



## หอพักนักศึกษา (Dorm)

มหาวิทยาลัยแห่งหนึ่งมีวิทยาเขตทั้งหมด  $N$  วิทยาเขตได้มีการจัดการสร้างหอพักไว้สำหรับนักศึกษาจำนวน 1 หอสำหรับทุกวิทยาเขต แต่ทว่าในช่วง 3 ปีที่ผ่านมาได้เกิดโรคระบาดที่กระจายตัวได้ง่าย ทำให้ทางมหาวิทยาลัยได้ปรับเปลี่ยนระบบหอพักใหม่

สำหรับการรับสมัครนักศึกษาเพื่อเข้าพักนั้นจะมีคำร้องมาทั้งหมด  $M$  ครั้ง ซึ่งจะขอเข้าพักในวิทยาเขตที่  $L_i$  ถึง  $R_i$  จำนวนหนึ่งคนต่อหนึ่งวิทยาเขต ในแต่ละหอพักในแต่ละวิทยาเขตหากมี  $x$  คนจะมีค่าความแออัด  $x^2$  ซึ่งทางมหาวิทยาลัยได้เล็งเห็นถึงผลกระทบของโรคระบาดเป็นอย่างมาก จึงได้ทำการจ้างบริษัทก่อสร้างมาสร้างหอพักเพิ่มเพื่อลดความแออัด

เนื่องจากการสร้างหอพักแบบไม่คำนึงถึงงบประมาณเป็นเรื่องในอุดมคติ จึงทำให้ทางมหาวิทยาลัยได้จำลองการสร้างหอพักทั้งหมด  $Q$  ครั้งโดยจะสร้างเพิ่มจากตอนแรกทั้งหมด  $P_j$  หอที่วิทยาเขตใดก็ได้ ถ้ารวมผลรวมความแออัดที่น้อยที่สุดในการจำลองแต่ละครั้งคือเท่าไรหากจัดนักศึกษาไปอยู่ที่หอต่าง ๆ ได้ดีที่สุด และสร้างหอพักเหมาะสมที่สุด การจำลองทั้งหมดจะทำหลังจากมีการขอเข้าพักครบทุกคำร้องแล้ว

## รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
void init(int N, vector<vector<int>> Req)
```

- จะมีการเรียกฟังก์ชันนี้เพียงครั้งเดียว โดยตัวแปร  $N$  และ  $M$  แทนจำนวนวิทยาเขต และจำนวนคำร้องเข้าพัก
- เวกเตอร์  $Req$  จะมีขนาด  $M$  และประกอบด้วยเวกเตอร์ที่มีขนาด 2 โดยที่  $Req[i][0], Req[i][1]$  แทน  $L_i, R_i$
- ฟังก์ชันนี้ไม่ต้องคืนค่า

```
long long least_crowded(int P)
```

- จะมีการเรียกฟังก์ชันนี้  $Q$  ครั้ง
- แต่ละครั้งจะรับค่า  $P$  แทน  $P_j$  ในแต่ละครั้ง
- ฟังก์ชันนี้จะคืนค่าความแออัดที่น้อยที่สุดหากจัดนักศึกษาไปอยู่ที่หอต่าง ๆ ได้ดีที่สุด และสร้างหอพักเหมาะสมที่สุดในแต่ละครั้ง

## ขอบเขต

- $1 \leq N \leq 3\,000$
- $1 \leq M \leq 100\,000$

- $0 \leq L_i \leq R_i \leq N - 1$
- $1 \leq Q \leq 300\,000$
- $1 \leq P_j \leq 1\,000\,000\,000$

## ปัญหาย่อย

1. (7 คะแนน)  $N \leq 3, M \leq 1\,000, Q = 50$
2. (11 คะแนน)  $L_i = 0, R_i = N - 1, P_j = N \times A$  เมื่อ  $A \in \mathbb{Z}^+$
3. (15 คะแนน)  $N, M, P_j \leq 300$
4. (13 คะแนน)  $P_j \leq 300\,000$
5. (24 คะแนน)  $Q = 1$
6. (30 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

## ตัวอย่าง

```
init(5, [[0, 3], [1, 4], [1, 2], [0, 4]])
```

ถัดมาจะมีการเรียกฟังก์ชัน `least_crowd` ทั้งหมด 4 ครั้ง

```
least_crowded(1)
```

จะคืนค่า 41

```
least_crowded(2)
```

จะคืนค่า 33

```
least_crowded(10)
```

จะคืนค่า 15

```
least_crowded(100)
```

จะคืนค่า 15

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะมีการรับข้อมูลดังนี้

- บรรทัดที่ 1:  $N\ M\ Q$
- บรรทัดที่ 2 ถึง  $M + 1$ :  $Req[i][0]\ Req[i][1]$

- บรรทัดที่  $M + 2$  ถึง  $M + Q + 1$ :  $P_j$

## ข้อจำกัด

- Time limit: 2 seconds
- Memory limit: 256 MB