

รถไฟ

จากเบตงถึงแม่สายมีสถานีรถไฟทั้งหมด n สถานี แทนแต่ละสถานีด้วย $1, 2, 3, \dots, n$ เรียงกันเป็นลำดับจากเบตงถึงแม่สาย ให้สถานีที่ 1 อยู่ที่เบตง และสถานีที่ n อยู่ที่แม่สาย โดยแต่ละสถานีสามารถเดินทางถึงกันได้ และค่ารถไฟจากสถานีหนึ่งไปสถานีหนึ่งแตกต่างกันและไม่ขึ้นอยู่กับระยะทาง ให้เขียนโปรแกรม เพื่อหาค่าโดยสารที่น้อยที่สุดในการเดินทางจากสถานีเบตงไปแม่สาย โดยในการเดินทางระหว่างสถานี i และ j ใดๆ i จะต้องน้อยกว่า j เสมอ

ข้อมูลนำเข้า

บรรทัดที่ 1 จำนวนเต็ม 1 จำนวน แทนจำนวนสถานี n

บรรทัดที่ 2 ถึง บรรทัดที่ $(n(n-1)/2)+1$ แต่ละบรรทัดมีจำนวนเต็ม 3 จำนวน ได้แก่ i, j, k โดยที่ i แทนสถานีต้นทาง j แทนสถานีปลายทาง และ k แทนค่าโดยสารรถไฟจากสถานี i ไปสถานี j เมื่อ $i < j$ เช่น 1 2 5 หมายความว่า ค่ารถไฟจากสถานี 1 ไปสถานี 2 เท่ากับ 5 เป็นต้น

ข้อมูลออก ค่ารถไฟที่น้อยที่สุดสำหรับเดินทางจากเบตงไปแม่สาย

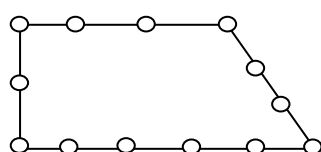
ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
5 1 2 5 1 3 4 1 4 8 1 5 9 2 3 1 2 4 2 2 5 2 3 4 4 3 5 5 4 5 3	7

ชาวนา

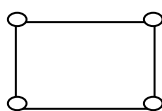
ชาวนามีสนามหลายสนาม แต่ละสนามล้อมรอบด้วยต้นไม้ นอกจากนี้ยังมีพื้นที่แถบ ซึ่งมีต้นไม้ปลูกเป็นแถว ทั้งสนามและพื้นที่แถบจะมีต้นมะกอกอยู่ 1 ต้น ปลูกคั่นอยู่ระหว่างต้นไม้สองต้นใดๆ ที่อยู่ติดกันเสมอ ต้นไม้ทั้งหมดที่ชาวนามีจะปลูกล้อมรอบสนามหรืออยู่ในพื้นที่แถบเท่านั้น และต้นมะกอกทั้งหมดที่ชาวนามีจะอยู่ระหว่างต้นไม้ 2 ต้นใดๆ ที่อยู่ติดกันเท่านั้น

วันหนึ่งชาวนาป่วยมากและรู้สึกที่กำลังจะตาย สองสามวันก่อนที่เขาจะเสียชีวิตลง เขาได้เรียกบุตรชายคนโตมาบอกว่า “พ่อจะให้ต้นไม้ต้นไม้ใดก็ได้ Q ต้น และต้นมะกอกทั้งหมดที่อยู่ระหว่างต้นไม้สองต้นใดๆ ที่อยู่ติดกันที่เลือก” เนื่องจากบุตรชายคนโตชอบต้นมะกอก จึงพยายามเลือกต้นไม้จำนวน Q ต้น ซึ่งจะทำให้ได้ต้นมะกอกจำนวนมากที่สุดเท่าที่จะเป็นไปได้ โดยจะเลือกต้นไม้ใดจากสนามหรือพื้นที่แถบใดก็ได้

