



การแข่งขันคณิตศาสตร์และวิทยาศาสตร์ระหว่างโรงเรียนครั้งที่ 18: TUMSO 18<sup>th</sup>

วิชาคอมพิวเตอร์

เวลา 13:00 น. - 16:00 น.

รอบที่ 2

เฉลย

การแข่งขันคณิตศาสตร์และวิทยาศาสตร์ระหว่างโรงเรียนครั้งที่ 18  
(18<sup>th</sup> Triam Udom Mathematics and Science Olympiad)

วิชาคอมพิวเตอร์ รอบที่ 2

วันที่ 9 มกราคม 2563 เวลา 13:00 น. - 16:00 น.

ID โจทย์	ชื่อโจทย์	Time	Memory	คะแนนชุดทดสอบย่อย	รวม (คะแนน)
G-final-crisis	Final Crisis	1 s	256 MB	30 70	100
H-forest-resorts	Forest Resorts	1 s	256 MB	25 25 50	100
I-mathmath	math math	1 s	256 MB	15 35 50	100
J-isekai	Isekai No Hajine	1 s	256 MB	10 20 70	100
K-precious-treasure	สมบัติล้ำค่า	1 s	256 MB	10 25 65	100
L-autocomplete	Autocomplete	1 s	256 MB	100	100



การแข่งขันคณิตศาสตร์และวิทยาศาสตร์ระหว่างโรงเรียนครั้งที่ 18: TUMSO 18<sup>th</sup>

วิชาคอมพิวเตอร์

เวลา 13:00 น. - 16:00 น.

รอบที่ 2

## Final crisis (100 คะแนน)

1 second, 256 megabytes

ในข้อนี้สังเกตว่าการอ่านหนังสือของเทพเอิร์ธจะมีผลทำให้จำนวนหน้าของวิชาที่อ่านมากกว่านั้นน้อยที่สุดเท่าที่จะเป็นไปได้ ดังนั้นแต่ละคำถามในโจทย์ข้อนี้จึงกลายเป็นการ minimize maximum ของ prefix sum สองชุด โดยที่แต่ละชุดต้องมีสมาชิกรวมกันเท่ากับ  $k$

ในแต่ละคำถาม วิธีตรงไปตรงมาในการหาคำตอบคือการไล่หาจากจำนวนหนังสือที่อ่านในวิชาหนึ่ง เช่น หาก  $k = 3$  เราจะไล่ชีววิทยา 0 เล่ม ประวัติศาสตร์ 3 เล่ม, ชีววิทยา 1 เล่ม ประวัติศาสตร์ 2 เล่ม, ชีววิทยา 2 เล่ม ประวัติศาสตร์ 1 เล่ม และ ชีววิทยา 3 เล่ม ประวัติศาสตร์ 0 เล่ม แล้วคำนวณค่า minimum ในทุก ๆ เคส

วิธีนี้ใช้เวลา  $O(qn)$  และจะได้ 20 คะแนน

ในการปรับปรุง เราจะใช้เทคนิคซึ่งคล้าย ๆ กับ binary search ก่อนอื่นเราจะทำการ "เดา" จำนวนหนังสือในวิชา ๆ หนึ่งมาก่อน เรียกจำนวนนั้นว่า  $x$  (ซึ่งทำให้เรารู้จำนวนหนังสือในอีกวิชาหนึ่ง:  $k - x$ ) จากนั้นเราจะมาเทียบจำนวนหน้าของสองวิชาจากจำนวนเล่มที่เราเดา (สามารถทำได้ใน  $O(1)$  โดย Quicksort) สมมติว่าวิชาแรกมีจำนวนหน้าเยอะกว่า เราก็จะรู้ทันทีว่าถ้าเราเลือกอ่านหนังสือวิชาแรกมากกว่า  $x$  เล่ม คำตอบจะไม่มีทางดีขึ้น ทำให้เราสามารถทิ้งทุกค่าที่มากกว่า  $x$  ไปได้เลย

วิธีนี้ใช้เวลา  $O(q \log n)$  และจะได้ 100 คะแนน



การแข่งขันคณิตศาสตร์และวิทยาศาสตร์ระหว่างโรงเรียนครั้งที่ 18: TUMSO 18<sup>th</sup>

วิชาคอมพิวเตอร์

เวลา 13:00 น. - 16:00 น.

