

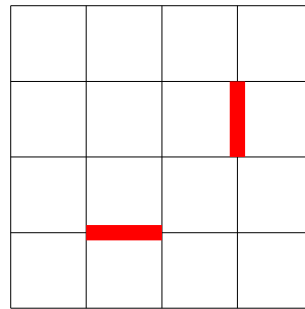
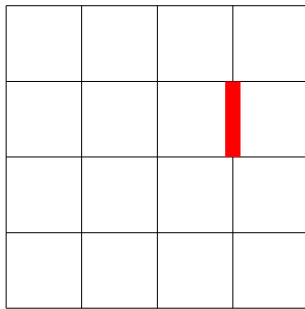
รถติด (2)

(1.5 second, 128 MB, interactive)

ในเมืองมีถนนเป็นตารางกริดขนาด $N \times N$ โดยมีถนนแนวตั้ง N เส้นและถนนแนวนอน N เส้น ($2 \leq N \leq 100$) ถนนในแต่ละแนวจะมีหมายเลขเรียงไปตั้งแต่ 1 ถึง N (นั่นคือมีถนนแนวตั้ง 1 ถึงถนนแนวตั้ง N และมีถนนแนวนอน 1 ถึงถนนแนวนอน N) ถนนทุกเส้นเดินทางได้สองทิศทาง ถนนเหล่านี้มีจุดตัดของถนนเป็นแยกจำนวน $N \times N$ แยก แต่ละแยกจะระบุด้วยคู่ลำดับประกอบไปด้วยหมายเลขของถนนแนวตั้งและถนนแนวนอนที่ตัดกัน (ดูตัวอย่างด้านล่าง) ถนนที่ตัดกันจะแบ่งถนนเป็นช่วงถนนย่อย จำนวน $2 \times (N-1) \times N$ ช่วงที่เชื่อมระหว่างแยกสองแยกที่ติดกัน

ขณะนั้นมีช่วงถนนย่อยจำนวน K ช่วง ($K \leq 2$) ที่มีรถติด ในการเดินทางนั้น ช่วงของถนนย่อยทั่วไปจะใช้เวลาเดินทาง 1 หน่วย ส่วนช่วงถนนย่อยที่มีรถติดจะใช้เวลา 2 หน่วย คุณสามารถสอบถามระยะเวลาเดินทางที่สั้นที่สุดระหว่างแยกสองแยกใด ๆ ในเมืองนี้ เพื่อหาว่าช่วงถนนย่อยที่เชื่อมระหว่างแยกใดที่มีรถติด คุณจะถามคำถามได้ไม่เกิน 222 ครั้ง

พิจารณาตัวอย่างสองตัวอย่างต่อไปนี้ (ที่ $N = 5$)



ให้ถนนแนวตั้งเริ่มนับหมายเลข 1 จากซ้ายไปขวา และถนนแนวนอนเริ่มนับหมายเลข 1 จากล่างขึ้นบน พิจารณาตัวอย่างแรกทางซ้ายมือที่ $K = 1$ ช่วงถนนย่อยที่รถติดเชื่อมระหว่างแยก (4,3) กับ (4,4) ถ้าสอบถามระยะเวลาสั้นที่สุดในการเดินทางจากแยก (3,4) ไปยัง (4,3) จะได้คำตอบว่าใช้เวลา 2 หน่วย แต่ถ้าสอบถามระยะเวลาสั้นที่สุดในการเดินทางจากแยก (4,5) ไปยัง (4,3) จะได้คำตอบว่าใช้เวลา 3 หน่วย

ในตัวอย่างที่สองทางขวามือ ในกรณีนี้ $K = 2$ ถ้าสอบถามเวลาสั้นที่สุดจากแยก (1,1) ไปยังแยก (5,5) จะได้คำตอบว่า 8 แต่ถ้าสอบถามเวลาสั้นที่สุดจากแยก (2,2) ไปยัง (3,2) จะได้คำตอบว่า 2 หน่วย เป็นต้น

ไลบรารี – ดาวน์โหลดไลบรารีทดสอบที่ <https://theory.cpe.ku.ac.th/~jittat/lib/traffic/>