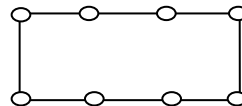
ดูจากตัวอย่างรูปที่ 1 สมมติว่าลูกชายได้ต้นไม้จำนวน $Q = 17$ ต้น เพื่อที่จะทำให้เขาได้ต้นมะกอกมากที่สุดเท่าที่จะเป็นไปได้ เขาควรที่จะเลือกต้นไม้ทั้งหมดที่อยู่ในสนามที่ 1 และสนามที่ 2 และได้ต้นมะกอกจำนวนรวมทั้งหมด 17 ต้น



สนามที่ 1 มีต้นไม้ 13 ต้น



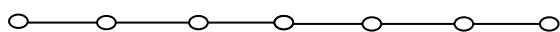
สนามที่ 2 มีต้นไม้ 4



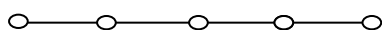
สนามที่ 3 มีต้นไม้ 8 ต้น



พื้นที่แถบที่ 1 มีต้นไม้ 4 ต้น



พื้นที่แถบที่ 2 มีต้นไม้ 8 ต้น



พื้นที่แถบที่ 3 มีต้นไม้ 6 ต้น

รูปที่ 1 ตัวอย่างรูปแบบของต้นไม้ ส่วนต้นมะกอกไม่ได้แสดงอยู่ในรูป

จงเขียนโปรแกรมเพื่อหาจำนวนต้นมะกอกที่มากที่สุดเท่าที่จะเป็นไปได้ที่บุตรชายจะได้เป็นเจ้าของตามข้อมูลที่ให้เกี่ยวกับสนามและพื้นที่แถบและจำนวนต้นไม้ที่ลูกชายจะเลือกได้

ข้อมูลนำเข้า

บรรทัดที่ 1 มีสามจำนวน ตัวแรกเป็นจำนวนเต็ม Q แทนจำนวนของต้นไม้ที่บุตรชายได้รับ ตัวที่สองเป็นจำนวนเต็ม M แทนจำนวนของสนาม และตัวที่สามเป็นจำนวนเต็ม K แทนจำนวนของพื้นที่แถบ

บรรทัดที่ 2 มีตัวเลขจำนวนเต็ม M จำนวน N_i ได้แก่ N_1, N_2, \dots, N_M ซึ่งเป็นจำนวนต้นไม้ในสนามต่าง ๆ

บรรทัดที่ 3 มีตัวเลขจำนวนเต็ม K จำนวน R_i ได้แก่ R_1, R_2, \dots, R_K ซึ่งเป็นจำนวนของต้นไม้ที่อยู่ในพื้นที่แถบต่างๆ

ข้อมูลส่งออก

บรรทัดที่ 1 จำนวนเต็ม แสดงจำนวนต้นมะกอกที่มากที่สุดที่บุตรชายจะได้เป็นเจ้าของ

ตัวอย่างข้อมูลนำเข้า 1	ตัวอย่างข้อมูลส่งออก 1
17 3 3 13 4 8 4 8 6	17
ตัวอย่างข้อมูลนำเข้า 2	ตัวอย่างข้อมูลส่งออก 2
14 3 2 6 3 4 7 2	13

เงื่อนไข สำหรับข้อมูลนำเข้าทั้งหมด

$$0 \leq Q \leq 150000$$

$$0 \leq M \leq 2000$$

$$0 \leq K \leq 2000$$

$$3 \leq N_i \leq 150$$

$$3 \leq R_i \leq 150$$

จำนวนต้นไม้ทั้งหมดมีไม่น้อยกว่า Q ต้น

ข้อมูลถนน

มลรัฐแห่งหนึ่งในประเทศคอร์บัสถานประกอบด้วย N เมือง แต่ละเมืองแทนชื่อเมืองด้วยจำนวนเต็ม 1 ถึง N และมีถนน M สายเพื่อใช้เชื่อมเมืองต่าง ๆ เข้าด้วยกัน โดยที่ไม่มีถนนสายใดที่เชื่อมกับเมืองเพียงเมืองเดียว ทุกเมืองเชื่อมกับถนนอย่างน้อยหนึ่งสาย และเป็นไปได้ที่จะมีถนนหลายสายเชื่อมระหว่างสองเมืองใด ๆ

