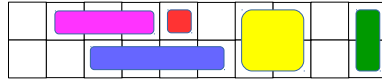


หาของ (scan)

1 second, 256MB

พื้นที่ขนาด $2 \times N$ ช่อง (2 แถว, N คอลัมน์; $1 \leq N \leq 100,000$) มีของอยู่ในนั้นจำนวนหนึ่ง (มีอย่างน้อย 1 ชิ้นแต่ไม่เกิน 40 ชิ้น) ของแต่ละชิ้นจะกินพื้นที่เป็นส่วนตารางสี่เหลี่ยมในพื้นที่ดังกล่าว โดยอาจจะกินเนื้อที่หนึ่งแถว หรือสองแถวก็ได้ ของแต่ละชิ้นจะไม่กินเนื้อที่ทับของช่องตารางทับกัน ด้านล่างแสดงตัวอย่างตารางขนาด 2×10 ที่มีของ 5 ชิ้น



คุณต้องการหาตำแหน่งและขนาดของของทุกชิ้น โดยคุณมีเครื่องตรวจสอบ โดยเมื่อให้เครื่องตรวจสอบ ส่วนของพื้นที่ เครื่องจะตอบจำนวนของที่พบในพื้นที่เหล่านั้น คุณเรียกใช้เครื่องตรวจสอบโดยเรียกไลบรารี

- `detect(r1, c1, r2, c2)` – เพื่อถามว่าในพื้นที่ที่อยู่ระหว่างแถวที่ $r1 - r2$ และคอลัมน์ $c1 - c2$ มีของอยู่ในนั้นกี่ชิ้น (จะนับของอยู่ในนั้นแค่บางส่วนด้วย) คุณเรียกฟังก์ชันนี้ได้ไม่เกิน 2,000 ครั้ง

มีอีกสองฟังก์ชันที่คุณต้องใช้คือ

- `scan_init()` – เพื่ออ่านค่า N คุณต้องเรียกฟังก์ชันนี้ก่อนการทำงานทั้งหมด
- `scan_report(k, items[][4])` – เพื่อส่งคำตอบ โดย k เป็นจำนวนของทั้งหมด และ `items` เป็นอาร์เรย์แสดงข้อมูลของของ จำนวน k ชุด แต่ละชุดเป็นอาร์เรย์ 4 ช่อง โดยที่ช่องที่ 0 ระบุตำแหน่งแถวของมุมซ้ายบนของของ ช่องที่ 1 ระบุคอลัมน์ของมุมซ้ายล่างของของ ช่องที่ 2 ระบุตำแหน่งแถวของมุมล่างขวาของของ และช่องที่ 3 ระบุคอลัมน์ของมุมล่างขวาของของ สามารถรายงานของในลำดับใดก็ได้ เมื่อเรียกฟังก์ชันนี้แล้วโปรแกรมจะจบการทำงาน

ตัวอย่างของการทำงานของไลบรารีกับตัวอย่างด้านบน เป็นดังนี้ (เป็นตัวอย่างการเรียกใช้ไลบรารีเท่านั้น ในการถามจริง ถามแค่นี้คงไม่สามารถตอบได้ถูก)

| | |
|-------------------------------------------------------------------------------------------------------|-----------------------------|
| <code>scan_init()</code> | 10 |
| <code>detect(1, 1, 2, 5)</code> | 3 |
| <code>detect(1, 5, 1, 5)</code> | 1 |
| <code>detect(1, 7, 2, 10)</code> | 2 |
| <code>detect(2, 1, 2, 7)</code> | 2 |
| <code>scan_report(5, {{1, 2, 1, 4}, {2, 3, 2, 6}, {1, 10, 2, 10}, {1, 5, 1, 5}, {1, 7, 2, 8}})</code> | โปรแกรมทดสอบจะพิมพ์ correct |

ปัญหาย่อย

- ปัญหาย่อย 1 (20%): $N \leq 100$
- ปัญหาย่อย 2 (30%): ของทุกชิ้นอยู่ในแถว 1 (นั่นคือ `detect(2, 1, 2, N) = 0`)
- ปัญหาย่อย 3 (50%): ไม่มีเงื่อนไขเพิ่มเติมจากในโจทย์ จะมีการให้คะแนนพิเศษในส่วนนี้ กล่าวคือถ้าสามารถตอบได้ถูกต้องในทุกกรณีจะได้คะแนน 40% อีก 10% จะคิดอัตราส่วนของ

จำนวนครั้งที่เรียก detect ที่มากสุดในการทำงาน เทียบกับจำนวนครั้งที่โปรแกรมของทุกคน
เรียก detect ที่มากที่น้อยที่สุด โดยคะแนน 10% จะคิดดังนี้
$$10 \times \text{จำนวนครั้งมากที่สุดที่น้อยสุด} / \text{จำนวนครั้งมากที่สุดของคุณ}$$

อย่างไรก็ตาม ระหว่างแข่ง จะไม่มีการคิดคะแนน 10% นี้ (อาจจะได้ 100 คะแนนไปก่อน) คะแนนอีก
10% จะคำนวณให้หลังแข่งขันเสร็จ

การใช้งานไลบรารีทดสอบ

ดาวน์โหลดไลบรารีทดสอบที่

<http://theory.cpe.ku.ac.th/~jittat/ioi/2016/hfiles/76gf4i-scan/>

ให้ `#include "scanlib.h"` และ compile `scanlib.cpp` ร่วมกับโปรแกรมของคุณ

ไลบรารีจะอ่านข้อมูลจาก standard input ในรูปแบบนี้

- บรรทัดแรกระบุจำนวนสองจำนวน N และ K
- อีก K บรรทัดระบุตำแหน่งของช่องแต่ละชั้น โดยระบุเป็นจำนวนเต็ม 4 จำนวน คือ r1 c1 r2 c2 แทนแถวน้อยสุด คอลัมน์น้อยสุด แถวมากสุดและคอลัมน์มากสุดของช่องในพื้นที่ที่มีของชั้นดังกล่าวอยู่

ตัวอย่าง standard input ของตัวอย่างด้านบน

```
10 5
1 10 2 10
2 3 2 6
1 2 1 4
1 5 1 5
1 7 2 8
```