



International Olympiad in Informatics 2012

23-30 September 2012

Sirmione - Montichiari, Italy

Competition tasks, day 2: Leonardo's art and science

city

ไทย — 1.2

เมืองในอุดมคติ

เช่นเดียวกับนักวิทยาศาสตร์และศิลปินชาวอิตาลีคนอื่นในยุคเดียวกันอีกหลายคน ลีโอนาร์โดคั้งโคล์ในการวางแผนเมืองและการออกแบบชุมชนเมือง เขาตั้งเป้าที่จะสร้างแบบจำลองเมืองในฝัน: สะดวกสบาย กว้างขวางและสมเหตุผลในการใช้ทรัพยากร ห่างไกลจากเมืองที่อัดอัดคับแคบของยุคกลาง

เมืองในอุดมคติ

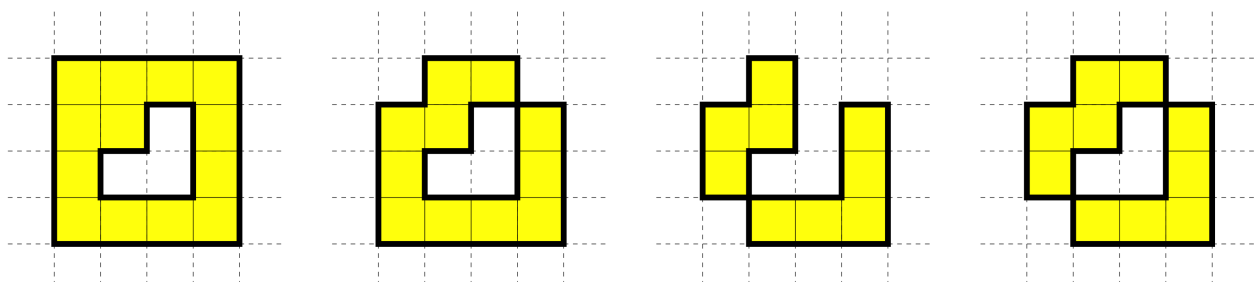
เมืองถูกสร้างด้วย N บล็อกวางอยู่บนกริดในจินตนาการที่มีความกว้าง/ยาวเป็นอนันต์ กำหนดให้ช่อง (i, j) มีช่องที่อยู่ติดกันคือ $(i-1, j)$ $(i+1, j)$ $(i, j-1)$ และ $(i, j+1)$ บล็อกแต่ละก้อนเมื่อถูกวางไว้บนกริดจะพอดีกับหนึ่งช่อง บล็อกสามารถวางบนช่อง (i, j) ได้ ถ้าและเพียงแต่ถ้า $1 \leq i, j \leq 2^{31} - 2$ เราจะใช้ลำดับของช่องอ้างอิงถึงบล็อกที่อยู่บนช่องนั้น บล็อกสองบล็อกอยู่ติดกันถ้ามันถูกวางอยู่บนช่องที่ติดกัน ในเมืองในอุดมคติ บล็อกทั้งหมดจะต้องเชื่อมต่อกันโดยไม่มี "รู" อยู่ภายใน กล่าวคือช่องต้องเป็นไปตามเงื่อนไขดังต่อไปนี้

สำหรับช่อง "ว่าง" สองช่องใด ๆ จะมีลำดับของช่อง "ว่าง" ที่อยู่ติดกันอย่างน้อยหนึ่งลำดับเชื่อมต่อสองช่องดังกล่าว

สำหรับช่อง "ไม่ว่าง" สองช่องใด ๆ จะมีลำดับของช่อง "ไม่ว่าง" ที่อยู่ติดกันอย่างน้อยหนึ่งลำดับเชื่อมต่อสองช่องดังกล่าว

ตัวอย่างที่ 1

รูปแบบของบล็อกด้านล่างนี้ ไม่มีรูปแบบใดเลยที่เป็นเมืองในอุดมคติ รูปแบบสองรูปแบบทางซ้ายนั้นไม่เป็นไปตามเงื่อนไขแรก รูปแบบที่สามไม่เป็นไปตามเงื่อนไขที่สอง และ รูปแบบที่สี่ไม่เป็นไปตามเงื่อนไขทั้งสองข้อ

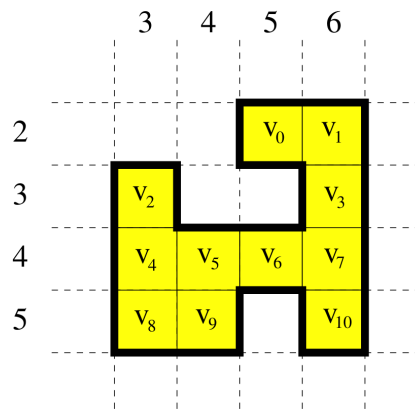


ระยะ

สำหรับการเดินทางในเมืองนั้น *hop* หมายถึงการเดินทางจากบล็อกหนึ่งไปยังบล็อกที่ติดกัน เราเดินทางผ่านช่องว่างไม่ได้ กำหนดให้ v_0, v_1, \dots, v_{N-1} เป็นตำแหน่งของบล็อก N บล็อกที่วางอยู่บนกริด สำหรับบล็อกสองบล็อกที่แตกต่างกัน ณ ตำแหน่ง v_i และ v_j ระยะ $d(v_i, v_j)$ จะเป็นจำนวนของ *hop* ที่น้อยที่สุดในการเดินทางจากบล็อกหนึ่งไปยังอีกบล็อกหนึ่ง