ถนนทุกสายในมลรัฐแห่งนี้เสื่อมโทรมมาก ดังนั้นรัฐบาลกลางของประเทศคอร์บัสถานจึงต้องการมอบงบประมาณให้กับมลรัฐแห่งนี้เพื่อใช้ในการปรับปรุงถนนสายเหล่านั้น โดยมีข้อกำหนดดังต่อไปนี้

- ถนนสายใดที่ถูกปรับปรุงต้องถูกปรับปรุงทั้งสาย (ห้ามทำครึ่ง ๆ กลาง ๆ แบบประเทศไทย)
- จำนวนสายของถนนที่ถูกปรับปรุงต้องมีจำนวนน้อยที่สุด แต่ทุกเมืองต้องเชื่อมกับถนนที่ถูกปรับปรุงอย่างน้อยหนึ่งสาย

รัฐบาลกลางจะใช้งบประมาณเพื่อใช้ในการปรับปรุงถนนเป็นจำนวนเงินเท่ากับ $\$100,000 \times$ ความยาวรวมในหน่วยกิโลเมตร ของถนนทุกสายที่ถูกปรับปรุง

นายจอห์นเป็นผู้ว่าการของมลรัฐแห่งนี้ เขาจะเป็นผู้เสนอว่าถนนสายใดบ้างควรได้รับการปรับปรุง และด้วยเหตุผลส่วนตัว เขาต้องการได้รับงบประมาณมากที่สุดจากรัฐบาลกลาง ดังนั้นนายจอห์นต้องการเลือกถนนจำนวน K สาย $K \leq M$ ที่ความยาวรวมของถนนเหล่านั้นมีค่ามากที่สุด แต่ยังเป็นไปตามข้อตกลงของรัฐบาลกลาง

จงเขียนโปรแกรมที่มีประสิทธิภาพเพื่อคำนวณหาความยาวรวมที่มากที่สุดของถนนที่ถูกปรับปรุง และจำนวนสายของถนนที่ถูกปรับปรุง

ข้อมูลนำเข้า

บรรทัดที่ 1	มีจำนวนเต็มสองจำนวน N และ M แต่ละจำนวนถูกคั่นด้วยช่องว่างหนึ่งช่องว่าง โดยที่ N แสดงจำนวนเมือง และ M แสดงจำนวนถนน $2 \leq N \leq 40,000$; $1 \leq M \leq 100,000$
บรรทัดที่ 2 ถึง $M+1$	แต่ละบรรทัดแสดงถนนแต่ละสาย ด้วยจำนวนเต็มสามจำนวน s_i , d_i และ l_i โดยที่ s_i และ d_i แสดงชื่อเมืองที่ถนนแต่ละสายเชื่อมไว้ และ l_i แสดงความยาวของถนนแต่ละสายในหน่วยกิโลเมตร

ข้อมูลส่งออก

บรรทัดที่ 1 แสดงความยาวรวมที่มากที่สุดของถนนทุกสายที่ถูกปรับปรุง

บรรทัดที่ 2 แสดงจำนวนสายของถนนที่ถูกปรับปรุง

ตัวอย่างข้อมูลนำเข้า 1	ตัวอย่างข้อมูลส่งออก 1
6 9 1 2 8 2 3 6 1 4 6 4 2 6 4 5 8 2 5 7 5 6 5 2 6 9 3 6 5	38 5
ตัวอย่างข้อมูลนำเข้า 2	ตัวอย่างข้อมูลส่งออก 2
9 11 1 2 6 2 3 7 1 9 2 2 5 9 1 6 4 3 4 4 5 7 8 6 7 2 4 7 5 4 8 1 3 5 3	42 8

เอ็มโพเดีย

พิธาอรัส เป็นนักคณิตศาสตร์และนักปรัชญากรีกผู้มีชื่อเสียงคนหนึ่ง ซึ่งมีความเชื่อว่า ความจริงตามธรรมชาติทุกเรื่องสามารถอธิบายได้ด้วยคณิตศาสตร์ ในปัจจุบัน นักชีววิทยาหลายคนก็ศึกษาข้อมูลทางชีววิทยากับแนวคิดทำนองเดียวกันนี้ โดยศึกษาสมบัติของ สายลำดับชีวภาพ (biosequence) โดยลำดับชีวภาพนี้จะเป็นลำดับของจำนวนเต็ม M ตัว ซึ่งลำดับชีวภาพนี้จะ

