



แจกจ่ายของขวัญ

1 second, 256 megabytes

By [Mingyuanz](#)

ในคืนของวัน Christmas Eve ของทุกปีนั้น Santa Claus จะจัดส่งของขวัญให้กับผู้คนทั่วโลก แต่เนื่องจากประชากรโลกกำลังเพิ่มขึ้นอย่างต่อเนื่อง ทำให้ Santa Claus ไม่สามารถนำของขวัญไปส่งให้ทุกคนได้ทันเวลาภายในคืนเดียว ได้แล้ว Santa Claus จึงเปลี่ยนระบบการจัดส่งของขวัญใหม่ เป็นการจัดส่งของขวัญของทุกคนในแต่ละเมืองไปร่วมกันไว้ที่จุดเดียวในเมืองนั้นก่อน แล้วให้แต่ละเมืองจัดการแจกจ่ายของขวัญต่อจากนั้นเอง ซึ่ง เพราะความหลากหลายของชานด้านนี้ เขาจึงจะไปส่งไว้ที่ตึกที่สูงสุดของเมืองเสมอ

ณ เมืองแห่งหนึ่ง DevJ และ gjoop ได้รับหน้าที่เป็นคนแจกจ่ายของขวัญจากตึกที่สูงที่สุดของเมืองไปให้ทุกคนในเมือง โดยเมืองที่ทั้งสองคนอยู่มีตึกอยู่ทั้งหมด n ตึก จะเรียกว่าตึกที่ i ถึงตึกที่ n เรียงกันจากซ้ายไปขวา แต่ละตึกมีความสูงเป็นจำนวนเต็มที่อยู่ระหว่าง 1 ถึง n และไม่มีตึกใดที่มีความสูงเท่ากันเลย

พวกราคาต้องแจกจ่ายของขวัญรวมทั้งหมด q ครั้ง ในครั้งที่ i ตอนที่เริ่มการแจกจ่าย DevJ และ gjoop จะยืนอยู่บนตึกที่ a_i และ b_i ตามลำดับ เป้าหมายของทั้งสองคนคือ การกระโดดตามตึกขึ้นไปเจอกันบนตึกที่มีความสูง n เพื่อไปเอาของขวัญกลับมาแจกจ่าย โดยเมื่อคนใดคนหนึ่งอยู่บนตึกที่ j จะสามารถเลือกกระโดดได้อย่างมากสองทางเลือกดังนี้

- กระโดดไปบนตึกที่ใกล้ที่สุดทางด้านซ้ายของตึกที่ j ที่มีความสูงมากกว่า j (ถ้ามี)
- กระโดดไปบนตึกที่ใกล้ที่สุดทางด้านขวาของตึกที่ j ที่มีความสูงมากกว่า j (ถ้ามี)

แต่มีเงื่อนไขว่า ถ้าคนหนึ่งมีตัวเลือกกระโดดทั้งสองทาง จะสามารถเลือกตัวเลือกที่ตีกปลายทางหลังการกระโดดมีความสูงมากกว่าอีกด้วย นี่ได้แก่ตัวเลือกหนึ่งที่ได้ก่อต่อเมื่อคนนั้นมีพลังงานเหลืออยู่ตั้งแต่ 1 หน่วยขึ้นไป และหลังจากเลือกกระโดดไปทางนั้นแล้ว คนนั้นจะเสียค่าพลังงานไป 1 หน่วย

ซึ่งเมื่อคนใดก็ตาม กระโดดรอดห่วงสองตึกใดๆ จะมีค่าความเห็นอย่างการกระโดดรอนั้นเท่ากับ ผลลัพธ์ค่า XOR ของค่าความต้านอากาศด้านบนตึกทั้งหมดที่อยู่ระหว่างทั้งสองตึกนั้น (รวมถึงตึกต้นทางและปลายทางด้วย) คุณตัวอย่างต่างของความสูงของตึกต้นทางและปลายทางนั้น

