เฉลย

แบ่งเขตเลือกตั้ง (100 คะแนน)

3 seconds, 128 megabytes

ในข้อนี้ โจทย์สรุปคือ จะมีกราฟมาให้ และให้ตัดเส้นเชื่อมหลายๆครั้ง โดยที่แต่ละครั้งจะต้องคำนวณใหม่เรื่อยๆ โดยมีสูตรการคำนวณให้ในโจทย์

Subtask 1

วิธีการแก้ปัญหา ชุดทดสอบที่ 1 คือ สร้างกราฟใหม่ทุกครั้ง และทำ Graph search (DFS หรือ BFS ก็ได้) เพื่อหาว่าพื้นที่ส่วนไหนเชื่อมกันบ้าง เทียบเป็น connected component ทีละ component แล้วนำคะแนน $K_{i,j}$ ของแต่ละ i ใน component เดียวกัน มารวมกัน เป็น S_j หลังจากนั้นตรวจเช็คอีกรอบว่า ในแต่ละ j ตัวไหนทำให้ S_j มีค่ามากที่สุด ถ้ามีหลายตัว ตอบ -1 แล้วนำคำตอบนี้ ไปบวกรวมไว้ ทุกๆ component แล้วตรวจเช็คตัวมากสุดอีกรอบ หากมีหลายตัว ตอบ -1

Time Complexity: $\mathcal{O}((N+M)PQ)$

Subtask 2

วิธีการแก้ปัญหา ชุดทดสอบที่ 2 คือ มองย้อนกลับ แทนที่จะลบทีละเส้นเชื่อม มันจะเสียเวลา สังเกตว่าหากมองว่า เพิ่มทีละเส้นเชื่อม จะทำให้โจทย์ง่ายลง โดยอย่างแรกเราสร้างกราฟเริ่มต้นจากเส้นเชื่อมทั้งหมดเอาไปลบกับเส้นเชื่อมแต่ละเส้นที่ต้องการลบ จะได้กราฟที่มีทั้งหมด $M-Q$ เส้นเชื่อม เป็นกราฟเริ่มต้น เมื่อสร้างกราฟเสร็จก็ทำ Graph search เช่นเดียวกับวิธีการแก้ปัญหาชุดทดสอบที่ 1 เมื่อได้คำตอบแล้ว เก็บคำตอบไว้กับแต่ละ connected component

เมื่อต้องการเพิ่มเส้นเชื่อม จึงใช้โครงสร้างข้อมูล Disjoint-set data structure (DSU) เพื่อทำให้ union ได้ในเวลาอันสั้น โดยแต่ละครั้งที่ทำการ union 2 component เข้าด้วยกัน จะต้องลบข้อมูลเก่าที่เก็บเอาไว้ ของแต่ละ component หลังจากนั้นค่อยบวกเพิ่มกลับเป็นข้อมูลของ component ใหม่ ซึ่งการลบข้อมูล/เพิ่มข้อมูล ทำได้ใน $\mathcal{O}(P)$ และการ union ทำได้ใน $\mathcal{O}(\alpha N)$ จึงได้ว่าเวลาที่ใช้ในการตอบคำถามแต่ละคำถามคือ $\mathcal{O}(P\alpha(N))$

Time Complexity: $\mathcal{O}((N+M)+PQ\alpha(N))$

กดระเบิด (100 คะแนน)

3 seconds, 512 megabytes

โจทย์สรุปคือ มีจุดอยู่ทั้งหมด N จุด ให้หาว่าต้องเลือกอย่างน้อยกี่จุด ถึงจะทำให้พื้นที่ครอบคลุมต่อเนื่องคลุมครบทุกจุด เพื่อความสะดวกจะให้ M แทนขนาดของระบบพิกัด (กล่าวคือ หาก $1 \leq x_i, y_i \leq 500$ จะกล่าวว่า $M = 500$ และหาก $1 \leq x_i, y_i \leq 250$ จะกล่าวว่า $M = 250$)

Subtask 1

วิธีการแก้ปัญหาชุดทดสอบที่ 1 คือ Brute-force: ลองเลือกทุกรูปแบบที่เป็นไปได้ แล้วตรวจสอบว่า พื้นที่จะครอบคลุมต่อเนื่องครบทุกจุดหรือไม่ แล้วเช็คคำตอบที่ใช้จำนวนจุดเริ่มต้นน้อยสุด เนื่องจากการเลือกทุกรูปแบบที่เป็นไปได้มีทั้งหมด $\mathcal{O}(2^N)$ แบบ และในแต่ละแบบ จะต้องไล่จับคู่ตรวจสอบว่าครอบคลุมกันหรือไม่ หนึ่งในวิธีที่ทำได้ง่ายคือ Transitive Closure ซึ่งทำได้ใน $\mathcal{O}(N^3)$ หรือจะจับคู่แล้วไล่ไปเรื่อยๆ N รอบ ก็ใช้เวลา $\mathcal{O}(N^3)$ เช่นกัน