- ประกอบด้วยจำนวนเต็ม $0, 1, 2, \dots, M-1$
- เริ่มด้วย 0 และสิ้นสุดด้วย $M-1$
- ไม่มีจำนวนเต็ม $E, E+1$ ในตำแหน่งที่ติดกันใด ๆ ภายในลำดับ

ลำดับย่อยที่ประกอบด้วยจำนวนเต็มที่ต่อเนื่องกัน (subsequence consisting of adjacent elements) ของลำดับชีวภาพหนึ่ง ๆ จะเรียกว่า เซกเมนต์ (segment)

เซกเมนต์ของลำดับชีวภาพใด ๆ จะถูกเรียกว่า ช่วงล้อมกรอบ (framed interval) ถ้าเซกเมนต์ประกอบด้วยจำนวนเต็มทุกตัวซึ่งมีค่าระหว่างตัวแรกและตัวสุดท้ายของเซกเมนต์ โดยตัวแรกของเซกเมนต์จะเป็นจำนวนที่มีค่าน้อยที่สุด และตัวสุดท้ายของเซกเมนต์ (ซึ่งต้องเป็นคนละตัวกับตัวแรก) จะเป็นจำนวนที่มีค่ามากที่สุดของเซกเมนต์นั้น ๆ นอกจากนี้ ถ้าช่วงล้อมกรอบหนึ่งไม่มีช่วงล้อมกรอบย่อยที่สั้นกว่าภายในตัวมันเองแล้ว ช่วงล้อมกรอบนี้จะเรียกว่า “เอ็มโพดีโอ (empodio)”

ตัวอย่างเช่น ให้พิจารณาลำดับชีวภาพ $(0, 3, 5, 4, 6, 2, 1, 7)$ จะสังเกตได้ว่า ลำดับชุดนี้เป็นช่วงล้อมกรอบ อย่างไรก็ตามลำดับชุดนี้ประกอบด้วยช่วงล้อมกรอบย่อย $(3, 5, 4, 6)$ อยู่ด้วย ทำให้ช่วงล้อมกรอบ $(0, 3, 5, 4, 6, 2, 1, 7)$ ไม่เป็นเอ็มโพดีโอ แต่ว่า $(3, 5, 4, 6)$ เป็นเอ็มโพดีโอ เพราะว่าไม่มีช่วงล้อมกรอบที่สั้นกว่าอยู่ภายใน สรุปว่าลำดับชีวภาพ $(0, 3, 5, 4, 6, 2, 1, 7)$ นี้มีเพียง $(3, 5, 4, 6)$ เป็นเอ็มโพดีโออยู่เพียงชุดเดียวเท่านั้น

จงเขียนโปรแกรมเพื่อหาเอ็มโพดีโอทั้งหมดในลำดับชีวภาพที่โจทย์ให้มา

ข้อมูลนำเข้า

บรรทัดแรก จำนวนเต็ม M ซึ่งเป็นจำนวนของจำนวนเต็มทั้งหมด

บรรทัดที่ $2 - M+1$ เป็นจำนวนเต็มของลำดับชีวภาพเรียงตามลำดับบรรทัดละหนึ่งตัว

ข้อมูลส่งออก

บรรทัดแรก จำนวนเต็ม H เป็นจำนวนเอ็มโพดิโอทั้งหมด

บรรทัดที่ $2-H+1$ แต่ละบรรทัดแสดงเอ็มโพดิโอแต่ละตัว โดยเรียงลำดับจากเริ่มต้นปรากฏในลำดับ

ชีวภาพ โดยแต่ละบรรทัดแสดงโดยใช้ตัวเลขสองตัว A และ B โดยที่ A เป็นตำแหน่งเริ่มต้น และ B เป็นตำแหน่งสุดท้ายของเอ็มโพดิโออื่นๆ