โปรแกรมของคุณจะติดต่อกับไลบรารี ให้ include "traffic.h" ที่ต้นโปรแกรม และในตอนคอมไพล์ให้นำ trafficlib.cpp ไปคอมไพล์ด้วย จะมีฟังก์ชันต่อไปนี้ให้ใช้งาน ห้ามโปรแกรมอ่านข้อมูลเข้าและเขียนข้อมูลออกเอง

- `traffic_init(int* N, int* K)` – จะส่งค่า N และ K กลับมาทางตัวแปร N และ K ต้องเรียกฟังก์ชันนี้ก่อนเรียกฟังก์ชันอื่น
- `traffic_query(int x1, int y1, int x2, int y2)` – จะคืนระยะเวลาที่สั้นที่สุดในการเดินทางจากแยก (x1,y1) ไปยังแยก (x2,y2) เรียกได้ไม่เกิน 222 ครั้ง
- `traffic_report(int ax1, int ay1, int ax2, int ay2, int bx1, int by1, int bx2, int by2)` – ใช้ในการตอบว่า

ถนนย่อยใดที่มีรถติด โดยถนนย่อยแรกระบุด้วยแยก (ax1,ay1) – (ax2,ay2) และถนนย่อยที่สองระบุด้วยแยก (bx1,by1) – (bx2,by2) ลำดับในการตอบจะเป็นอย่างไรก็ตาม ถ้า K = 1 ให้ทั้งพารามิเตอร์สี่ตัวหลังเป็น 0 ฟังก์ชันนี้เรียกได้ครั้งเดียว เมื่อเรียกฟังก์ชันนี้แล้วโปรแกรมสามารถจบการทำงานได้เลย

เวลาการทำงานของโปรแกรมของคุณจะรวมเวลาการทำงานของไลบรารีด้วย เวลาการทำงานรวมของไลบรารีโดยทั่วไปแล้วจะไม่เกิน 0.5 วินาที

ปัญหาย่อย

- ปัญหาย่อย 1 (10%): $N \leq 10$
- ปัญหาย่อย 2 (40%): $K = 1$
- ปัญหาย่อย 3 (50%): $K = 2$

ตัวอย่างการทำงาน

แสดงการเรียกใช้ไลบรารีในกรณีตัวอย่างแรก (รูปซ้ายมือ)

การเรียกฟังก์ชัน	ผลลัพธ์
<code>traffic_init(&n, &k);</code>	$n = 5, k = 1$
<code>traffic_query(3,4,4,3);</code>	2
<code>traffic_query(4,5,4,3);</code>	3
<code>traffic_report(4,3,4,4, 0,0,0,0);</code>	คำตอบถูกต้อง

แสดงการเรียกใช้ไลบรารีในกรณีตัวอย่างที่สอง (รูปขวามือ)

การเรียกฟังก์ชัน	ผลลัพธ์
<code>traffic_init(&n, &k);</code>	$n = 5, k = 2$
<code>traffic_query(1,1,5,5);</code>	8
<code>traffic_query(2,2,3,2);</code>	2
<code>traffic_report(3,2,2,2, 4,4,4,3);</code>	คำตอบถูกต้อง

การใช้งานไลบรารีตัวอย่าง

ไลบรารีตัวอย่างจะอ่านข้อมูลนำเข้าระบุในรูปแบบต่อไปนี้

- บรรทัดแรก N และ K
- อีก K บรรทัดระบุข้อมูลถนนย่อยที่รถติด แต่ละช่วงระบุด้วย $x1\ y1\ x2\ y2$

ข้อมูลส่งออกจะตอบว่า correct หรือ incorrect เมื่อเรียก `traffic_report` หรือถ้าเรียกใช้ฟังก์ชันผิด จะตอบ incorrect สามารถแก้ไลบรารีตัวอย่างเพื่อทดสอบได้

ในการตรวจที่เกรดเดอร์อาจจะมีการใช้ไลบรารีอีกชุดหนึ่ง

ข้อมูลนำเข้าไลบรารีตัวอย่างแรก	ข้อมูลนำเข้าไลบรารีตัวอย่างสอง
5 1 4 4 4 3	5 2 4 4 4 3 2 2 3 2