Time Complexity: $\mathcal{O}(2^N N^3)$

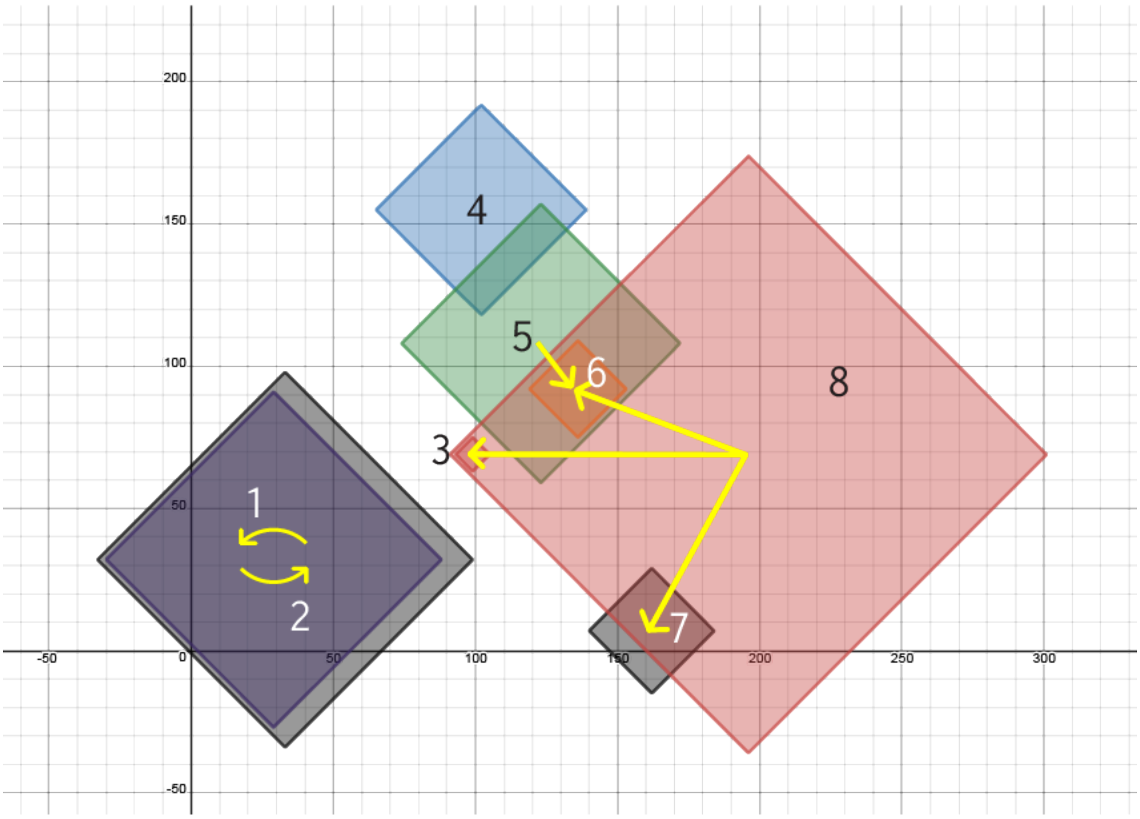
Subtask 2

วิธีการแก้ปัญหาชุดทดสอบที่ 2 คือ Breadth-First Search (BFS): สำหรับแต่ละจุด เนื่องจาก d_i ไม่เกิน 5 แสดงว่าพื้นที่ที่ครอบคลุมจะมีไม่เกิน $\mathcal{O}(d_i^2)$ ช่อง (เมื่อคำนวณจริงๆแล้วจะได้ 61 ช่อง ซึ่งถือว่าไม่มาก) จึงสามารถไล่ช่องที่แต่ละจุดครอบคลุมได้ จากนั้นสร้างเส้นเชื่อมแบบมีทิศทาง จากแต่ละจุดไปยังพื้นที่ที่ครอบคลุม จะได้เป็นกราฟแบบมีทิศทาง (ตัวอย่างที่ 2) ดังรูปที่ 1

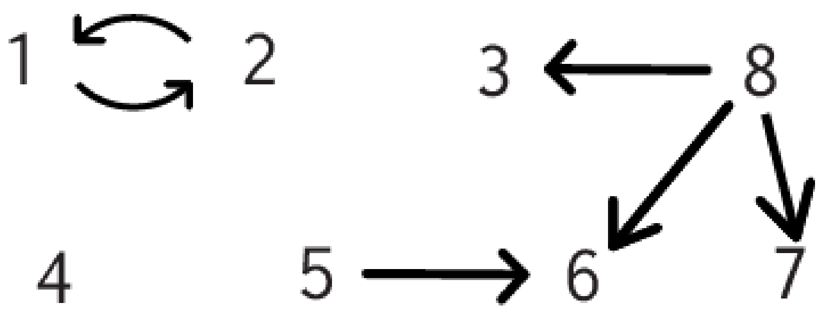
หลังจากนั้นสามารถทำการหาจุดที่จำเป็นจะต้องเลือก โดยเริ่มจากการเรียงลำดับแต่ละจุดตาม Topological Order (เนื่องจากกราฟไม่จำเป็นจะต้องเป็น DAG อาจจะมี cycle ได้ จึงไม่สามารถรับประกัน topological order ได้) แต่ในที่นี้ สามารถพิสูจน์ว่า ลำดับที่เสมือนเป็น topological order (นั่นคือสำหรับแต่ละจุด u ทุกๆจุด v ที่มีเส้นเชื่อมเข้าหา u จะได้ว่า $rank(v) \leq rank(u)$ นั่นคือจะต้องมาก่อน หรือมีลำดับเดียวกันก็ได้) จะเป็นลำดับที่เหมาะสมสำหรับการเลือกจุดเริ่มต้น

