48\_Table

ก็ข้อนี้นะถ้าสังเกตจุดอ่อนของโจทย์จะพบว่ามีจุดอ่อนอยู่ที่ R น้อย 300 เอง ซึ่งจะสังเกตว่าเราสามารถวน M\*R ได้ และจากคำสั่งจะพบว่ามีคำสั่งที่ 2 ที่จะทำให้เราวนตารางด้วยค่า C ซึ่งถ้าเราวนตามโจทย์ก็จะติด t นั่นเอง เพราะงั้นเราจะไม่วนตามค่า C แต่เราจะมีตัวเก็บข้อมูลตัวนึงแทนทั้งแถว (แถวๆนึงมี C คอลัมน์ ถ้าเราเก็บทั้งแถวได้ เราก็จะไม่วน C) อันนี้ทำไว้สำหรับคำสั่งที่ 2 โดยเฉพาะนะ ทีนี้จะเริ่มบอกความยากของข้อนี้ ก็คือการ implement เหตุมาจากการที่ว่าเราต้องเก็บทั้งแบบตารางช่องเดี่ยวสำหรับคำสั่งแรก กับเก็บแบบทั้งแถวสำหรับคำสั่งที่ 2 เพราะงั้นเวลาเราจะตอบคำถามแรก ที่เอาค่าของช่องๆนั้นมา เราก็จะต้องเลือกว่าช่องนั้นโดน setValue หรือ setRowValue ทีหลัง อันไหนเป็นทำทีหลังก็ต้องตอบอันนั้น เพราะงั้นในอาเรย์ที่เราก็ค่าที่เซ็ทไว้สำหรับตัวเดียวหรือทั้งแถว ก็ต้องเก็บด้วยว่ามันโดนเปลี่ยนครั้งล่าสุดตอนไหน ส่วนสำหรับ getMin อันนี้จะยาก เราจะใช้ heap เข้ามาช่วยในการหาค่า min โดยทุกรอบที่เราทำการเซ็ทค่าจะต้องเอาค่า เวลา ช่อง/แถว ลงใน heap เพื่อที่จะได้เช็คตอนเราดู min ว่ามันเป็นน้อยสุดที่ยังไม่โดนเซ็ทหรือป่าว หากโดนเซ็ททับไปแล้วก็ป้อบทิ้งได้ วิธีการเช็คก็ หากเป็นแบบช่อง ก็ดูว่าช่องนั้นโดนเซ็ทไปยัง ถ้ายังก็ต้องดูตรงเซ็ทแถวต่อ ว่าแถวที่มันอยู่โดนเซ็ททับไปยัง ถ้ายังก็ตอบได้ ถ้าโดนแล้วก็ป้อบ heap แล้วก็ไปหาอันใหม่ ต่อไปสำหรับตัวที่เราป้อบมาพบว่าเป็นการเซ็ทแบบทั้งแถว อันนี้ เราก็เช็คก่อนว่าแถวนั้นโดนเซ็ทใหม่ยัง ถ้าโดนแล้วก็ป้อบทิ้ง ถ้ายังก็เช็คต่อ ซึ่งสำหรับการเช็ค min ทั้งแถวเนี่ย ถ้าเราไปนั่งวนเช็คคงเสียเวลาจนติด t เลย เหตุผลที่เราต้องวนเช็คก็คือเราไม่รู้ว่าทั้งแถวนั้นโดนเซ็ททีละช่องจนครบทั้งแถวหรือป่าว ไรงี้ เพราะงั้นเวลาเราพบว่าค่าน้อยสุดเกิดจากการเซ็ททั้งแถว เราเลยต้องดูว่ามีช่องไหนในแถวนั้นที่ไม่โดนเซ็ททับหรือป่าว หากมีก็ที่ไม่โดนก็แสดงว่าค่าเซ็ทแถวรอบนี้ใช้ได้ แต่หากโดนทั้งแถวเลยก็ใช้ไม่ได้ วิธีการทำให้เร็วๆ ก็คือเราจะมีอาเรย์ขึ้นมาตัวนึง สำหรับการนับว่าคำสั่งเซทเถวนี้เหลือกี่ตัวที่ยังไม่เซ็ททับ ซึ่งอาเรย์นี้เริ่มต้นค่าก็ต้องเท่ากัน C แต่พอมี setValue มาโดนในแถวตัวที่ยังไม่เปลี่ยน ก็ลดค่าลง แต่หาก setValue โดนอันที่เคยทับแล้วก็ปล่อยได้ หลังว่าจะพอเข้าใจนะ