รอบที่ 2

## Forest Resorts (100 คะแนน)

1 second, 256 megabytes

สังเกตก่อนว่ารีสอร์ตนั้นควรจะถูกสร้างบนโหนดที่เป็น leaf (โหนดที่มีดีกรี 1) ของต้นไม้เท่านั้น เพราะหากสร้างที่อื่น เราสามารถย้ายลงมาสร้างที่ leaf เพื่อที่จะทำให้จำนวนถนนที่สร้างขึ้นนั้นมากกว่าเดิมได้

ในตอนเริ่มแรกนั้นให้เราจินตนาการว่าเราได้สร้างรีสอร์ตไว้บน leaf ทุก ๆ leaf แล้ว (ทำให้เราจำเป็นที่จะต้องสร้างถนนทุกเส้น) จากนั้นในการหาคำตอบสำหรับจำนวนรีสอร์ตที่น้อยลงมา เราจะเลือกแบบ greedy คือเลือกทำลายรีสอร์ตที่เมื่อทำลายแล้วจะทำให้ต้องลบถนนออกน้อยที่สุดจาก configuration ก่อนหน้า

ในการทำเช่นนี้ เราสามารถใช้ priority queue ในการตัดสินใจว่าจะเลือกทำลายรีสอร์ตที่โหนดไหน รวมกับการ update จำนวนถนนที่จะหายไปหากทำลายรีสอร์ตที่โหนดนั้น ๆ

เมื่อจบแล้ว  $ans[x]$  จะเก็บว่าหากจะสร้างรีสอร์ต  $x$  ที่ จะต้องลบ edge ออกจาก tree ทั้งหมดกี่เส้น

อัลกอริธึมนี้ใช้เวลา  $O(n \log n)$  และจะได้ 100 คะแนน



## Math Math (100 คะแนน)

1 second, 256 megabytes

คำตอบของข้อนี้สำหรับแต่ละค่า  $k$  คือ  $567k + 9$  วิธีแก้โจทย์ข้อนี้มีหลายวิธี อาทิเช่น ผู้เข้าแข่งขันอาจจะ implement BigNum ในการหา pattern ของคำตอบ แต่เราได้ทำการเขียนบทพิสูจน์ของสูตรนี้ให้อ่านได้เข้าใจ

กำหนดให้  $p$  เป็นจำนวนเฉพาะคี่และ  $a$  เป็นจำนวนเต็มที่  $(a, p) = 1$  สัญลักษณ์เลอฌ็องด์ร์ (Legendre symbol)  $\left(\frac{a}{p}\right)$  นิยามดังนี้

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{ถ้า } a \text{ เป็นส่วนตกค้างกำลังสองของ } p \\ -1 & \text{ถ้า } a \text{ ไม่เป็นส่วนตกค้างกำลังสองของ } p \end{cases}$$

จาก  $\left(\frac{10}{127}\right) = -1$  จะได้ว่า  $\frac{10^{63} + 1}{127}$  เป็นจำนวนเต็ม

ให้  $\frac{1}{127} = \frac{a_1}{10} + \frac{a_2}{100} + \dots + \frac{a_n}{10^n} + \dots$  เมื่อ  $a_i$  เป็นเลขโดด

จะได้ว่า  $\frac{10^{63}}{127} = k + \frac{a_{64}}{10} + \frac{a_{65}}{100} + \dots + \frac{a_{n+63}}{10^n} + \dots$  เมื่อ  $k$  เป็นจำนวนเต็มบวก

$$\text{ดังนั้น } \frac{10^{63} + 1}{127} - k = \frac{a_1 + a_{64}}{10} + \frac{a_2 + a_{65}}{100} + \dots + \frac{a_n + a_{n+63}}{10^n} + \dots$$

