า D_{k+1}
สำหรับทุก $1 \leq k \leq n - 1$
- กลุ่ม 3 : เลขนั้นคือ x

โดยจำนวนของตัวเลข Croissaria ในแต่ละกลุ่มมีวิธีหาดังนี้

กลุ่มที่ 1 : ทำในทำนองเดียวกันกับกรณีก่อนหน้า จะได้เป็น $\sum_{d=1}^{D_1-1} d^{n-1}$

กลุ่มที่ 2 : สำหรับแต่ละ k ถ้ามีบางจำนวนเต็ม i ซึ่ง $2 \leq i \leq k$ และ $D_i \geq D_1$ แล้วทุกจำนวนเต็มที่มี k หลักแรกตรงกับ x จะไม่ใช่จำนวน Croissaria แต่ถ้าไม่มี i ที่สอดคล้อง จะได้ว่าเลขโดดในหลักที่ $k + 1$ จะต้องมีค่าน้อยกว่า $\min(D_1, D_{k+1})$ และเลขโดดหลักที่ $k + 2, k + 3, \dots, n$ จะไม่ต้องมีค่าไม่เกิน D_1

ได้ว่าจำนวนของตัวเลข Croissaria ในกรณีนี้มีค่าเป็น $\sum_k \min(D_1, D_{k+1}) \cdot D_1^{n-k-1}$

สังเกตว่าหากมี k ที่ทำให้ i ที่สอดคล้องกับเงื่อนไขดังกล่าว แล้วสำหรับทุกจำนวนเต็มที่มีอย่างน้อย k หลักแรกตรงกับ x จะไม่เป็น Croissaria Number การเขียนโปรแกรมในกรณีนี้สามารถใช้ Loop ได้ โดยหากพบ k ที่ทำให้ i ที่สอดคล้องกับเงื่อนไข ก็ให้จบการทำงานของ Loop ทันที

กลุ่มที่ 3 : สามารถเช็คได้โดยง่าย ขอละไว้เป็นแบบฝึกหัด

*หมายเหตุ : เหตุผลที่ไม่สามารถใช้ Loop ในการนับตรง ๆ เป็นเพราะว่าในกรณีที่แย่สุดโปรแกรมต้องรัน Loop ประมาณ 10^{18} ครั้ง ซึ่งเกินขีดจำกัดของคอมพิวเตอร์ทั่วไป

(หากอยากรู้รายละเอียดเพิ่มเติม ลองไปอ่านเรื่อง Time Complexity ของวิชา Data&Algo ได้ครับ)