Hotel Room 2

(1.5 sec, 512mb)

โจทย์ข้อนี้คือส่วนที่ 3 ของข้อสอบ นิสิตสามารถเลือกทำโจทย์ข้อนี้โดยการเขียนในกระดาษก็ได้ โดยให้ปฏิบัติตามคำข้อบังคับด้านท้ายโจทย์

โรงแรมแห่งหนึ่งมีทั้งหมด N ชั้น (แต่ละชั้นกำกับด้วยหมายเลขตั้งแต่ 1 ถึง N โดยชั้น 1 อยู่ล่างสุด และชั้น N อยู่สูงสุด) ให้ R[i] คือจำนวนห้องในชั้นที่ i เราต้องการบริหารจัดการการจองห้องของโรงแรมนี้ เรามีลูกค้าอยู่ M กลุ่มที่จองเข้ามา ลูกค้าแต่ละกลุ่มจะระบุจำนวนห้องที่ต้องการ และเราจะต้องจัดหาห้องพักที่ยังว่างอยู่ในชั้นต่าง ๆ ให้กับลูกค้าในกลุ่มนั้นทั้งหมด โดยจะพิจารณาลูกค้าทีละกลุ่มตามลำดับที่ได้รับมา

ลูกค้าในแต่ละกลุ่มนั้นต้องการพักในชั้นเดียวกันหากเป็นไปได้ และต้องการห้องที่อยู่ชั้นสูง ๆ ก่อนเสมอ ดังนั้น เมื่อมีลูกค้ากลุ่มหนึ่งแจ้งเข้าพัก (check in) โดยต้องการห้อง p ห้อง เราจะต้องกำหนดห้องพักให้กับลูกค้าตามกฎดังต่อไปนี้

1) หากมีห้องว่างในโรงแรมทั้งหมดน้อยกว่า p ห้อง ให้ปฏิเสธการเข้าพักของลูกค้าไป

2) เราจะหาชั้นที่มีจำนวนห้องว่างน้อยที่สุดที่มากกว่าหรือเท่ากับ p และกำหนดห้องจำนวน p ห้องในชั้นนั้นให้กับลูกค้า (หากมีหลายชั้นตรงตามเงื่อนไขดังกล่าว ให้เลือกชั้นที่อยู่ “สูง” ที่สุด เช่น ให้ p = 3 และ ห้องว่างของชั้น 1 ถึง 5 คือ 2, 3, 4, 3, 5 เราจะต้องเลือกชั้น 4 เพราะมีอยู่สองชั้นที่มีห้องว่างน้อยที่สุดที่มากกว่าหรือเท่ากับ 3 คือชั้น 2 และ ชั้น 4 ทำให้ต้องเลือกชั้นสูงที่สุด คือชั้น 4)

3) หากไม่มีชั้นใดเลยที่มีห้องที่ว่างอยู่ไม่น้อยกว่า p ห้อง เราจะเลือกชั้นที่มีห้องว่างมากสุด (หากมีหลายชั้นที่มีห้องว่างมากสุดเท่ากัน ให้เลือกชั้นที่สูงที่สุด) สมมติให้ชั้นที่เลือกมีห้องว่าง k ห้อง เราจะให้ k ห้องในชั้นนั้นให้กับลูกค้ากลุ่มดังกล่าว แล้วทำการหาห้องว่างจำนวน p-k ห้องให้กับลูกค้าที่เหลืออยู่ของกลุ่มดังกล่าวตามวิธีการเดิมต่อไป

เราต้องเขียนโปรแกรมเพื่อจำลองการทำงานการกำหนดห้องพักนี้

(ข้อความโจทย์จนถึงตรงนี้ เหมือนกับข้อ Hotel Room ใน quiz ที่ผ่านมาแล้ว แต่หลังจากนี้เป็นต้นไป มีสิ่งที่ต้องทำเพิ่มเติม)