หลังจากนี้สามารถไล่แต่ละจุดตามลำดับดังกล่าว หากจุดปัจจุบันเคยผ่านมาแล้ว จะถือได้ว่าจุดปัจจุบันไม่จำเป็นจะต้องเลือก (เพราะจะมีจุดก่อนหน้าในลำดับที่เมื่อเลือกแล้วจะครอบคลุมจุดปัจจุบัน) แต่หากจุดปัจจุบันไม่เคยผ่าน แสดงว่าจำเป็นต้องเลือก เพราะหลังจากการ search ตามลำดับที่เรียงไว้แล้ว ตัวก่อนหน้าไม่มีตัวใดเลยที่ครอบคลุมจุดปัจจุบัน จึงได้ว่าจุดปัจจุบันไม่มีตัวครอบคลุมเลย จึงจำเป็นต้องเลือก เมื่อเลือกตามขั้นตอนดังกล่าว จะได้เซตของจุดที่ต้องเลือก เพื่อให้ครอบคลุมทั้งหมดทุกจุด โดยมีคำตอบน้อยที่สุด ตามที่โจทย์ต้องการ

Time Complexity: $\mathcal{O}(N + \sum\limits_{i=1}^{N} d_i^2)$



รูปที่ 1: รูปแสดงการสร้างเส้นเชื่อม



รูปที่ 2: รูปแสดงกราฟจากการลดรูป

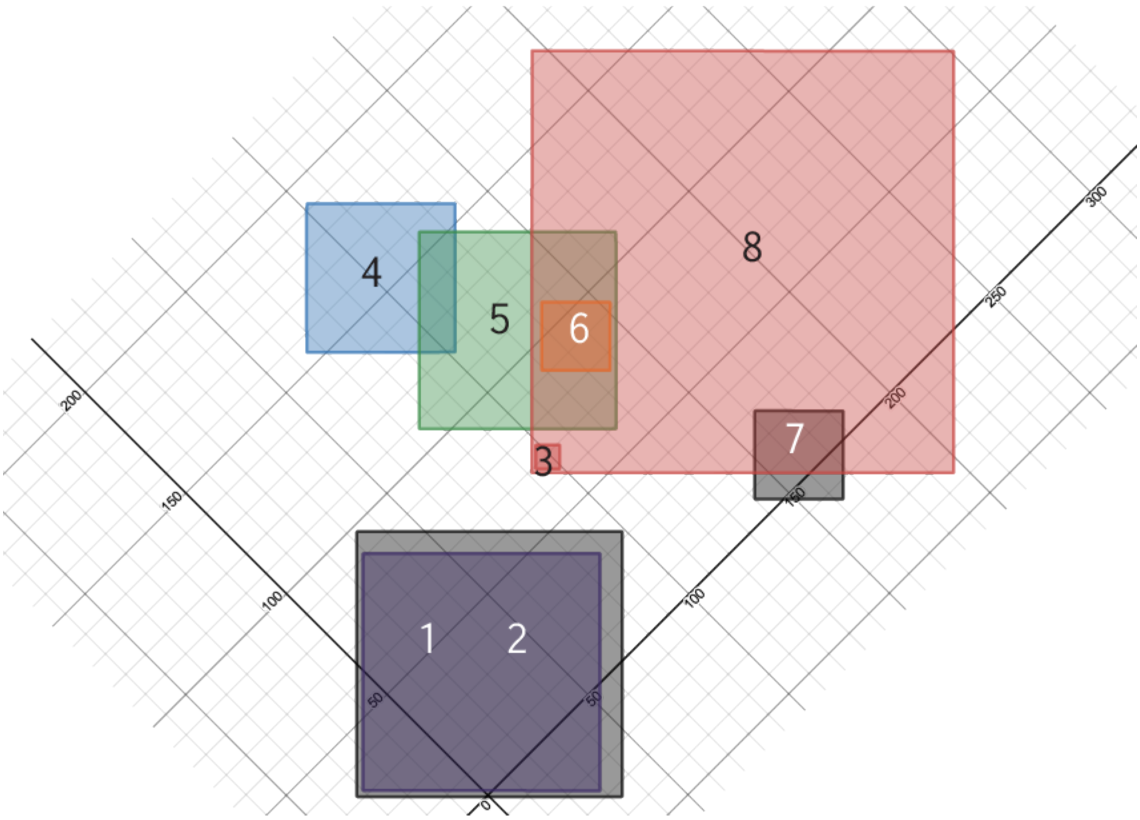
Subtask 3

วิธีการแก้ปัญหาชุดทดสอบที่ 3 คือ จับคู่จุดที่สามารถไปหากันได้ใน $\mathcal{O}(N^2)$ และทำกราฟแบบทิศทาง แล้วใช้วิธีเดียวกับปัญหาชุดทดสอบที่ 2 ซึ่งจะใช้เวลาในขั้นตอนนี้ $\mathcal{O}(N)$