ให้  $\text{RHS} = m$  จากการ bound จะได้ว่า  $0 < m < 2$  ดังนั้น  $m = 1$  จะได้  $a_i + a_{i+63} = 9$  สำหรับ  $i = 1, 2, \dots$

จาก  $m^{\phi(n)} \equiv 1 \pmod{n}$  เมื่อ  $(m, n) = 1$  และ  $n$  เป็นจำนวนเฉพาะ และ  $\phi(127) = 126$  จะได้ว่า  $10^{126} \equiv 1 \pmod{127}$  ซึ่งหมายความว่า  $127 \mid 10^{126k} - 1$

$$\text{ให้ } \frac{10^{126k} - 1}{127} = \overline{a_1 a_2 a_3 \dots a_n}$$

$$\text{เราจะได้ว่า } \frac{10^{126k+3} - 1000}{127} = \overline{a_1 a_2 a_3 \dots a_n 000}$$

$$\text{และ } \frac{10^{126k+3} + 143}{127} = \overline{a_1 a_2 a_3 \dots a_n 009}$$

$$\text{จาก } \frac{1}{127} = \frac{a_1}{10} + \frac{a_2}{100} + \dots + \frac{a_n}{10^n} + \dots \text{ เมื่อ } a_i \text{ เป็นเลขโดด}$$

$$\text{จาก } \frac{10^{126k}}{127} = \overline{a_1 a_2 \dots a_{126k}} + \frac{a_{126k+1}}{10} + \dots$$

$$\frac{10^{126k} - 1}{127} = \overline{a_1 a_2 \dots a_{126k}} + \sum_{i=1}^{\infty} \frac{a_{i+126k} \cdot a_i}{10^i}$$

$$\text{เนื่องจาก } \sum_{i=1}^{\infty} \frac{a_{i+126k} \cdot a_i}{10^i} \text{ เป็นจำนวนเต็ม และ } -1 < \sum_{i=1}^{\infty} \frac{a_{i+126k} \cdot a_i}{10^i} < 1$$



การแข่งขันคณิตศาสตร์และวิทยาศาสตร์ระหว่างโรงเรียนครั้งที่ 18: TUMSO 18<sup>th</sup>

วิชาคอมพิวเตอร์

เวลา 13:00 น. - 16:00 น.

รอบที่ 2

$$\text{ดังนั้น } \sum_{i=1}^{\infty} \frac{a_{i+126k} \cdot a_i}{10^i} = 0$$

$$\text{ดังนั้นผลรวมเลขโดด } \frac{10^{126k} - 1}{127} = a_1 + a_2 + \dots + a_{126k}$$

$$\text{จาก } a_i + a_{i+63} = 9 \text{ จะได้ว่า } a_1 + a_2 + \dots + a_{126k} = 9 \cdot 63 \cdot k$$

$$\text{ดังนั้นผลรวมเลขโดด } \frac{10^{126k+3} + 143}{127} = 9 \cdot 63 \cdot k + 9 = 567k + 9$$



การแข่งขันคณิตศาสตร์และวิทยาศาสตร์ระหว่างโรงเรียนครั้งที่ 18: TUMSO 18<sup>th</sup>

วิชาคอมพิวเตอร์

เวลา 13:00 น. - 16:00 น.