ตัวอย่างข้อมูลนำเข้า 1	ตัวอย่างข้อมูลส่งออก 1
8 0 3 5 4 6 2 1 7	1 2 5
ตัวอย่างข้อมูลนำเข้า 2	ตัวอย่างข้อมูลส่งออก 2
15 0 2 4 3 5 7 1 6 8 13 9 11 10 12 14	2 2 5 11 14

เครื่องจำหน่ายตั๋ว

เครื่องจำหน่ายตั๋วแบบใหม่สำหรับขนส่งสาธารณะ จะใช้การ์ดเพียงใบเดียวในการซื้อตั๋วโดยสาร โดยมีตั๋วโดยสาร 3 แบบดังนี้

1. ตั๋วสำหรับการเดินทาง 1 ครั้ง ราคา 20 บาท
2. ตั๋วสำหรับ 90 นาที ราคา 50 บาท
3. ตั๋วสำหรับ 1 วัน (1440 นาที) ราคา 120 บาท

สมมติให้ การเดินทาง 1 ครั้ง ใช้เวลา 1 นาที และตั๋วสำหรับ x นาทีใดๆ สามารถใช้ได้ตั้งแต่เวลา t ถึง $t+x-1$ เมื่อ t คือเวลาที่เริ่มใช้ตั๋ว เครื่องจำหน่ายตั๋วเครื่องนี้สามารถเลือกตั๋วที่ดีที่สุดให้กับผู้โดยสารได้อัตโนมัติ โดยจะพิจารณาจากการเดินทางที่ผ่านมาและการเดินทางปัจจุบัน แล้วเลือกตั๋วที่ดีที่สุด ที่ทำให้ผู้โดยสารจ่ายค่าโดยสารน้อยที่สุด

กำหนดให้ a คือราคาตั๋วโดยสารรวมน้อยที่สุดสำหรับการเดินทางตั้งแต่ครั้งที่ 1 ถึงครั้งที่ k ใดๆ และ b คือค่าโดยสารรวมน้อยที่สุดของการเดินทางตั้งแต่ครั้งที่ 1 ถึงครั้งที่ $k-1$ ดังนั้น $a-b$ คือค่าโดยสารที่ผู้โดยสารต้องจ่ายในการเดินทางครั้งที่ k งานของคุณ เขียนโปรแกรมเพื่อคำนวณค่าโดยสารที่ผู้โดยสารต้องจ่ายในการเดินทางแต่ละครั้ง

ข้อมูลนำเข้า ประกอบด้วย $n + 1$ บรรทัด

บรรทัดแรก จำนวนเต็ม n ($1 \leq n \leq 10^5$) แสดงจำนวนครั้งของการเดินทาง

บรรทัดที่ 2 ถึง $n+1$ เวลา t_k ($1 \leq k \leq 10^5$) หน่วยเป็นนาที แสดงเวลาเริ่มต้นของการเดินทางแต่ละครั้ง โดยแต่ละ t_k จะแตกต่างกันและเรียงลำดับจากน้อยไปมาก ($t_{k+1} > t_k$ เมื่อ $1 \leq k \leq n$)

ข้อมูลออก ประกอบด้วย n บรรทัด โดยแต่ละบรรทัดแสดงค่าโดยสารที่ผู้โดยสารต้องจ่ายในการเดินทางแต่ละครั้ง

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลออก
3	20
10	20
20	10
30	
10	20
13	20
45	10
46	0
60	20
103	0
115	0
125	20
150	20
256	10
516	

ลงทุน

ลงทุนได้ลงทุนในหุ้นด้วยเงินจำนวนหนึ่ง เป็นเวลา n วัน ในแต่ละวันลงทุนได้จัดยอดกำไรและขาดทุนไว้ทุกวัน ยกตัวอย่างเช่นลงทุนลงทุนมาเป็นเวลา 10 วัน ในแต่ละวันลงทุนได้กำไรหรือขาดทุนดังนี้

31, -41, 59, 26, -53, 58, 97, -93, -23, 84

หมายความว่า วันที่หนึ่งลงทุนได้กำไร 31 บาท วันที่สองลงทุนขาดทุน 41 บาท และวันที่สามลงทุนได้กำไร 59 บาท เป็นต้น

