

เมืองคนบาป 1sec, 256mb

ในเมืองที่แบ่งเป็นตารางขนาด R แถว C คอลัมน์ ($0 < R, C \leq 1,000$) มีการแบ่งเขตของแก๊งค์มาเพียอยู่ N แก๊งค์ ($0 < N \leq 10$) แต่ละแก๊งค์กำหนดด้วยหมายเลข 0 ถึง $N-1$ อยู่มาวันหนึ่งมหาเมฟท์รอร์ อยากจะเข้ามายึดเมืองดังกล่าว มหาเมฟท์จึงส่งนักสืบมา M คน ($0 < M \leq 100$) แต่ละคนกำหนดด้วยหมายเลข 0 ถึง $M-1$ มาประจำ ณ ช่องต่าง ๆ นักสืบเหล่านี้สามารถสืบทราบได้ว่าช่องที่ตัวเองอยู่นั้นเป็นเขตของมาเฟียใด มหาเมฟท์ได้กำหนดเป้าหมายให้นักสืบแต่ละคนว่าจะต้องสืบข้อมูลของแก๊งค์ใด นักสืบแต่ละคนนั้นจะต้องเดินไปยังช่องที่เป็นของแก๊งค์ที่เป็นเป้าหมายของตัวเองให้ได้

ภัทธอร์ไม่รู้ว่แต่ละช่องนั้นเป็นของแก๊งค์ใด คุณต้องช่วยภัทธอร์โดยพยายามพานักสืบเหล่านี้วิ่งไปยังช่องที่เป็นเขตของเป้าหมายของตัวเองให้ได้ ภัทธอร์ก็ไม่อยากให้นักสืบของตัวเองทำงานหนัก (ประเด็นไม่ใช่เรื่องเงิน มหาเมฟท์มีเงินอนันต์ แต่การใช้ลูกน้องมากเกินไปเป็นเรื่องที่ไม่ดี) คุณต้องหาวิธีการเดินที่ใช้จำนวนช่องในการเดินรวมกันให้น้อยที่สุดเท่าที่เป็นไปได้

งานของคุณ

ให้เขียนโปรแกรมรับข้อมูลขนาดของเมือง, จำนวน, ตำแหน่ง และ เป้าหมายของนักสืบ แล้วทำการสั่งให้นักสืบเดินไปยังช่องต่าง ๆ โดยที่เมื่อคุณทำงานเสร็จ นักสืบทุกคนจะต้องไปอยู่ยังช่องที่เป็นของแก๊งค์ที่เป็นเป้าหมายของตัวเอง

โจทย์ข้อนี้เป็นโจทย์ที่เรียกใช้ API ที่คุณจะต้องเขียน จงเขียนฟังก์ชันต่อไปนี้

- `void solve(int R, int C, int N, int M, int x[], int y[], int c[], int target[])`
รับข้อมูลของแผนที่เมือง และตำแหน่งเริ่มต้นของนักสืบ โดย R, C, N นั้นระบุขนาดและจำนวนแก๊งค์ มีนักสืบอยู่ M คนโดยคนที่ i อยู่ ณ ตำแหน่ง $x[i], y[i]$ และตำแหน่ง $x[i], y[i]$ นั้นเป็นของแก๊งค์ $c[i]$ โดยนักสืบคนนั้นจะต้องไปสืบแก๊งค์ $target[i]$