Time Complexity: $\mathcal{O}(N^2)$

Subtask 4

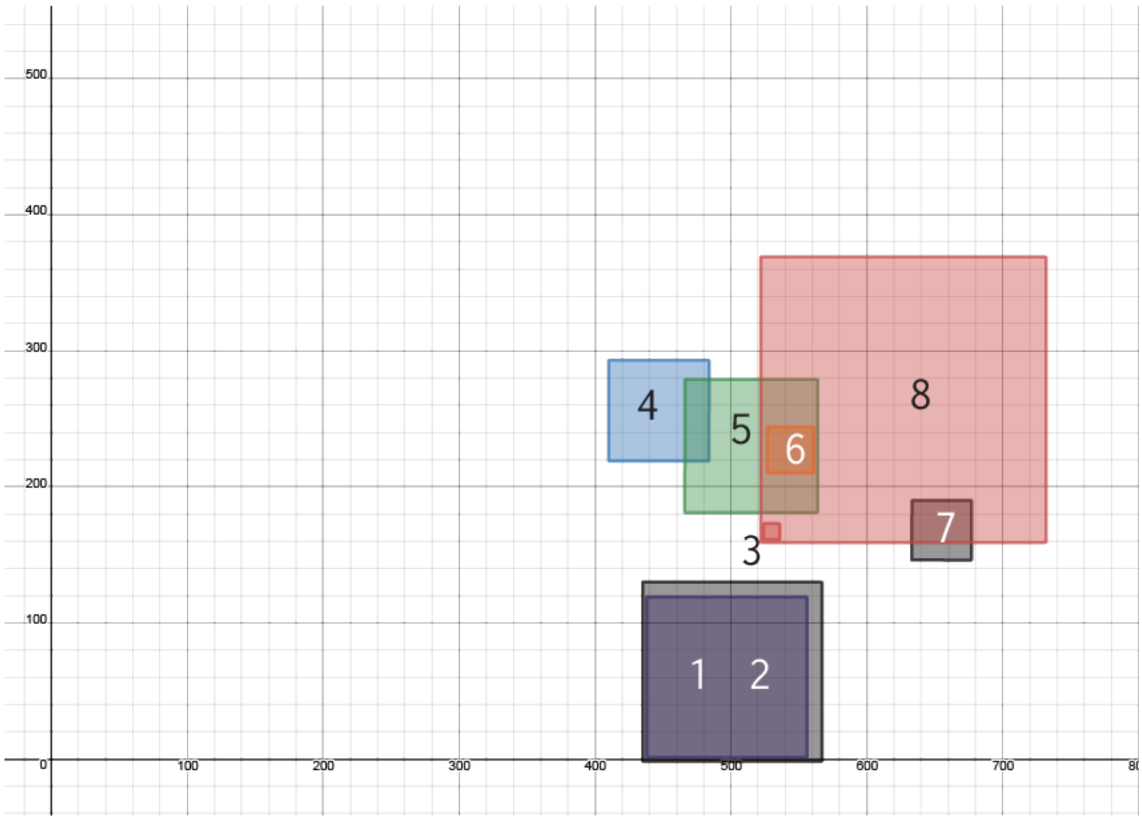
วิธีการแก้ปัญหาชุดทดสอบที่ 4 คือ หมุนตารางและพิกัดที่ให้มาตั้งแต่ต้น จะเปลี่ยนปัญหาจากสี่เหลี่ยมจัตุรัสที่ทำมุม 45 องศากับแกนเป็นสี่เหลี่ยมจัตุรัสที่แต่ละด้านขนานกับแกน โดยสามารถหมุนตารางจะทำให้กราฟเป็นดังรูปที่ 3