ก่อนเริ่มการแจกจ่ายในแต่ละครั้ง DevJ และ gjoop จะมีค่าพลังงานอยู่ 0 หน่วยเสมอ และพวกราคามีแค่นี้คือ ชุดกำลังรวมทั้งหมด k แห่ง ที่แต่ละแห่งจะเพิ่มค่าพลังงานให้กับคนที่กินไป 1 หน่วย (ไม่จำเป็นต้องกินครบทั้ง k แห่ง ก็ได้) และในเมื่อพวกราคาต้องแจกจ่ายของขวัญหลายครั้งมากๆ ในแต่ละครั้งที่แจกจ่าย จึงพยายามกระโดดขึ้นไปเอาร่องของขวัญ โดยให้ผลบวกของค่าความเห็นอย่างทั้งสองคนมีค่าน้อยที่สุดเท่าที่จะทำได้ คุณซึ่งเป็นผู้ช่วยงาน DevJ และ gjoop ณ ตอนนั้น จึงต้องหาวิธีการแบ่งแคนดี้ ken ชุดกำลังให้เข้าทั้งคู่ ที่ทำให้ผลบวกค่าความเห็นอย่างในการกระโดดขึ้นรวมทุกครั้งของพวกราคานั้น มีค่าน้อยที่สุด และหาผลบวกค่าความเห็นอย่างที่น้อยที่สุดเท่าที่เป็นไปได้



ถ้ามีวิธีการแบ่งแคนดี้เคนช์กำลังได้หลายแบบที่ถูกต้อง ให้ตอบวิธีที่เหลือแคนดี้เคนช์กำลังอยู่มากที่สุด และถ้ายังมีหลายวิธีอีก ให้ตอบวิธีที่ DevJ ได้กินแคนดี้เคนช์กำลังมากที่สุด

โจทย์ จะเขียนโปรแกรมเพื่อหาวิธีการแบ่งแคนดี้เ肯ช์กำลังให้ทั้งสองคนที่เหมาะสมที่สุด และหาค่าความเหลื่อมล้ำที่น้อยที่สุดที่เป็นไปได้เมื่อแบ่งตามนั้นแล้ว ในการแจกจ่ายแต่ละครั้ง

ข้อมูลนำเข้า

บรรทัดแรก จำนวนเต็ม 2 จำนวนได้แก่ $n \ q$:

โดยจำนวนเต็ม n คือ จำนวนติกที่มีอยู่ในเมืองแห่งนี้ ($1 \leq n \leq 2 \cdot 10^5$)

โดยจำนวนเต็ม q คือ จำนวนครั้งที่ DevJ และ gjoop ต้องแจกจ่ายของขวัญ ($1 \leq q \leq 2 \cdot 10^5$)

บรรทัดที่ 2 จำนวนเต็ม n ตัว, h_1, h_2, \dots, h_n :

โดย h_i ($1 \leq i \leq n$) คือ ค่าความสูงของตึกที่ i ในเมืองแห่งนี้ ($1 \leq h_i \leq n$)

บรรทัดที่ 3 จำนวนเต็ม n ตัว, r_1, r_2, \dots, r_n :

โดย r_i ($1 \leq i \leq n$) คือ ค่าความต้านทานอากาศด้านบนตึกที่ i ในเมืองแห่งนี้ ($0 \leq r_i \leq 2^{30} - 1$)

บรรทัดที่ 4 ถึง $q + 3$ บรรทัดที่ $i + 3$ ($1 \leq i \leq q$) มีจำนวนเต็ม 3 จำนวนได้แก่ $a_i \ b_i \ k_i$:

โดยจำนวนเต็ม a_i คือ หมายเลขติกที่ DevJ อยู่ก่อนเริ่มการแจกจ่ายครั้งที่ i ($1 \leq a_i \leq n$)

โดยจำนวนเต็ม b_i คือ หมายเลขติกที่ gjoop อยู่ก่อนเริ่มการแจกจ่ายครั้งที่ i ($1 \leq b_i \leq n$)

โดยจำนวนเต็ม k_i คือ จำนวนแคนดี้เคนช์กำลังทั้งหมดที่มีอยู่ ก่อนเริ่มการแจกจ่ายครั้งที่ i ($1 \leq k_i \leq 30$)