โปรแกรมของเราต้องรองรับการแจ้งออกจากห้องพักของลูกค้าด้วย (check out) โดยลูกค้าแต่ละห้อง (ลูกค้าในกลุ่มเดียวกัน อาจจะ check out ไม่พร้อมกันก็ได้) สามารถ check out เมื่อไรก็ได้หลังจากที่แจ้งเข้าพักแล้ว โดย ขณะที่ลูกค้าทั้งกลุ่ม check in นั้น ลูกค้าจะระบุ “หมายเลขการจองของกลุ่ม” มาด้วย และเมื่อลูกค้าที่พักในห้องพักใด ๆ ของกลุ่มนั้น check out ก็จะบอก “หมายเลขการจองของกลุ่ม” พร้อมด้วย “หมายเลขชั้นของห้องพัก” มาด้วย

เมื่อลูกค้าห้องใด check out แล้ว ห้องดังกล่าวก็จะถือเป็นห้องว่าง พร้อมที่จะรับการ check in ต่อทันที

นอกจากนี้ โปรแกรมของเราจะต้องคำนวณ “จำนวนวันรวมของทุกห้องที่พัก” ของลูกค้าทั้งกลุ่มนั้นเข้าพัก โดยเมื่อทำการ check in เราจะถือว่าลูกค้าทั้งกลุ่มนั้น check in ในวันเดียวกันพร้อมกัน และเมื่อลูกค้าแต่ละห้องทำการ check out ก็จะนับวันที่ห้องนั้นถูกใช้ โดยจะถือว่าจำนวนวันที่ใช้ของห้องนั้น นับตั้งแต่วันที่ check in จนถึงวันที่ check out เต็มวัน (เช่น check in วันที่ 5 และ check out วันที่ 7 ถือว่าใช้ห้องนั้น 3 วัน หรือ check in วันที่ 9 และ check out วันที่ 9 ก็ถือว่าใช้ห้อง 1 วัน) ให้ “จำนวนวันรวมของทุกห้องที่พัก” ของกลุ่มใดก็คือผลรวมของจำนวนวันของแต่ละห้องที่กลุ่มนั้นพักนั่นเอง

โปรแแกรมจะต้องตอบผลรวมของจำนวนห้องจำนวนวันของลูกค้าทั้งกลุ่มเมื่อลูกค้าทั้งกลุ่ม check out ครบหมดแล้ว

งานที่ต้องทำ

จงเขียน class hotel ซึ่งต้องมีฟังก์ชันต่อไปนี้

- hotel(vector <int> rooms) เป็น constructor ของคลาสนี้ ซึ่งบอกว่าโรงแรมแห่งนี้มีห้องแต่ละชั้นกี่ห้อง ให้ rooms.size()-1 คือจำนวนชั้น และให้ rooms[i] คือจำนวนห้องในชั้นนั้น ให้ถือว่าตอนเริ่มต้นทุกห้องในโรงแรมว่างทั้งหมด และให้ถือว่าเมื่อเรียก constructor ให้นับว่าเป็นวันที่ 1 (รับประกันว่า rooms[0] = 0)
- void next_day() ฟังก์ชันนี้จะถูกเรียกเพื่อบอกว่า เวลาได้ผ่านไป 1 วันแล้ว
- vector <pair<int, int>> check_in(int group, int num) เป็นฟังก์ชันที่ใช้สำหรับการแจ้งเข้าพัก โดย group คือหมายเลขของกลุ่มลูกค้าที่เข้าพัก และ num คือจำนวนห้องที่ต้องการทั้งหมด รับประกันว่าการเรียก check_in แต่ละครั้งจะมีค่า group ไม่ซ้ำกันเลย และ num มีค่าอย่างน้อย 1