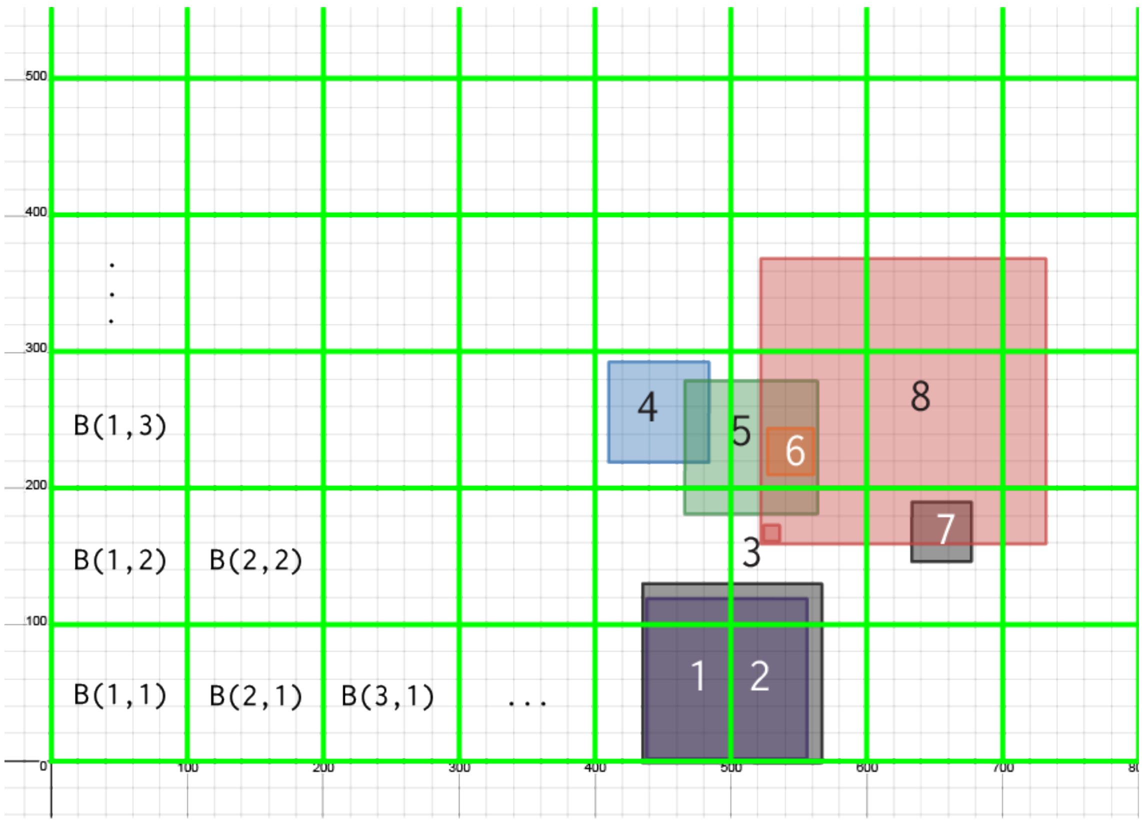
รูปที่ 3: รูปแสดงการหมุนระบบพิกัด

หลังจากนั้นทำการแปลงพิกัด (x, y) เป็น $(x - y + M, x + y - 1)$ ได้กราฟดังรูปที่ 4 ซึ่งสามารถพิสูจน์ได้ว่าพิกัดหลังแปลงแล้วจะมีค่าระหว่าง 1 ถึง $2M$ และสำหรับสี่เหลี่ยมที่เลยระบบพิกัดใหม่ ให้ถือว่าพื้นที่นอกอาณาเขต 1 ถึง $2M$ นั้นไม่มีความหมาย เพราะไม่มีทางครอบคลุมจุดใดๆได้ เนื่องจากไม่มีจุดใดอยู่นอกอาณาเขต

หลังจากนั้นสังเกตว่าเราสามารถแบ่งระบบพิกัดออกเป็นตารางกริดใหญ่ โดยแต่ละช่องมีขนาด $k \times k$ และสร้างจุดยอดในกราฟแบบมีทิศทางเพิ่มสำหรับแต่ละช่อง เพื่อป้องกันความสับสนเราจะเรียกแต่ละช่องว่าบล็อก ดังรูปต่อไปนี้ (เพื่อความสะดวก สำหรับตัวอย่างในรูปภาพเหล่านี้ จะให้ $k = 100$) โดยตารางกริดใหญ่ จะแทนด้วยเส้นสีเขียว และแต่ละช่องในตาราง นับเป็นแต่ละบล็อก ดังรูปที่ 5