ข้อมูลส่งออก

มี $2q$ บรรทัด: โดยทุกๆ ค่า i ($1 \leq i \leq q$)

บรรทัดที่ $2i - 1$ จำนวนเต็ม 2 จำนวนได้แก่ $x_i \ y_i$: จำนวนแคนดี้เคนช์กำลังที่แบ่งให้ DevJ และ gjoop ตามลำดับ ก่อนเริ่มการแจกจ่ายครั้งที่ i

บรรทัดที่ $2i$ จำนวนเต็ม ans_i : ผลรวมค่าความเหลื่อมล้ำของ DevJ และ gjoop ที่น้อยที่สุดเท่าที่เป็นไปได้ หลังการแจกจ่ายครั้งที่ i



ตัวอย่างข้อมูลนำเข้าและข้อมูลส่งออก

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
6 4	0 0
4 1 6 2 5 3	37
4 3 2 7 5 1	1 0
2 4 0	11
2 4 1	1 0
2 4 2	11
2 2 1	1 0
	36

คำอธิบายตัวอย่าง

ในการแยกจ่ายครั้งที่ 1

ก่อนแยกจ่าย DevJ อยู่บันตึกที่ 2 และ njoop อยู่บันตึกที่ 4 และมีแม่นดี้เคนชูกำลังเลย

DevJ จะเดินทางจากตึกที่ 2 \rightarrow 1 \rightarrow 3 ซึ่งมีค่าความเห็น้อยรวม $(3 \oplus 4) \cdot (4 - 1) + (4 \oplus 3 \oplus 2) \cdot (6 - 4) = 31$

njoop จะเดินทางจากตึกที่ 4 \rightarrow 5 \rightarrow 3 ซึ่งมีค่าความเห็น้อยรวม $(7 \oplus 5) \cdot (5 - 2) + (5 \oplus 7 \oplus 2) \cdot (6 - 5) = 6$

เพราะฉะนั้น ผลรวมค่าความเห็น้อยรวมของทั้งสองคนจะเท่ากับ 37

ในการแยกจ่ายครั้งที่ 2

ก่อนแยกจ่าย DevJ อยู่บันตึกที่ 2 และ njoop อยู่บันตึกที่ 4 และมีแม่นดี้เคนชูอยู่ 1 แห่ง

DevJ จะกินแม่นดี้เคนชู 1 แห่ง และเดินทางจากตึกที่ 2 \rightarrow 3 ซึ่งมีค่าความเห็น้อยรวม $(3 \oplus 2) \cdot (6 - 1) = 5$

njoop จะเดินทางจากตึกที่ 4 \rightarrow 5 \rightarrow 3 ซึ่งมีค่าความเห็น้อยรวม $(7 \oplus 5) \cdot (5 - 2) + (5 \oplus 7 \oplus 2) \cdot (6 - 5) = 6$

เพราะฉะนั้น ผลรวมค่าความเห็น้อยรวมของทั้งสองคนจะเท่ากับ 11

ในการแยกจ่ายครั้งที่ 3

ก่อนแยกจ่าย DevJ อยู่บันตึกที่ 2 และ njoop อยู่บันตึกที่ 4 และมีแม่นดี้เคนชูอยู่ 2 แห่ง

การแบ่งแม่นดี้เคนชู 2 แห่ง และกระโดดจะมีรูปแบบเหมือนกับการแยกจ่ายครั้งที่ 2 ถึงแม้ว่าแม่นดี้เ肯ชูจะอยู่ 2

แห่ง เนื่องจากรูปแบบนี้มีผลรวมค่าความเห็น้อยที่น้อยที่สุดแล้ว และเป็นวิธีที่เหลือแม่นดี้เคนชูอยู่มากที่สุด

