

Spoilers

May 7

Abnormal

ทำใน Contest ได้ 32 คะแนน หลังจากจบ contest ยังไม่ได้คิดต่อ แต่เฉลยจะเป็น

- แบบที่ 1 ใช้เพิ่ม L บิต
- แบบที่ 2 ใช้เพิ่ม $\mathcal{O}(\log N)$ บิต
- แบบที่ 3 ใช้เพิ่มประมาณ $L + \mathcal{O}(\log N)$ บิต

BTS

สมมติก่อนที่เราไม่สามารถสับเปลี่ยนราคาก่อนของหน่วยงาน ABC ได้ และให้ $A < B$

พิจารณา cycle C ใด ๆ ของกราฟ G จะมี MST T ที่ไม่มีเส้นเชื่อมที่มีมูลค่ามากที่สุดในนั้น เพื่อที่จะทำให้เส้นเชื่อม BTS ของเราโดนเลือกแน่ ๆ จะจึงต้องตั้งราคาให้น้อยกว่าทุกเส้น ๆ ของ ABC ที่ "คลุม" เส้นเชื่อมนั้น ๆ ดังนั้นถ้าเราไม่สามารถสับเปลี่ยนราคาได้ เราก็สามารถคำนวณคำตอบได้ทันที

จากนี้เราจะได้อีกว่า ถ้ามีคน ABC ที่มีค่าน้อยกว่าหรือเท่ากับ A จะไม่มีทางทำได้ เพราะจะมีเส้นหนึ่งของ ABC ที่เราไม่สามารถบังคับไม่ให้อยู่ใน MST ได้

ดังนั้นจะเหลือแค่เคสที่เส้นของ ABC มีค่า $> A$ เท่านั้น

คราวนี้มาลองพิจารณากรณีที่ราคาของ ABC สับเปลี่ยนได้ สังเกตต่อไปอีกว่ามูลค่าของเส้นเชื่อมนั้นไม่สำคัญ สำคัญแค่ว่ามัน $> B$ หรือไม่ ขอเรียกเส้นที่มีมูลค่า $> B$ ว่าแบบที่ 1 และอีกแบบเป็นแบบที่ 2

สมมติว่ามีเส้นแบบที่ 2 ทั้งหมด k เส้น สังเกตอีกว่า เพื่อที่จะรับประกันว่า BTS ถูกเลือกทั้งหมด ทุก ๆ เส้น BTS ที่โดนเส้นแบบที่ 2 คลุม จะต้องมีมูลค่าเป็น A ทั้งหมด ดังนั้นเราควรที่จะ minimize พื้นที่ที่ถูกคลุมโดยเส้นแบบที่ 2 เพื่อที่จะได้คำตอบที่ดีที่สุด โจทย์ของเราจึงกลายเป็น ให้ช่วงมา M ช่วง ให้เลือกมาทั้งหมด k ช่วงโดยที่ union ของพื้นที่ที่เลือกนั้นมีค่าน้อยที่สุด ซึ่งสามารถคำนวณได้ใน $\mathcal{O}(NMk)$ โดย dynamic programming เมื่อ N คือจำนวนระยะทั้งหมดที่เราจะพิจารณา (ก็คือจำนวนเส้นเชื่อม BTS นั้นเอง)

Camp

DP แต่อาจจะต้องใช้ segment tree ช่วย speedup การคำนวณ

Marathon

ทำไม่ได้ แต่ฟังเฉลยผ่าน ๆ เหมือนจะต้องพิจารณา BFS กราฟของกราฟนี้ แล้วพิจารณาแต่ละ layer ไป

May 9

Househouse

สิ่งที่สำคัญสำหรับข้อนี้คือ grader เป็น adaptive นั่นคือมันสามารถปรับคำตอบของตัวเองไประหว่างสิ่งที่เรากำหนดได้ ต้องพิจารณาเคสดี ๆ ถ้าจำไม่ผิดจะได้ $R = (2C - 1)/C$

Judtaew

DP optimization ไอเดียค่อนข้างแปลกใหม่ ไว้มาเขียนต่อ

Parent

LCA + DSU

Adder

สร้าง gate ที่คำนวณโดยตรงว่า carry จากหลักที่แล้วจะโดนส่งต่อไปหลักต่อไปหรือไม่ ผลลัพธ์ของแต่ละหลักจะเป็นการทำ operation จากบิต ๆ ก่อน ไม่เกิน 2-3 ชั้น แต่แต่ละชั้นจะมีหลาย ๆ bit ที่โดน operation เดียวกัน ซึ่งเราสามารถสร้างเป็น binary tree เพื่อลดความลึกของ gate ได้ optimal (ของอาจารย์) ของเคสที่ 3 และ 4 คือ 14 และ 16