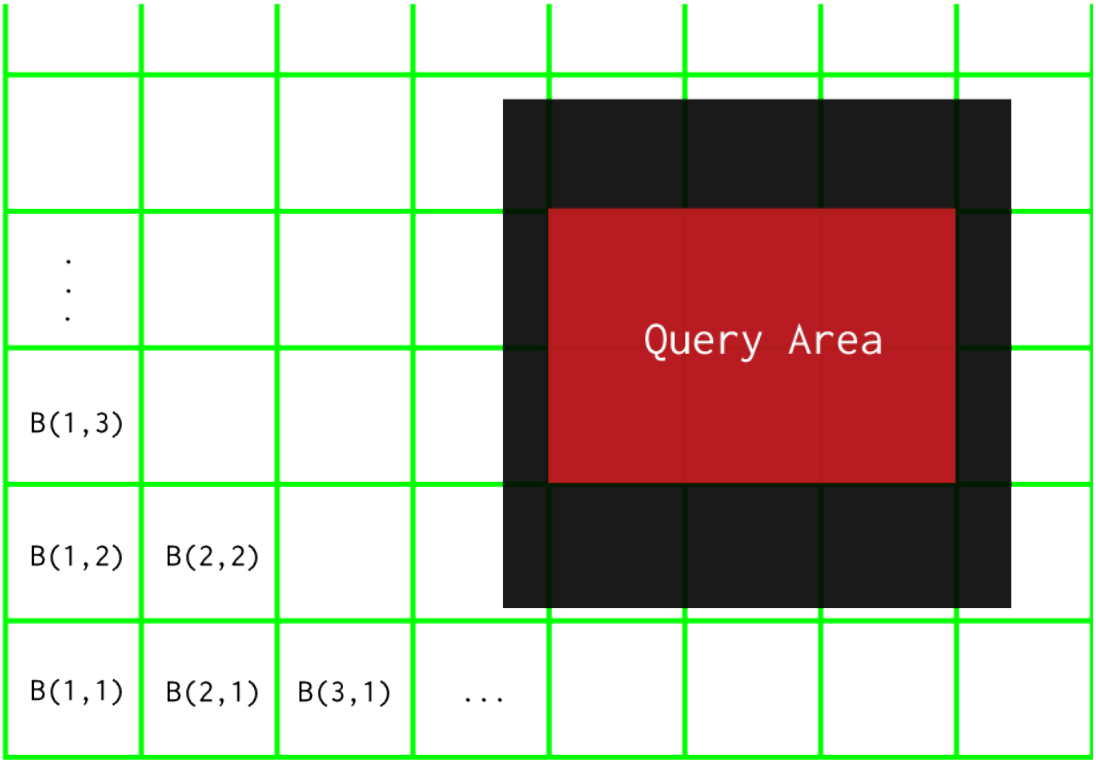


รูปที่ 4: รูปแสดงระบบพิกัดใหม่



รูปที่ 5: รูปแสดงการแบ่งบล็อกโดยใช้ $k = 100$

โดยในแต่ละบล็อก เราจะสร้างจุดเพิ่มเติมลงในกราฟระบุทิศทาง โดยแต่ละบล็อกจะมีเส้นเชื่อมชี้ไปยังจุดทุดจุดที่อยู่ภายในอาณาเขตของบล็อกนั้น ต่อมา เราจะไล่หาจุดทั้งหมดที่จุดปัจจุบันครอบคลุม ซึ่งโดยปกติจะใช้เวลา $\mathcal{O}(d_i^2)$ แต่เมื่อแบ่งบล็อกแล้ว เราจะสร้างรูปสี่เหลี่ยมจัตุรัสขนาด $d_i \times d_i$ บนกลุ่มบล็อก (พิจารณารูปที่ 6 แทนด้วยสี่เหลี่ยมจัตุรัสที่เป็นผลรวมของสีแดงและสีดำ) จะได้ว่ามีกลุ่มบล็อกอยู่ภายในอาณาบริเวณที่ใช้ ขนาดอย่างมาก $\mathcal{O}((\frac{M}{k})^2)$ บล็อก (พิจารณารูปที่ 6 แทนด้วยสีแดง) แต่ยังมีพื้นที่บางส่วนที่ไม่อยู่ภายในกลุ่มบล็อก (พิจารณารูปที่ 6 แทนด้วยสีดำ) แต่อยู่ภายในอาณาบริเวณที่ใช้ ซึ่งจะต้องเป็นขอบนอกของบริเวณกลุ่มบล็อกเท่านั้น



รูปที่ 6: รูปแสดงการถามหาจุดภายในขอบเขตของสี่เหลี่ยมสีดำ

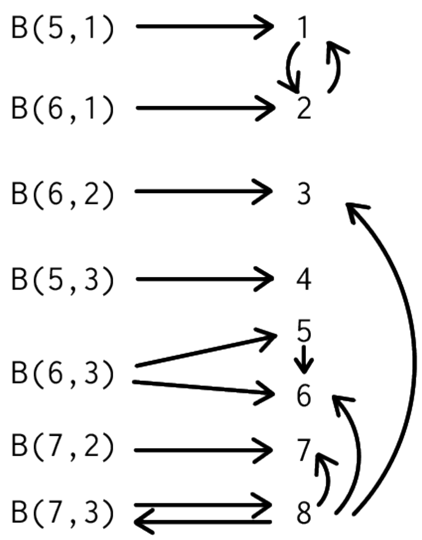
ซึ่งเนื่องจากกลุ่มบล็อกเป็นสี่เหลี่ยมจัตุรัสหรือผืนผ้าเท่านั้น จะได้ว่าบริเวณขอบนอกมีทั้งหมด 4 ด้าน คือไม่เกิน $4(\frac{M}{k}) - 4$ บล็อก เนื่องจากแต่ละบล็อกมีขนาด k^2 จึงใช้เวลาในกรณีนี้ $\mathcal{O}((\frac{M}{k})k^2)$ ทำให้เวลารวมเป็น

$\mathcal{O}((\frac{M}{k})^2 + \frac{M}{k}k^2) = \mathcal{O}((\frac{M}{k})^2 + Mk)$

เลือก $k = \sqrt{M}$ จะได้เวลา $\mathcal{O}(\frac{M}{\sqrt{M}} + M\sqrt{M})$

ต่อมาหลังจากจัดบล็อกแล้ว สร้างกราฟเติมแต่ง ดังรูปด้านล่าง แล้วทำตามวิธีเดียวกับปัญหาชุดทดสอบที่ 2 (สำหรับรูปภาพนี้ยังคงให้ $k = 100$ เพื่อความสะดวก แต่ในการใช้งานจริงจะให้ $k = \sqrt{M}$) พิจารณารูปที่ 7

Time Complexity: $\mathcal{O}(NM\sqrt{M})$



รูปที่ 7: รูปแสดงกราฟเติมแต่ง

Subtask 5

วิธีการแก้ปัญหาชุดทดสอบที่ 5 คือ ทำเช่นเดียวกับชุดทดสอบที่ 4 แต่ตอนเลือก k จะมีวิธีที่ดีกว่า $\sqrt{M}$ คือ $M^{\frac{1}{3}}$ เพราะว่าเมื่อพิจารณานิพจน์แสดงเวลา $\mathcal{O}((\frac{M}{k})^2 + Mk)$ เมื่อแทน $k = M^{\frac{1}{3}}$ จะได้เป็น $\mathcal{O}(\frac{M}{M^{\frac{1}{3}}}^2 + M(M^{\frac{1}{3}})$ จะได้เป็น $\mathcal{O}(NM^{\frac{4}{3}})$