รอบที่ 2

## Isekai No Hajime (100 คะแนน)

1 second, 256 megabytes

วิธีการแก้โจทย์ข้อนี้ที่ง่ายที่สุดคือการ simulate การพังทลายของหุบเขาและความสูงที่เปลี่ยนไป เพื่อหา damage ที่มากที่สุดที่เราสามารถทำได้ต่อராชาปีศาจในแต่ละวัน

ข้อสังเกตสำคัญ 3 ข้อในการแก้ปัญหาคือ

1. สมมติปีศาจยิง laser ระดับที่  $x$  แล้ว ถ้าหากในคืนหลังจากนั้นปีศาจยิง laser ระดับที่  $y$  โดยที่  $y \geq x$  จะไม่เกิดอะไรขึ้น
2. หากเราไม่พิจารณา laser ที่ "useless" จากข้อสังเกตข้อแรกแล้ว ช่วงของหุบเขาใด ๆ ที่โดนทำลายมาก่อนหน้านี้ จะโดนทำลายต่อไปในอนาคตเสมอ
3. เราสามารถเปรียบเทียบพื้นที่ที่ช่วง ๆ หนึ่งถูกทำลาย เหมือนการ "เชื่อม" พื้นที่ข้าง ๆ ทั้งสองด้านเข้าด้วยกัน (ตราบใดที่ช่วง ๆ นั้น ถูกทำลายแล้ว)

ข้อสังเกตทั้ง 3 ข้อนี้สามารถนำไปสู่อัลกอริธึมที่ใช้แก้โจทย์ปัญหานี้ได้ โดยในอัลกอริธึมของเรา เราจะทำการสร้าง event point ของการที่พื้นที่หนึ่งถูกทำลาย ยุบ และถูกเชื่อมรวมกัน เป็นการ merge สองพื้นที่เข้าด้วยกัน โดยใช้ disjoint set ในการเก็บข้อมูลนั้น ๆ และสำหรับแต่ละคืนเราจะคำนวณผลรวม damage ที่เราสามารถทำได้ต่อராชาปีศาจ เมื่อทำเช่นนั้นแล้ว การตอบปัญหาแต่ละข้อก็เป็นการ binary search หาวน ๆ แรกที่เราสามารถฆ่าராชาปีศาจได้สำเร็จนั่นเอง

วิธีนี้ใช้เวลา  $O((Q + N) \log N)$  และจะได้ 100 คะแนน



## สมบัติลําค่า (100 คะแนน)

1 second, 256 megabytes

ให้  $f(n)$  เป็นจำนวนวิธีการปูกระเบื้องเป็นทางยาวขนาด  $2 * n$  และให้  $g(n)$  เป็นจำนวนวิธีการปูกระเบื้องเป็นทางยาวขนาด  $2 * n$  โดยที่ทางยาวนี้ต้องนำไปต่อกระเบื้องสีดำ 1 ชิ้นที่ปลายข้างหนึ่ง และต้องเป็นไปตามกฎทุกอย่าง

เราจะให้  $f(0) = g(0) = 1$  (การที่ไม่ปูอะไรเลย ถือเป็นวิธีการปูกระเบื้องวิธีหนึ่ง) จากนั้นเราจะความสัมพันธ์เวียนเกิดให้  $f(n)$  และ  $g(n)$

เรามาศึกษา  $f(n)$  ก่อน ให้จินตนาการทางยาวขนาด  $2 * n$  ที่วางตามแนวนอนอยู่ การวางกระเบื้อง 2 ชิ้นทางด้านขวาจะเป็นไปได้ 3 กรณี

1. ปูกระเบื้องขาวทั้ง 2 ชิ้น
2. ปูกระเบื้องขาว 1 ชิ้น ดำ 2 ชิ้น
3. ปูกระเบื้องดำทั้ง 2 ชิ้น (ซึ่งเป็นไปไม่ได้ เพราะจะผิดกฎ)

หากเราปูกระเบื้องในวิธีแรกนั้น วิธีในการปูกระเบื้องที่เหลือ  $n - 1$  หลักจะเท่ากับ  $f(n - 1)$  เพราะการปูกระเบื้องสีขาวสองชิ้นจะไม่ทำให้การปูกระเบื้องที่เหลือมีสิทธิผิดกฎเลย สำหรับวิธีที่สองวิธีการปูกระเบื้องจะเท่ากับ  $g(n - 1)$  ตามนิยามที่เราได้ให้ไปก่อนหน้านี้ และเนื่องจากเราสามารถปูกระเบื้องตามวิธีที่ 2 ได้ 2 วิธี จะได้ว่า

$$f(n) = f(n - 1) + 2g(n - 1)$$

สำหรับ  $g(n)$  นั้น เราต้องปูกระเบื้องโดยคิดว่ามีกระเบื้องสีดำวางไว้อยู่แล้ว สำหรับกระเบื้อง 2 ชิ้นที่ปลายนั้น 1 ชิ้นที่ติดกับกระเบื้องสีดำจะต้องเป็นสีขาว (มิเช่นนั้นจะผิดกฎ) จากนั้นจะเหลือ 2 กรณี