May 11

Exam

ถ้าเราอยากให้มี path จากทุก ๆ โหนด ใน A ไปทุก ๆ โหนดใน B เราไม่จำเป็นต้องสร้างเส้นเชื่อมทั้งหมด $|A||B|$ เส้น ที่เหลือก็ตรงตามนิยามของสิ่ง ๆ หนึ่งใน graph theory (ไม่ได้ทำอย่างจริงจังตอนสอบ)

Flagarea

Geometry ที่ solution สวยมาก เป็นการพิจารณา ความยาวที่โดนคลุม ของ midpoint ของ interval จากสมบัติความสมมาตรของสามเหลี่ยม

Insert

Greedy. ลองหา configuration ทั้งหมดที่ดีที่สุดของคำตอบก่อน

Twomachinejobs

สมมติว่า schedule งานเราทำงานชิ้นที่ i แล้วค่อยทำงานชิ้นที่ j สมมติให้ $B_i < B_j$ และชิ้นที่ i เข้าเครื่องแบบที่ 1 ที่เวลา T จะได้ว่างานชิ้นที่ i จะเสร็จที่เวลา $T + A_i + B_i$ และงานชิ้นที่ j จะเสร็จที่เวลา $T + A_i + A_j + B_j$ แต่ถ้าหากสลับกัน งานชิ้นที่ i จะเสร็จที่เวลา $T + A_j + A_i + B_i$ และงานชิ้นที่ j จะเสร็จที่เวลา $T + A_j + B_j$ เนื่องจากงานชิ้นต่อจาก j และก่อนหน้า j จะเริ่มที่เวลาเดียวกันทั้งสองกรณี (ขึ้นต่อจาก j เริ่มที่เวลา $T + A_i + A_j$) เราจึงสนใจแค่ max ของสองอันนี้ ให้

$X = \max(T + A_i + B_i, T + A_i + A_j + B_j)$ และ $Y = \max(T + A_i + A_j + B_i, T + A_j + B_j)$ เนื่องจาก $T + A_j + B_j < T + A_i + A_j + B_j$, $T + A_i + A_j + B_i < T + A_i + A_j + B_j$ (จาก assumption ด้านบน) จะได้ว่า $X > Y$ เสมอ ดังนั้นจาก exchange argument เราควรทำงานจาก B_i มากไปน้อย สิ่งที่เหลือก็คือการคำนวณ

หาก B_i ให้มา offline ก็คำนวณได้ไม่ยาก แต่ถ้าจะทำ online ต้องใช้โครงสร้าง implicit cartesian tree ซึ่งเป็น segment tree แบบสามารถ insert หรือ delete ข้อมูลได้

มาลองดูวิธีหา comparator P (สำหรับข้อนี้คือ $B_i > B_j$) สำหรับโจทย์แนวนี้นี้ ปกติแล้ว proof ของ exchange argument จะเป็นไปตามนี้

"ถ้า P ไม่เป็นจริงแล้ว การสับเปลี่ยนลำดับจะทำให้คำตอบไม่แย่ลง" (1)

จากกฎ contrapositive คำพูดข้างบนจะมีค่าเท่ากับคำว่า

"ถ้าการสับเปลี่ยนลำดับทำให้คำตอบแย่ลง P จะเป็นจริง" (2)

ดังนั้นเวลาหา P เราควรจะเริ่มจาก scenario ที่การสับเปลี่ยนคำตอบมันแย่ลง และถ้าโชคดี วิเคราะห์เสร็จเราจะได้ P มาในที่สุด (แนะนำให้ลองเช็ค P ตามเงื่อนไข (1) อีกทีด้วย) ในข้อนี้ scenario คือ

$$X = \max(T + A_i + B_i, T + A_i + A_j + B_j) < Y = \max(T + A_i + A_j + B_i, T + A_j + B_j)$$

ซึ่งเมื่อลองวิเคราะห์ดูแล้วจะสรุปได้ว่า $B_j < B_i$ ซึ่งก็คือ P ของเรานั่นเอง

May 15

Mooz Server

ไอเดียประมาณ 85 ค่ะเนนคือ สร้าง dfs tree แล้วพิจารณา edge

Repeats

Sliding Window

Triquery

Fancy Way of Keeping Quicksum

Candyreserve

เราสามารถบอกได้ว่าเราต้องจับติดกับขอบซ้ายของจุดเริ่มเท่านั้น ที่เหลือก็ sweep line + segment tree DP

May 16

Fanclub

Segment Tree + LCA

GG

Recursive Graph, ค่อย ๆ หาคำตอบของชั้นน้อย ๆ แล้วรวมขึ้นไป

Mastermind

จริง ๆ แล้วข้อนี้เป็น output only ต้องสร้าง decision tree ไว้ก่อน แล้วใช้เป็น constant array ของโปรแกรมที่จะรัน

Polesropes

Segment Tree