Time Complexity: $\mathcal{O}(NM^{\frac{4}{3}})$

ผลรวมเฉพาะ (100 คะแนน)

0.5 seconds, 8 megabytes

โจทย์สรุปคือ ให้หาจำนวนเฉพาะ A, B ที่ $A + B = N$ และ $A \leq B$ ที่ A น้อยที่สุดเท่าที่เป็นไปได้

Subtask 1

จำเอาไว้ก่อนเลยว่า ถ้า N เป็น 1, 2, 3, ..., 10 ต้องตอบอะไรบ้าง ตามลำดับ

Time Complexity: $\mathcal{O}(1)$

Subtask 2

ไล่หา A ทั้งหมดตั้งแต่ 1 ถึง N และไล่ B ทั้งหมดตั้งแต่ 1 ถึง N และตรวจสอบว่า A เป็นจำนวนเฉพาะหรือไม่ และ B เป็นจำนวนเฉพาะหรือไม่ โดยสามารถตรวจสอบไว้ล่วงหน้าภายใน $\mathcal{O}(N^2)$ แล้วค่อยตรวจสอบว่า $A + B = N$ หรือไม่

Time Complexity: $\mathcal{O}(N^2)$

Subtask 3

ไล่หา A ทั้งหมดตั้งแต่ 1 ถึง N เนื่องจาก $B = N - A$ เสมอจึงไม่จำเป็นต้องไล่ B ที่เหลือก็ทดสอบว่า B เป็นจำนวนเฉพาะหรือไม่ โดยใช้ Sieve of Eratostenes เก็บไว้ก่อน โดยในที่นี้อาจสร้าง Sieve of Eratostenes ใน $\mathcal{O}(N \log N)$

Time Complexity: $\mathcal{O}(N \log N)$

Subtask 4

วิธีเดียวกับชุดทดสอบที่ 3 แต่การเก็บ Sieve of Eratostenes โดยใช้ Boolean Array จะใช้หน่วยความจำเกินกว่าที่กำหนด จึงต้องใช้ bitset และต้องปรับการเขียน Sieve of Eratostenes ให้ดียิ่งขึ้น โดย Sieve จะใช้ $\mathcal{O}(N \log \log N)$

Time Complexity: $\mathcal{O}(N \log \log N)$

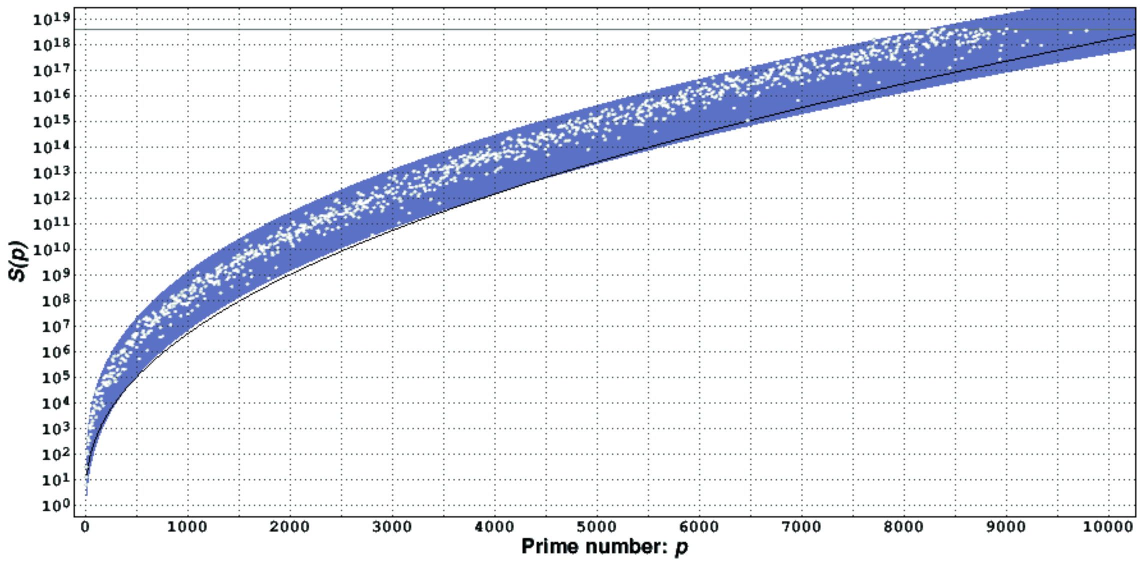
Subtask 5

วิธีนี้ มีพื้นฐานมาจาก Goldbach's conjecture ซึ่งยังไม่สามารถพิสูจน์ได้ ก่อนอื่น สำหรับ N ที่เป็นจำนวนเต็มคี่ จะต้องมาจาก $2 + N - 2$ แน่ๆ เพราะว่าจำนวนคี่จะมาจากจำนวนคู่บวกกับจำนวนคี่ และ 2 เป็นจำนวนคู่และเป็นจำนวนเฉพาะที่น้อยที่สุดด้วย ที่เหลือก็ต้องทดสอบว่า $N - 2$ เป็นจำนวนเฉพาะหรือไม่ ซึ่งทำได้ใน $\mathcal{O}(\sqrt{N})$

