



ปัญหา

ในบ้านของคุณภัก บลังคอน มีแมลงอยู่ N ตัว หมายถึงตั้งแต่ 0 ถึง $N - 1$ แมลงแต่ละตัวจะมี **สายพันธุ์** ซึ่งมีค่าได้ตั้งแต่ 0 ถึง 10^9 แมลงหลายตัวอาจจะมีสายพันธุ์เดียวกันได้

สมมติว่าเราจัดกลุ่มแมลงตามสายพันธุ์ เรานิยามว่าระดับสายพันธุ์ (cardinality) ที่**พบบ่อยที่สุด** คือจำนวนของแมลงในแต่ละกลุ่มสายพันธุ์ที่มีจำนวนแมลงมากที่สุด ในทำนองเดียวกัน ระดับสายพันธุ์ที่**หายากที่สุด** คือจำนวนของแมลงในแต่ละกลุ่มสายพันธุ์ที่มีจำนวนแมลงน้อยที่สุด

ยกตัวอย่างเช่น มีแมลง 11 ตัว โดยที่สายพันธุ์ของแมลงแต่ละตัวเป็น $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$ ในกรณีนี้ ระดับสายพันธุ์ที่**พบบ่อยที่สุด** คือ 3 เนื่องจากกลุ่มที่มีจำนวนแมลงมากที่สุดคือสายพันธุ์ 9 และสายพันธุ์ 11 โดยแต่ละกลุ่มสายพันธุ์นี้มีแมลง 3 ตัว ระดับสายพันธุ์ที่**หายากที่สุด** คือ 1 เนื่องจากกลุ่มที่มีจำนวนแมลงน้อยที่สุดคือสายพันธุ์ 7, สายพันธุ์ 0 และสายพันธุ์ 100 โดยแต่ละกลุ่มสายพันธุ์นี้มีแมลง 1 ตัว

คุณภัก บลังคอนไม่รู้จักสายพันธุ์ของแมลงเลย แต่เขามีเครื่องวิเคราะห์ที่มีปุ่มหนึ่งปุ่มที่สามารถให้ข้อมูลบางอย่างเกี่ยวกับสายพันธุ์ของแมลงได้ โดยเริ่มแรก เครื่องวิเคราะห์จะว่าง ในการใช้เครื่องวิเคราะห์ จะมีสามกระบวนการให้เลือก

1. ใส่แมลงเข้าไปในเครื่องวิเคราะห์
2. นำแมลงออกจากเครื่องวิเคราะห์
3. กดปุ่มบนเครื่องวิเคราะห์

สำหรับแต่ละกระบวนการ สามารถทำได้ไม่เกิน 40 000 ครั้ง

เมื่อปุ่มถูกกด เครื่องวิเคราะห์จะรายงานระดับสายพันธุ์ที่**พบบ่อยที่สุด** จากการพิจารณาเฉพาะแมลงที่อยู่ในเครื่อง

ปัญหาของคุณคือหาค่าระดับสายพันธุ์ที่**หายากที่สุด**ในบรรดาแมลงทั้ง N ตัวที่พบในบ้านของคุณภัก บลังคอนโดยใช้เครื่องวิเคราะห์ นอกจากนั้น ในบางปัญหาย่อย คะแนนของคุณจะขึ้นอยู่กับจำนวนการเรียกใช้กระบวนการที่มากที่สุด (ดูรายละเอียดในปัญหาย่อย)

รายละเอียดการเขียนโปรแกรม

คุณต้องเขียนฟังก์ชันต่อไปนี้

```
int min_cardinality(int N)
```

- N : จำนวนแมลง
- ฟังก์ชันนี้ต้องคืนค่าระดับสายพันธุ์ที่**หายากที่สุด**จากแมลงทั้ง N ตัวที่อยู่ในบ้านของคุณภัก บลังคอน
- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียวเท่านั้น

ฟังก์ชันดังกล่าวสามารถเรียกใช้ฟังก์ชันต่อไปนี้:

```
void move_inside(int i)
```

- i : ระบุหมายเลขของแมลงที่จะใส่เข้าไปในเครื่องวิเคราะห์ โดยที่ i มีค่าได้ตั้งแต่ 0 ถึง $N - 1$
- ถ้าแมลงตัวที่ระบุอยู่ในเครื่องวิเคราะห์อยู่แล้ว ฟังก์ชันนี้จะไม่มีผลต่อเซตของแมลงที่อยู่ในเครื่อง แต่จะถูกนับเป็นการเรียกใช้ฟังก์ชันหนึ่งครั้ง
- ฟังก์ชันนี้สามารถถูกเรียกได้สูงสุด 40 000 ครั้ง

```
void move_outside(int i)
```