1. วางกระเบื้องสีดำ
2. วางกระเบื้องสีขาว

ในกรณีแรกนั้นวิธีการปู  $n - 1$  หลักที่เหลือจะเท่ากับ  $g(n - 1)$  และในกรณีที่สองวิธีการปูจะเท่ากับ  $f(n - 1)$  ดังนั้น

$$g(n) = f(n - 1) + g(n - 1)$$

และคำตอบของเราจะเท่ากับ  $f(n)$  หากเราคำนวณ  $f(n)$  ไว้ก่อนตอบคำถาม วิธีนี้จะใช้เวลา  $f(N + t)$  เมื่อ  $N$  คือ  $n$  ที่มากที่สุด ในแต่ละคำถาม วิธีนี้จะได้ 35 คะแนน

เพื่อที่จะทำให้เร็วขึ้น เราสามารถเขียนสมการทั้งสองสมการข้างต้นได้ในรูปแบบต่อไปนี้

$$\begin{bmatrix} f(n) \\ g(n) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f(n - 1) \\ g(n - 1) \end{bmatrix}$$



การแข่งขันคณิตศาสตร์และวิทยาศาสตร์ระหว่างโรงเรียนครั้งที่ 18: TUMSO 18<sup>th</sup>

วิชาคอมพิวเตอร์

เวลา 13:00 น. - 16:00 น.

รอบที่ 2

จะได้ว่า

$$\begin{bmatrix} f(n) \\ g(n) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}^n \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

จากนี้จะเห็นได้ชัดว่า ปัญหาของเราคือการยกกำลัง matrix ให้เร็วกว่า  $O(n)$  ซึ่งจริงๆ แล้วสามารถทำได้ โดยใช้เทคนิค exponentiation by squaring ซึ่งทำการยกกำลัง matrix ได้ใน  $O(\log n)$

วิธีนี้จะใช้เวลา  $O(\log N + t)$  และจะได้ 100 คะแนน





การแข่งขันคณิตศาสตร์และวิทยาศาสตร์ระหว่างโรงเรียนครั้งที่ 18: TUMSO 18<sup>th</sup>  
วิชาคอมพิวเตอร์ เวลา 13:00 น. - 16:00 น.  
รอบที่ 2

---

## Autocomplete (100 คะแนน)

1 second, 256 megabytes

การแก้ปัญหานี้จะสามารถแยกออกได้เป็นสองส่วนคือ

1. Scanning: เปลี่ยนจาก text/code เป็นลำดับของ token
2. Parsing: เปลี่ยนลำดับของ token เป็น parse tree

เทคนิคที่สามารถใช้ได้ในการ scan และการ parse มีดังต่อไปนี้

1. token ใด ๆ คือหน่วยที่เกิดจากการต่อตัวอักษรบางส่วนเข้าด้วยกัน
2. เราไม่จำเป็นต้องสนใจเครื่องหมาย + หรือ - และเราสามารถยุบตัวเลขให้กลายเป็นเลขเดียวกันได้เลย เช่น 0
3. เราสามารถแยกชนิดของ token ออกได้ง่ายเลย คือ 0-9 เป็นตัวเลข a-z เป็น identifier และ A-Z เป็น keyword

จากนั้นให้เราพิจารณาตาม สิ่งที่สามารถแทน ได้โดยใช้ parse tree และรายชื่อของ identifier และ keyword ทั้งหมด