ในฟังก์ชันดังกล่าว คุณสามารถเรียกใช้ฟังก์ชันต่อไปนี้ได้

- `int testmove(int x1, int y1, int x2, int y2, int x[], int y[])` ฟังก์ชันนี้จะคำนวณการเดินจากตำแหน่ง $(x1, y1)$ ไปยังตำแหน่ง $(x2, y2)$ โดยจะคืนค่าจำนวนช่องที่เดินผ่านทั้งหมด (ไม่รวม $(x1, y1)$ แต่รวม $(x2, y2)$ ด้วย) และตำแหน่งช่องที่เดินผ่านทั้งหมด จะอยู่ในตัวแปร $x[i], y[i]$ แต่การเรียกฟังก์ชันนี้จะไม่ใช่การเดินจริง ๆ โดยมีเงื่อนไขคือ $0 \leq idx < M, 0 \leq x1, x2 < C, 0 \leq y1, y2 < R$ ขนาดของอาร์เรย์ x, y ต้องมีจำนวนไม่ต่ำกว่า $(R+C)$ ถ้าสั่งผิดเงื่อนไขโปรแกรมจะหยุดทำงานทันที
- `int move(int idx, int nx, int ny, int x[], int y[], int c[])` สั่งให้นักสืบหมายเลข idx เดินไปยังตำแหน่ง (nx, ny) ฟังก์ชันนี้จะคืนค่าจำนวนช่องที่เดินผ่าน และช่องที่เดินผ่านทั้งหมด (ช่องที่เดินผ่านนั้นจะคำนวณด้วยวิธีเดียวกันกับ `testmove`) นอกจากนี้ $c[i]$ จะระบุแก๊งค์ที่ควบคุมช่อง $(x[i], y[i])$ ด้วย การเรียกฟังก์ชันนี้จะใช้เงินเท่ากับค่าที่

คืนกลับมา (จำนวนช่องที่เดินผ่านทั้งหมด) โดยมีเงื่อนไขคือ $0 \leq \text{idx} < M$, $0 \leq \text{nx} < C$, $0 \leq \text{ny} < R$ ขนาดของอาร์เรย์ x, y, c ต้องมีจำนวนไม่ต่ำกว่า $(R+C)$ ถ้าสิ่งผิดเงื่อนไขโปรแกรมจะหยุดทำงานทันที

หลังจากจบการทำงานของ solve แล้ว นักสืบทุกคนจะต้องไปอยู่ยังช่องที่เป็นของแองค์ที่เป็นเป้าหมายของตัวเอง

วิธีการคำนวณช่องที่เดินผ่านจากช่อง (x_1, y_1) ไปยังช่อง (x_2, y_2) คือ Bresenham's Line Drawing Algorithm

มีโครงสำหรับเขียนอยู่ในแฟ้ม move.cpp ที่สามารถดาวน์โหลดพร้อมกับ grader.cpp ที่เป็นโปรแกรมหลักและไฟล์อื่น ๆ ที่จำเป็นต้อง include

ข้อกำหนดต่าง ๆ

- รับประกันว่าตอนเริ่มต้นไม่มีนักสืบคนคู่ใดอยู่ในช่องเดียวกัน
- เราสามารถสั่งให้นักสืบสองคนหรือมากกว่าเดินไปยังช่องเดียวกันได้ (ช่องหนึ่งช่องสามารถมีนักสืบอยู่ได้หลายคน)
- ช่องที่เป็นจุดเริ่มต้นของนักสืบนั้นจะถูกล้อมในทิศบนล่างซ้ายขวาด้วยช่องว่างที่เป็นของแองค์เดียวกันเสมอ และเราจะถือว่าช่องดังกล่าวนั้นเป็นของแองค์ที่ล้อมรอบอยู่เช่นกัน

API สำหรับเขียน

รูปแบบของข้อมูลป้อนเข้าที่ grader.cpp ตัวอย่างจะอ่านจาก standard input เป็นแฟ้มภาพประเภท png โดยภาพดังกล่าวจะแทนช่องต่าง ๆ ของเมือง สีใน PNG จะอยู่ในรูปแบบ RGB ค่าสีที่เป็นไปได้มีอยู่สองประเภทคือ

1. สีเทา ($R = G = B$) โดยที่ $R = 16 * k$; ซึ่งจะระบุช่องว่างที่เป็นของแองค์ k
2. สีโทนฟ้า ($R = 16 * k$, $G = 1$, $B = 255$) ซึ่งจะระบุตำแหน่งเริ่มต้นของนักสืบที่ต้องไปยังช่องที่เป็นของแองค์ k

หมายเลขของนักสืบนั้นจะเรียงจากตำแหน่งที่เป็นนักสืบจากบนลงล่าง ถ้าอยู่ในแถวเดียวกันจะเรียงจากซ้ายไปขวา

คุณสามารถแก้ไขแฟ้มนี้ได้ตามต้องการ

โปรแกรมจะพิมพ์ค่าที่คำสั่งการเดินทั้งหมดออกมาทาง standard output