ในการแยกจ่ายครั้งที่ 4

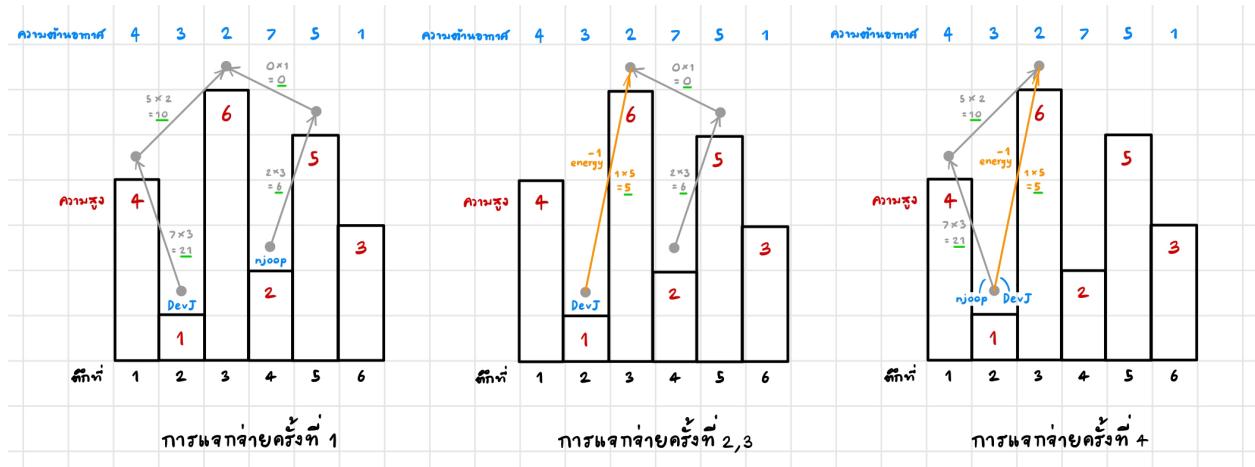
ก่อนแยกจ่ายทั้ง DevJ และ njoop อยู่บันตึกที่ 2 และมีแม่นดี้เ肯ชูอยู่ 1 แห่ง

DevJ จะกินแม่นดี้เ肯ชู 1 แห่ง และเดินทางจากตึกที่ 2 \rightarrow 3 ซึ่งมีค่าความเห็น้อยรวม $(3 \oplus 2) \cdot (6 - 1) = 5$

njoop จะเดินทางจากตึกที่ 2 \rightarrow 1 \rightarrow 3 ซึ่งมีค่าความเห็น้อยรวม $(3 \oplus 4) \cdot (4 - 1) + (4 \oplus 3 \oplus 2) \cdot (6 - 4) = 31$

ที่ต้องทำเช่นนี้ เนื่องจากรูปแบบนี้มีผลรวมค่าความเห็น้อยที่น้อยที่สุดแล้ว มีแม่นดี้เ肯ชูอยู่มากที่สุดแล้ว

และเป็นวิธีที่ทำให้ DevJ ได้กินแม่นดี้เ肯ชูมากที่สุด



ภาพประกอบตัวอย่างการแจกจ่าย

การให้คะแนน

คะแนนเต็ม 300 คะแนน มี 9 Subtasks

5 คะแนน: $h_i < h_{i+1}$ ($1 \leq i < n$)

10 คะแนน: ไม่มีค่า i ใดที่ $h_{i-1} > h_i$ พร้อมกับ $h_i < h_{i+1}$ ($1 < i < n$)

15 คะแนน: $k = 0, q = 1$

30 คะแนน: $k = 0$

30 คะแนน: $q = 1$

15 คะแนน: $n \leq 50$

30 คะแนน: $n \leq 500$

45 คะแนน: $n \leq 10000$

120 คะแนน: ไม่มีเงื่อนไขเพิ่มเติม

**จะได้คะแนนในแต่ละกลุ่มชุดทดสอบ ก็ต่อเมื่อโปรแกรมให้ผลลัพธ์ถูกต้องในชุดทดสอบอย่างทั้งหมด

คำแนะนำ

หากใช้ภาษา C++ และนำให้พิมคำสั่ง `cin.tie(nullptr)->sync_with_stdio(false);`
และให้ใช้ '\n' แทน endl เช่น `cout << "Hello World" << '\n';`

หากใช้ภาษา C/C++ และนำให้ใช้คอมไพล์เตอร์ GNU G++17 7.3.0 ในการ Submit Code