  ฟังก์ชันนี้จะต้องคืนค่าที่ระบุว่าลูกค้ากลุ่มนี้ได้ห้องในชั้นต่าง ๆ อย่างไรบ้าง สมมติให้ค่าที่คืนคือ v เราขอกำหนดว่า v[i].first คือหมายเลขชั้นที่ได้ห้อง และ v[i].second คือจำนวนห้องในชั้นนั้นที่ถูกกำหนดให้ลูกค้ากลุ่มนี้ โดยค่าหมายเลขชั้นใน v ต้องเรียงจากน้อยไปมาก หากเราไม่สามารถหาห้องให้ลูกค้าได้ ให้คืนค่าเป็น vector ว่าง
- int check_out(int group, int floor) เป็นการแจ้งออกจากห้องพักของลูกค้า 1 ห้อง โดย group คือหมายเลขกลุ่มของลูกค้า และ floor คือหมายเลขชั้นของลูกค้า รับประกันว่าการเรียกฟังก์ชันนี้จะเป็นไปตามความถูกต้อง กล่าวคือ group จะเป็นหมายเลข group ของลูกค้าที่ยังเข้าพักอยู่จริง ๆ และ floor ก็จะเป็นหมายเลขชั้นที่มีลูกค้ากลุ่มนั้นยังคงพักอยู่จริง ๆ

  ฟังก์ชันนี้จะต้องคืนผลรวมจำนวนห้องจำนวนวันที่ลูกค้ากลุ่มนั้นพัก เมื่อฟังก์ชันนี้ถูกเรียกโดยลูกค้าห้องสุดท้ายของกลุ่มดังกล่าว (กล่าวคือ เราจะต้องคืนค่าก็ต่อเมื่อลูกค้าทั้งกลุ่ม check out ไปครบหมดแล้ว)

  ในกรณีที่ลูกค้ากลุ่มนั้นยัง check out ไปไม่หมด ให้คืนค่า 0

ตัวอย่าง

เริ่มต้นเรียก hotel({0,2,3,2}) เพื่อบอกว่ามี 3 ชั้น โดยชั้น 1 มี 2 ห้อง ชั้น 2 มี 3 ห้องและชั้น 3 มี 2 ห้อง และเริ่มนับวันที่ 1 ตารางต่อไปนี้แสดงถึงการเรียกฟังก์ชันต่าง ๆ และความหมาย

ฟังก์ชันที่เรียก	ค่าที่คืนกลับ	คำอธิบาย	ห้องว่างที่เหลือ ของชั้น 1, 2, 3 หลังจบฟังก์ชัน
hotel({0,2,3,2})	-	constructor เริ่มวันที่ 1	[2, 3, 2]
check_in(5,3)	{ {2,3} }	ลูกค้ากลุ่ม 5 ต้องการ 3 ห้อง โดยได้ห้องชั้น 2 ทั้งหมด	[2, 0, 2]
next_day()	-	เปลี่ยนเป็นวันที่ 2	
check_in(9,3)	{ {1,1}, {3,2} }	ลูกค้ากลุ่ม 9 ต้องการ 3 ห้อง ได้ชั้น 3 2 ห้อง และชั้น 1 อีก 1 ห้อง	[1,0,0]
check_out(5,2)	0	ลูกค้ากลุ่ม 5 ชั้น 2 check out ออกไป 1 ห้อง (ห้องนี้นับเป็น 2 วัน)	[1, 1, 0]