ตัวอย่างที่ 2

รูปแบบต่อไปนี้แสดงถึงเมืองในอุดมคติ ที่สร้างด้วยจำนวนบล็อก $N = 11$ บล็อก ณ ตำแหน่ง $v_0 = (2, 5)$, $v_1 = (2, 6)$, $v_2 = (3, 3)$, $v_3 = (3, 6)$, $v_4 = (4, 3)$, $v_5 = (4, 4)$, $v_6 = (4, 5)$, $v_7 = (4, 6)$, $v_8 = (5, 3)$, $v_9 = (5, 4)$, and $v_{10} = (5, 6)$ โดยมีตัวอย่าง $d(v_1, v_3) = 1$, $d(v_1, v_8) = 6$, $d(v_6, v_{10}) = 2$, and $d(v_9, v_{10}) = 4$



ปัญหา

งานของคุณคือ จงเขียนโปรแกรมเพื่อคำนวณ ผลรวมของระยะทั้งหมดระหว่างบล็อก v_i และ v_j โดยที่ $i < j$ จากเมืองในอุดมคติที่กำหนดมาให้ หรือกล่าวอย่างเป็นทางการว่า โปรแกรมของคุณควรคำนวณค่าของผลรวมต่อไปนี้:

$$\sum d(v_i, v_j), \text{ โดยที่ } 0 \leq i < j \leq N - 1$$

คุณจะต้องเขียนโปรแกรมย่อย DistanceSum(N, X, Y) เพื่อคำนวณสูตรข้างต้น โดยมี N และ อาร์เรย์ X และ Y เป็นค่าที่กำหนดให้ที่อธิบายเมืองในฝัน ทั้ง X และ Y มีขนาดเป็น N บล็อก i อยู่ที่คู่ลำดับ $(X[i], Y[i])$ สำหรับ $0 \leq i \leq N - 1$ และ $1 \leq X[i], Y[i] \leq 2^{31} - 2$ เนื่องจากผลลัพธ์อาจจะใหญ่เกินไปในการเก็บด้วยข้อมูลขนาด 32 บิต คุณจึงควรรายงานผลด้วยการ modulo 1,000,000,000 (หนึ่งพันล้าน)

ในตัวอย่างที่ 2, มีคู่ของบล็อกทั้งหมด $11 \times 10 / 2 = 55$ คู่ และผลรวมของระยะของแต่ละคู่ทั้งหมดคือ 174

งานย่อยที่ 1 [11 คะแนน]

รับประกันว่า $N \leq 200$.

งานย่อยที่ 2 [21 คะแนน]

รับประกันว่า $N \leq 2\,000$.

งานย่อยที่ 3 [23 คะแนน]

รับประกันว่า $N \leq 100,000$

นอกจากนี้ ตามเงื่อนไขสองข้อต่อไปนี้จะจริง: สำหรับช่องไม่ว่างสองช่อง i และ j ที่มี $X[i] = X[j]$ ทุกช่องระหว่างนั้นจะต้องเป็นช่องไม่ว่างด้วยเช่นกัน; สำหรับช่องไม่ว่างสองช่อง i และ j ที่มี $Y[i] = Y[j]$ ทุกช่องระหว่างนั้นจะต้องเป็นช่องไม่ว่างด้วยเช่นกัน

งานย่อยที่ 4 [45 คะแนน]

รับประกันว่า $N \leq 100,000$

รายละเอียดในการเขียนโปรแกรม

คุณต้องส่งหนึ่งแฟ้มในชื่อ `city.c`, `city.cpp` หรือ `city.pas` แฟ้มนี้จะต้องมีโปรแกรมย่อยตามที่อธิบาย

ไว้ข้างต้น โดยใช้รูปแบบดังต่อไปนี้

ภาษา C/C++

```
int DistanceSum(int N, int *X, int *Y);
```

ภาษา Pascal

```
function DistanceSum(N : LongInt; var X, Y : array of LongInt) : LongInt;
```

โปรแกรมย่อยเหล่านี้จะต้องทำงานตามที่ได้ระบุไว้ข้างต้น คุณสามารถเขียนโปรแกรมย่อยอื่น ๆ สำหรับใช้งานของคุณเอง โปรแกรมของคุณจะต้องไม่ยุ่งเกี่ยวกับ standard input/output หรือแฟ้มใด ๆ

grader ตัวอย่าง

grader ตัวอย่างที่ให้มาพร้อมกับสภาพแวดล้อมของโจทย์ จะรับอินพุตตามรูปแบบต่อไปนี้

- บรรทัดที่ 1: N;
- บรรทัดที่ 2, ..., N + 1: X[i], Y[i].

ข้อจำกัดของเวลาและหน่วยความจำ

- ข้อจำกัดของเวลา: 1 วินาที
- ข้อจำกัดของหน่วยความจำ: 256 MiB.