



## ขุดทางลับใต้ปราสาท

ปราสาทแห่งความมืดตั้งอยู่บนพื้นที่ดินขนาดกว้าง  $R$  เมตร ยาว  $C$  เมตร เราจะสามารถพิจารณาแผนที่ของปราสาทว่าเป็นตารางขนาด  $R$  แถว  $C$  คอลัมน์ได้ เพื่อความสะดวกเราจะเรียกแถวว่าเป็นแถวที่ 0 ถึงแถวที่  $R - 1$  และคอลัมน์เป็นคอลัมน์ที่ 0 ถึงคอลัมน์ที่  $C - 1$  เวลากล่าวถึงตำแหน่ง  $(i, j)$  คือช่องที่แถวที่  $i$  และคอลัมน์ที่  $j$

ด้านใต้ปราสาทเป็นดินแข็ง เพื่อรักษาความปลอดภัย ทีมงานจะขุดทางเดินไว้ใต้ดินเพื่อเคลื่อนย้ายกำลังพล กำลังพลกลุ่มแรกมีที่ทำการหลักอยู่ที่ตำแหน่งด้านบนตรงกับช่องที่ตำแหน่ง  $(0, 0)$  และที่ทำงานสำรองอยู่ที่ช่องที่ตำแหน่ง  $(R - 1, C - 1)$  กำลังพลกลุ่มที่สองมีที่ทำการหลักอยู่ที่ตำแหน่ง  $(R - 1, 0)$  และที่ทำงานสำรองอยู่ที่ตำแหน่ง  $(0, C - 1)$

เพื่อให้การเคลื่อนกำลังพลระหว่างที่ทำการหลักและรองทำได้สะดวกโดยไม่มีใครทราบเราต้องการขุดทางเดินใต้ดิน โดยจะขุดสองเส้นทางคือเส้นทางจาก  $(0, 0)$  ไป  $(R - 1, C - 1)$  และเส้นทางจาก  $(R - 1, 0)$  ไปยัง  $(0, C - 1)$  ในการเดินทางนั้น กำลังพลจะมุดลงใต้ดินที่ตำแหน่งของที่ทำการ จากนั้นสามารถเดินไปช่องใต้ดินที่ขุดไว้ทั้งสิ้น 4 ทิศทาง ห้ามเดินออกนอกขอบเขตของปราสาท กำลังพลสองกลุ่มจะเคลื่อนย้ายคนละเวลากัน ดังนั้น เส้นทางทั้งสองอาจจะทับกันได้ ช่องที่ทับกันจะเสียค่าใช้จ่ายในการขุดแค่ครั้งเดียว

ดินใต้ปราสาทมีความแข็งไม่เท่ากัน การขุดที่ตำแหน่ง  $(i, j)$  จะต้องใช้ค่าใช้จ่าย  $A[i][j]$  ให้หาวิธีการขุดทางลับให้ใช้ค่าใช้จ่ายน้อยที่สุด

พิจารณาตัวอย่างต่อไปนี้ ด้านล่างแสดงค่าใช้จ่ายในการขุดดินใต้ปราสาทขนาด  $R = 5$  และ  $C = 6$

1	20	20	25	50	1
2	1	45	4	1	1
65	1	67	2	50	1
3	1	2	1	52	3
1	95	85	25	57	1

ในการขุดให้ประหยัดที่สุดจะใช้เส้นทางตามที่แสดงเป็นสีเข้มด้านบน ซึ่งมีค่าใช้จ่ายเท่ากับ 27 หน่วย

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
int dig_paths(int R, int C,
              vector<vector<int>>& A)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง
- $A$  จะมีเวกเตอร์ด้านในจำนวน  $R$  เวกเตอร์ แต่ละเวกเตอร์มีขนาด  $C$  ค่าใช้จ่ายในการขุดช่องใต้ดินที่ตำแหน่ง  $(i, j)$  คือ  $A[i][j]$
- ฟังก์ชันจะต้องคืนค่าค่าใช้จ่ายน้อยที่สุดที่เป็นไปได้ในการขุดเส้นทางสองเส้น จาก  $(0, 0)$  ไป  $(R - 1, C - 1)$  และจาก  $(R - 1, 0)$  ไปยัง  $(0, C - 1)$

## ขอบเขต

- $1 \leq R \leq 500, 1 \leq C \leq 500$
- $1 \leq A[i][j] \leq 2000$

## ปัญหาย่อย

1. (5 คะแนน)  $A[i][j] = 1$
2. (17 คะแนน)  $R \leq 20, C \leq 20$
3. (27 คะแนน)  $R \leq 40, C \leq 40$
4. (51 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

## ตัวอย่าง

จากตัวอย่างข้างต้น การเรียกฟังก์ชัน `dig_paths` เริ่มต้นจะเป็นดังนี้

```
dig_paths(5, 6,
          [[1, 20, 20, 25, 50, 1],
           [2, 1, 45, 4, 1, 1],
           [65, 1, 67, 2, 50, 1],
           [3, 1, 2, 1, 52, 3],
           [1, 95, 85, 25, 57, 1]])
```

ฟังก์ชันจะต้องคืนค่า 27

## เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างอ่านข้อมูลดังนี้

- บรรทัดที่ 1:  $R, C$
- บรรทัดที่  $2, \dots, R + 1$ :  $A[i][0] \ A[i][1] \ \dots \ A[i][C - 1]$

เกรดเดอร์จะพิมพ์ค่าที่คืนจากฟังก์ชัน `dig_paths`

## ข้อจำกัด

- Time limit: 1 second
- Memory limit: 512 MB