ต่อมา สำหรับ N ที่เป็นจำนวนคู่ จะตรงกับ Goldbach's conjecture พอดี ซึ่งยังไม่สามารถพิสูจน์ได้ในปัจจุบัน แต่มีผู้แสดงให้เห็นว่าสำหรับทุกๆจำนวนเต็มที่ไม่เกิน 10^9 จะใช้จำนวนเฉพาะ A ไม่เกิน $2,00$ โดยทำ Simulation จากคอมพิวเตอร์ (Tomás Oliveira e Silva, 2013)

ดังนั้นจึงสามารถไล่จำนวนเฉพาะ A ทั้งหมดที่ไม่เกิน $2,000$ หลังจากนั้น สำหรับแต่ละจำนวนเฉพาะ A ให้ตรวจสอบว่า $B = N - A$ เป็นจำนวนเฉพาะหรือไม่ โดยใช้ $\mathcal{O}(\sqrt{B})$ Primality Test

Time Complexity: $\mathcal{O}(\sqrt{N})$ (Assumption: 2,000 เป็นค่าคงที่)



รูปที่ 8: รูปแสดงผลรวม $N = S(p)$ ทั้งหมดที่เป็นไปได้ และ $A = p$

ภาพจาก `http://sweet.ua.pt/tos/goldbach.html` (Tomás Oliveira e Silva, 2013).

สามกล่อง (100 คะแนน)

0.1 seconds, 64 megabytes

โจทย์สรุปคือ ให้หาความน่าจะเป็นที่ หยิบกล่องสามกล่อง (i, j, k) แล้ว $max(A_i, A_j, A_k) > K$

Subtask 1

สามารถลองทุกกรณี (i, j, k) ที่เป็นไปได้ แล้วตรวจสอบว่า $max(A_i, A_j, A_k) > K$ หรือไม่ หารด้วยจำนวนกรณีทั้งหมดที่ไม่ได้ตรวจสอบ

Time Complexity: $\mathcal{O}(N^3)$

Subtask 2

ลองไล่ j ตั้งแต่ 1 ถึง N แล้วดูว่าหากให้ j เป็นตัวตรงกลางจะมีกี่วิธีที่เป็นไปได้ ซึ่งนับได้โดยแยก 2 กรณีคือ

1. $A_j > K$ จะได้ว่ามีทั้งหมด $(j-1)*(N-j)$ วิธี เนื่องจากมีตัวทางซ้ายทั้งหมด $j-1$ ตัว และตัวทางขวา $N-j$ ตัว

2. $A_j \leq K$ จะได้ว่าจะต้องหยิบเฉพาะกรณีที่ $max(A_i, A_k) > K$ ซึ่งนับได้โดยการไล่หาว่ามีกี่ตัวที่อยู่ทางซ้ายของ j และมีค่ามากกว่า K คูณกับ มีกี่ตัวที่อยู่ทางขวาของ j และมีค่ามากกว่า K

ต่อมา นำจำนวนวิธีของทั้งสองกรณี หารด้วยวิธีทั้งหมด นั่นคือ $\binom{N}{3}$ ซึ่งเท่ากับ $\frac{(N-1)(N-2)(N-3)}{6}$

Time Complexity: $\mathcal{O}(N^2)$

Subtask 3

สังเกตว่าลำดับในที่นี้ไม่สำคัญ เพราะว่าโจทย์กำหนดไว้แล้วว่าต้องเลือกจากซ้ายไปขวา จึงได้ปัญหาย่อยคือ "มีทั้งหมดกี่วิธี ที่จะเลือกจำนวนเต็ม 3 จำนวนจากเซตซ้ำขนาด N แล้ว ค่ามากที่สุดของ 3 จำนวนนี้มากกว่า K"

วิธีการแก้ปัญหานี้คือเรียงลำดับจำนวนเต็มจากน้อยไปมากก่อน แล้วค่อยๆไล่หาจากซ้ายไปขวา จะมีจุดแบ่ง x ซึ่ง $A_x, A_{x+1}, A_{x+2}, \ldots, A_N$ มีค่ามากกว่า K ทั้งหมด และ $A_{x-1}, A_{x-2}, \ldots, A_1$ มีค่าน้อยกว่าหรือเท่ากับ K เพื่อความสะดวก จะเรียกว่า ส่วนขวา และ ส่วนซ้าย ตามลำดับ

จากนั้น จะได้ว่าจะต้องเลือกให้จำนวนเต็มอย่างน้อยหนึ่งตัว อยู่ภายในส่วนขวา ที่เหลืออีก 2 จำนวน อาจจะอยู่ส่วนซ้ายหรือขวาก็ได้ จึงไล่หาแต่ละตัวในส่วนขวา สมมติให้ดัชนีของตัวนั้นเป็น k จะมีวิธีเลือก i, j ได้ $\binom{k-1}{2}$ วิธี เพราะ $i < j < k$ และ $max(A_i, A_j, A_k) = A_k > K$ ซึ่งอสมการหลังนี้สามารถตัดทิ้งได้เพราะเราเรียงลำดับแล้วและเลือก k เฉพาะในส่วนขวา

หลังจากนั้นนำไปหารด้วยจำนวนวิธีทั้งหมด ซึ่งเท่ากับ $\binom{N}{3}$

Time Complexity: $\mathcal{O}(N \log N)$