ลงทุนอยากรู้ว่าลงทุนสามารถทำกำไรได้มากที่สุดกี่บาท หากลงทุนสามารถเลือกช่วงเวลาที่เหมาะสมที่สุดได้ ตัวอย่างเช่น จากข้อมูลการลงทุนในหุ้น 10 วันของลงทุน

31, -41, 59, 26, -53, 58, 97, -93, -23, 84

ลงทุนสามารถทำกำไรได้มากที่สุด 187 บาท หากลงทุนลงทุนตั้งแต่วันที่ 3 ถึงวันที่ 7 ซึ่งแต่ละวันจะได้กำไรหรือขาดทุนดังนี้

59, 26, -53, 58, 97

โดยที่ $59+26-53+58+97 = 187$

จงเขียนโปรแกรมที่มีประสิทธิภาพเพื่อหากำไรมากที่สุดที่เป็นไปได้ จากข้อมูลกำไรและขาดทุนเมื่อลงทุนลงทุนไปได้ n วัน เมื่อกำไรที่มากที่สุดที่เป็นไปได้มีค่าเท่ากับผลบวกของลำดับย่อยที่อยู่ติดกันที่มีผลรวมมากที่สุด

ข้อมูลนำเข้า

- บรรทัดที่ 1 จำนวนเต็ม n เป็นจำนวนวันที่ลงทุนลงทุนไปในหุ้น
- บรรทัดที่ 2 จำนวนเต็ม n จำนวน เป็นข้อมูลกำไรและขาดทุนในแต่ละวัน ตั้งแต่วันที่ 1 ถึงวันที่ n เรียงตามลำดับ

ข้อมูลส่งออก

- บรรทัดที่ 1 กำไรมากที่สุดที่เป็นไปได้
- บรรทัดที่ 2 ลำดับย่อยที่ติดกันที่ทำให้ได้กำไรมากที่สุด

ตัวอย่างข้อมูลนำเข้า 1	ตัวอย่างข้อมูลส่งออก 11
10 31 -41 59 26 -53 58 97 -93 -23 84	187 59 26 -53 58 97
ตัวอย่างข้อมูลนำเข้า 2	ตัวอย่างข้อมูลส่งออก 2
15 -26 34 45 20 -2 -40 54 -30 53 -9 87 -51 10 5 -77	212 34 45 20 -2 -40 54 -30 53 -9 87

น้องมะลิ

ในแต่ละวัน น้องมะลิได้รับการติดต่อให้ไปออกงาน n งาน โดยแต่ละงานให้ค่าตอบแทนที่แตกต่างกันและอาจมีเวลาที่ทับซ้อนกัน น้องมะลิจึงจำเป็นต้องเลือกงาน m งาน ($m \leq n$) ที่ทำให้น้องมะลิรับงานได้มากที่สุดและได้รับค่าตอบแทนสูงสุดที่สุด ตัวอย่างเช่น น้องมะลิได้รับการติดต่อไปออกงาน 6 งาน ได้แก่

- งานที่ 1 เวลา 8.00-9.00 น. จะได้รับค่าตอบแทน 20 บาท
- งานที่ 2 เวลา 9.00-12.00 น. จะได้รับค่าตอบแทน 15 บาท
- งานที่ 3 เวลา 13.00-17.00 น. จะได้รับค่าตอบแทน 40 บาท
- งานที่ 4 เวลา 9.00-12.00 น. จะได้รับค่าตอบแทน 40 บาท
- งานที่ 5 เวลา 14.00-18.00 น. จะได้รับค่าตอบแทน 60 บาท
- งานที่ 6 เวลา 8.00-10.00 น. จะได้รับค่าตอบแทน 30 บาท

จะได้ว่า น้องมะลิต้องเลือกไปงานที่ 1 4 และ 5 ซึ่งจะได้รับค่าตอบแทน 120 บาท

จงเขียนโปรแกรมช่วยน้องมะลิเลือกรับงาน เพื่อรับงานให้ได้มากที่สุดและได้รับค่าตอบแทนสูงสุดที่สุด (หากมีตัวเลือกมากกว่าหนึ่งตัวเลือก ให้เลือกตัวเลือกที่ใช้เวลาน้อยที่สุด และหากเลือกตัวเลือกที่ใช้เวลาน้อยสุดแล้ว ยังเหลือตัวเลือกมากกว่าหนึ่งตัวเลือก ให้เลือกตัวเลือกใดตัวเลือกหนึ่ง)