check_out(5,2)	0	ลูกค้ากลุ่ม 5 ชั้น 2 check out ออกไปอีก 1 ห้อง (ห้องนี้นับ เป็น 2 วัน รวมเป็น 4 วัน)	[1, 2, 0]
next_day()	-	เปลี่ยนเป็นวันที่ 3	
check_out(5,2)	7	ลูกค้ากลุ่ม 5 ชั้น 2 check out ออกไปอีก 1 ห้อง (ห้องนี้นับ เป็น 3 วัน รวมเป็น 7 วัน) กลุ่ม 5 check out หมดแล้ว ดังนั้นต้องคืนค่า 7	[1, 3, 0]
check_out(9,1)	0	ลูกค้ากลุ่ม 9 ชั้น 1 check out (ห้องนี้นับเป็น 2 วัน)	[2, 3, 0]
check_in(1,2)	{ {1,2} }	ลูกค้ากลุ่ม 1 ต้องการ 1 ห้อง ได้ชั้น 1 2 ห้อง	[0, 3, 0]
check_out(1,1)	0	ลูกค้ากลุ่ม 1 ชั้น 1 check out (ห้องนี้นับเป็น 1 วัน)	[1, 3, 0]
check_out(1,1)	2	ลูกค้ากลุ่ม 1 ชั้น 1 check out ออกไปอีก 1 ห้อง (ห้องนี้นับ เป็น 1 วัน รวมเป็น 2 วัน) กลุ่ม 1 check out หมดแล้ว ดังนั้นต้องคืนค่า 2	[2, 3, 0]
next_day()	-	เปลี่ยนเป็นวันที่ 4	
next_day()	-	เปลี่ยนเป็นวันที่ 5	
next_day()	-	เปลี่ยนเป็นวันที่ 6	
check_out(9,3)	0	ลูกค้ากลุ่ม 9 ชั้น 3 check out (ห้องนี้นับเป็น 5 วัน รวมเป็น 7 วัน)	[2, 3, 1]
next_day()	-	เปลี่ยนเป็นวันที่ 7	
check_out(9,3)	13	ลูกค้ากลุ่ม 9 ชั้น 3 check out ออกไปอีก 1 ห้อง (ห้องนี้นับเป็น 6 วัน รวมเป็น 13 วัน) กลุ่ม 9 check out หมดแล้ว ดังนั้นต้องคืนค่า 13	[2, 3, 2]

คำอธิบายฟังก์ชัน main

- บรรทัดแรกประกอบด้วยจำนวนเต็ม N ซึ่งระบุจำนวนชั้นของโรงแรม (1 < = N <= 100,000)
- บรรทัดที่สองประกอบด้วยจำนวนเต็ม N จำนวนคือ R[1] ถึง R[N] ซึ่งระบุจำนวนห้องพักที่ว่างอยู่ในแต่ละชั้น (1 <= R[i] <= 1000)

main จะสร้าง hotel h ด้วยค่า R แล้วจะอ่านคำสั่งมาทีละ 1 บรรทัด แต่ละบรรทัดคือคำสั่งที่จะเรียกใช้ฟังก์ชันต่าง ๆ ของ h ในรูปแบบต่อไปนี้

- q เป็นการบอกให้ main หยุดทำงาน

- n เป็นการเรียก h.next_day()
- i A B เป็นการเรียก h.check_in(A, B) แล้วจะพิมพ์ค่าที่ฟังก์ชันนั้นคืนมาออกมาทางหน้าจอ (1 <= A <= 10^9 และ 1 <= B <= 1,000)
- o C D เป็นการเรียก h.check_out(C, D) แล้วจะพิมพ์ค่าที่ฟังก์ชันนั้นคืนมาออกมาทางหน้าจอ (1 <= C <= 10^9 และ 1 <= D <= N)

ตัวอย่างในตารางข้างต้นจะมี input ของ main เป็นดังนี้

ชุดข้อมูลทดสอบ

อัตราส่วน	ขอบเขตข้อมูลเพิ่มเติม	ลักษณะ
15%	group <= 100,000	ไม่มีการเรียก next_day หรือ check_out เลย (กล่าวคือคลาสนี้แค่ทำงานทำงานเหมือนข้อ hotel ใน quiz ก็พอ)
10%	N <= 100 และ group <= 100,000	การเรียก check_in แต่ละครั้งมีค่า num เป็น 1 เสมอ
10%	N <= 100 และ group <= 100,000	เมื่อเรียก check_in แล้ว จะไม่เรียก check_in อีกจนกว่ากลุ่มที่ check in แล้ว จะ check out ออกไปหมด
20%	-	การเรียก check_in แต่ละครั้งมีค่า num เป็น 1 เสมอ
15%	-	เมื่อเรียก check_in แล้ว จะไม่เรียก check_in อีกจนกว่ากลุ่มที่ check in แล้ว จะ check out ออกไปหมด
30%	-	-