ข้อมูลนำเข้า

- บรรทัดที่ 1 จำนวนเต็ม n แทนจำนวนงานที่ได้รับการติดต่อ
- บรรทัดที่ 2 ถึง $(n+1)$ จำนวนเต็ม 3 จำนวน ได้แก่ s_i , e_i , และ v_i แทนเวลาเริ่มงาน เวลาสิ้นสุดของงาน และค่าตอบแทนของงานลำดับที่ i ตามลำดับ

ข้อมูลส่งออก

- บรรทัดที่ 1 ค่าตอบแทนที่ได้รับ
- บรรทัดที่ 2 แสดงงานที่ได้รับเลือก

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
6	120
8 9 20	1 4 5
9 12 15	
13 17 40	
9 12 40	
14 18 60	
8 10 30	

แนวคิดการแก้ปัญหา

เอ็มโพเดีย

กำหนดให้ $data[k]$ คือข้อมูลตัวที่ k , i คือตำแหน่งต้นช่วง และ j คือตำแหน่งปลายช่วง

วิธีที่ 1: พิจารณาหาช่วง $[i, j]$ ทุกช่วง ว่าช่วงไหนคือ empodio โดยเริ่มไล่จากค่า i และตามด้วยค่า j

การพิจารณาว่าช่วงใดเป็น empodio สามารถเช็คได้ดังนี้

1. $data[i]$ และ $data[j]$ ควรจะมีค่าห่างกัน $j-i$
2. ภายในช่วง $[i, j]$ ทุกค่าต้องอยู่ในช่วง $data[i]$ ถึง $data[j]$ เช่นถ้า $data[i]=2$ และ $data[j]=5$ ค่าที่อยู่ในช่วง $[i, j]$ ต้องประกอบด้วย (2, 3, 4, 5) โดยตัวแรกต้องเป็นตัวที่มีค่าน้อยที่สุด ตัวสุดท้ายต้องเป็นตัวที่มีค่ามากที่สุด
3. ภายในช่วง $[i, j]$ ต้องไม่มีช่วงใด ๆ ที่มีคุณสมบัติตามข้อ 1 และข้อ 2

วิธีที่ 2: วิธีที่หนึ่งใช้เวลานาน $O(n^2)$ พยายามลดขนาดของข้อมูลโดยการลดข้อมูลที่เราสนใจลง โดยพิจารณาเฉพาะช่วง $[i, j]$ ที่ $j=k$ โดยที่ $data[k] > data[k-1]$ และนำวิธีที่ 1 มาประยุกต์ใช้

ชานา

จากโจทย์ สนามหญ้ากับพื้นที่แถบมีความต่างกันที่ ถ้าเราเลือกต้นไม้ทั้งหมด n ต้นในสนามหญ้าแล้ว เราจะได้ต้นไม้กอก n ต้น แต่ถ้าเราเลือกต้นไม้ที่พื้นที่แถบ เราจะได้ต้นไม้กอกทั้งหมด $n-1$ ต้น ดังนั้น เราควรเลือกต้นไม้ในสนามหญ้าไว้ก่อน

กำหนดให้ Q คือจำนวนต้นไม้ที่จะได้รับ และ S คือจำนวนต้นไม้ในสนามหญ้าทั้งหมดทุกสนาม รวมกัน ถ้าต้องการเลือกต้นไม้ Q ต้น แล้ว $Q > S$ ก็ให้เลือกต้นไม้ S ต้นในสนามหญ้าทั้งหมดก่อน ซึ่งจะทำให้ได้ต้นไม้กอก S ต้น และจำนวนต้นไม้ที่เหลือ $Q-S$ ต้น จากนั้นให้เลือกต่อในพื้นที่แถบ โดยการเลือกพื้นที่แถบใช้วิธี Greedy ก็คือเลือกต้นไม้จากพื้นที่แถบที่มีต้นไม้มากไว้ก่อน

แต่ถ้า $Q \leq S$ พยายามเลือกต้นไม้ให้ลงตัวกับจำนวนต้นไม้ในสนามหญ้าได้หรือไม่ ถ้าได้จะได้จำนวนต้นไม้ทั้งหมด Q ต้น แต่ถ้าไม่สามารถเลือกได้ลงตัว ก็จะได้จำนวนต้นไม้กอกจำนวน $Q-1$ ต้น

ข้อมถน

หา Maximum Spanning Tree

เครื่องจำหน่ายตัว

ให้ $\text{mincost}(k)$ เก็บค่าโดยสารรวมที่น้อยที่สุดที่ผู้โดยสารต้องจ่ายตั้งแต่เดินทางครั้งที่ 1 ถึงครั้งที่ k เราจะได้ว่า

$$\text{mincost}(k) = \begin{cases} 20 & \text{if } k = 1 \\ \min(\text{mincost}(k-1) + 20, \text{mincost}(i) + 50, \text{mincost}(j) + 120) & \text{if } k > 1 \end{cases}$$

เมื่อ $\text{mincost}(i)$ คือ ค่าโดยสารรวมที่น้อยที่สุด ตั้งแต่เดินทางครั้งที่ 1 ถึงครั้งที่ i โดยที่ $t_k - t_{i+1} \leq 90$

และ $\text{mincost}(j)$ คือ ค่าโดยสารรวมที่น้อยที่สุด ตั้งแต่เดินทางครั้งที่ 1 ถึงครั้งที่ j โดยที่ $t_k - t_{j+1} \leq 1440$

ลงทุน

กำหนดให้ $a[i]$ เก็บข้อมูลตัวเลขตัวที่ i และ ให้ $OPT(i)$ มีค่าเท่ากับผลรวมที่มากที่สุดของลำดับย่อยที่อยู่ติดกันที่มีช่องสุดท้ายอยู่ที่ช่อง $a[i]$ เราจะได้ว่า

$$OPT(i) = \begin{cases} a[1] & i = 1 \\ \max(a[i], OPT(i-1) + a[i]) & i > 1 \end{cases}$$

เราสามารถพิสูจน์ว่าสมการข้างบนเป็นจริงได้ด้วย induction ดังต่อไปนี้

(Base Case) ในกรณีนี้ $i = 1$ เราจะได้ว่ามีลำดับย่อยที่ติดกันที่จบที่ช่อง $a[1]$ อยู่เพียงแค่ลำดับย่อยเดียว คือ ลำดับย่อยที่มีช่อง $a[1]$ เพียงช่องเดียว ฉะนั้น $OPT(1) = a[1]$ ตามต้องการ

(Induction Case) สมมติว่า $OPT(i)$ มีค่าเท่ากับผลรวมที่มากที่สุดของลำดับย่อยที่ติดกันที่มีช่องสุดท้ายอยู่ที่ช่อง $a[i]$ สำหรับ $i \geq 1$ บางค่า พิจารณาลำดับย่อยที่มีผลบวกมากที่สุดจบที่ช่อง $a[i+1]$ เราจะได้ว่ามีความเป็นไปได้สองกรณีด้วยกัน

1. ลำดับย่อยนั้นเริ่มต้นที่ช่อง $a[i+1]$ ด้วย ในกรณีนี้ $OPT(i+1) = a[i+1]$
2. ลำดับย่อยนั้นไม่ได้เริ่มต้นที่ช่อง $a[i+1]$ ในกรณีนี้เราจะได้ว่าลำดับย่อยนั้นเกิดจากการนำเอาลำดับย่อยที่จบที่ช่อง $a[i]$ มาต่อกับช่อง $a[i+1]$ ฉะนั้นผลรวมที่มากที่สุดของลำดับย่อยนี้คือ $OPT(i) + a[i+1]$

เมื่อรวมสองกรณีข้างบนเข้าด้วยกัน เราจะได้ว่า

$$OPT(i+1) = \max(a[i+1], OPT(i) + a[i+1])$$

ดังนั้นเราจึงสรุปได้ว่าสมการข้างบนเป็นจริงสำหรับค่า i ทุกค่า