- i : ระบุหมายเลขของแมลงที่จะนำออกจากเครื่องวิเคราะห์ โดยที่ i มีค่าได้ตั้งแต่ 0 ถึง $N - 1$
- ถ้าแมลงตัวที่ระบุอยู่นอกเครื่องวิเคราะห์อยู่แล้ว ฟังก์ชันนี้จะไม่มีผลต่อเซตของแมลงที่อยู่ในเครื่อง แต่จะถูกนับเป็นการเรียกใช้ฟังก์ชันหนึ่งครั้ง
- ฟังก์ชันนี้สามารถถูกเรียกได้สูงสุด 40 000 ครั้ง

```
int press_button()
```

- ฟังก์ชันนี้คืนค่าเป็นระดับสายพันธุ์ที่**พบบ่อยที่สุด** โดยการวิเคราะห์จากแมลงที่อยู่ในเครื่องเท่านั้น
- ฟังก์ชันนี้สามารถถูกเรียกได้สูงสุด 40 000 ครั้ง
- เกรดเดอร์จะ**ไม่มีการเปลี่ยนแปลง** กล่าวคือสายพันธุ์ของแมลงทั้ง N ตัว จะถูกกำหนดไว้ล่วงหน้าก่อนที่ `min_cardinality` จะถูกเรียก

ตัวอย่าง

พิจารณาตัวอย่างสถานการณ์ที่มีแมลง 6 ตัว แต่ละตัวมีสายพันธุ์เป็น [5, 8, 9, 5, 9, 9] ตามลำดับ ฟังก์ชัน `min_cardinality` ถูกเรียกดังต่อไปนี้

```
min_cardinality(6)
```

ฟังก์ชันดังกล่าวอาจจะเรียกฟังก์ชัน `move_inside`, `move_outside` และ `press_button` ตามลำดับต่อไปนี้

ฟังก์ชัน	ค่าที่ได้	แมลงในเครื่องวิเคราะห์	สายพันธุ์ของแมลงในเครื่องวิเคราะห์
		{}	[]
move_inside(0)		{0}	[5]
press_button()	1	{0}	[5]
move_inside(1)		{0, 1}	[5, 8]
press_button()	1	{0, 1}	[5, 8]
move_inside(3)		{0, 1, 3}	[5, 8, 5]
press_button()	2	{0, 1, 3}	[5, 8, 5]
move_inside(2)		{0, 1, 2, 3}	[5, 8, 9, 5]
move_inside(4)		{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]
move_inside(5)		{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
press_button()	3	{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
move_inside(5)		{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
press_button()	3	{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
move_outside(5)		{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]
press_button()	2	{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]

ณ จุดนี้ จะมีข้อมูลเพียงพอที่จะสรุปได้ว่าระดับสายพันธุ์ที่หายากที่สุดคือ 1 ดังนั้น ฟังก์ชัน `min_cardinality` ควรจะคืนค่า 1 ในตัวอย่างนี้ `move_inside` ถูกเรียก 7 ครั้ง, `move_outside` ถูกเรียก 1 ครั้ง และ `press_button` ถูกเรียก 6 ครั้ง

ข้อจำกัด

- $2 \leq N \leq 2000$

ปัญหาย่อย

1. (10 points) $N \leq 200$
2. (15 points) $N \leq 1000$
3. (75 points) ไม่มีข้อจำกัดเพิ่มเติม

ถ้ามีกรณีทดสอบใดที่การเรียกฟังก์ชัน `move_inside`, `move_outside` หรือ `press_button` ไม่สอดคล้องกับข้อจำกัดที่ระบุไว้ในรายละเอียดการเขียนโปรแกรม หรือคืนค่า `min_cardinality` ที่ไม่ถูกต้อง คะแนนของคุณในปัญหาย่อยนั้นจะเป็น 0

กำหนดให้ q คือค่าที่มากที่สุดจากสามค่าดังต่อไปนี้: จำนวนการเรียก `move_inside`, จำนวนการเรียก `move_outside` และ จำนวนการเรียก `press_button`

ในปัญหาย่อยที่ 3 คุณสามารถได้คะแนนบางส่วน กำหนดให้ m คือค่าที่มากที่สุดของ $\frac{q}{N}$ จากกรณีทดสอบทุกกรณีในปัญหาย่อยนี้ คะแนนของคุณในปัญหาย่อยนี้จะถูกคำนวณตามตารางต่อไปนี้:

เงื่อนไข	คะแนน
$20 < m$	0 (รายงานเป็น "Output isn't correct" ใน CMS)
$6 < m \leq 20$	$\frac{225}{m-2}$
$3 < m \leq 6$	$81 - \frac{2}{3}m^2$
$m \leq 3$	75

เกรดเดอร์ตัวอย่าง

กำหนดให้ T เป็นอาร์เรย์ของจำนวนเต็ม N ตัว โดยที่ $T[i]$ คือสายพันธุ์ของแมลงตัวที่ i

เกรดเดอร์ตัวอย่างจะอ่านค่าอินพุตตามรูปแบบต่อไปนี้:

- บรรทัดที่ 1: N
- บรรทัดที่ 2: $T[0] T[1] \dots T[N-1]$

ถ้าเกรดเดอร์ตรวจพบว่าการละเมิดโปรโตคอล เอาท์พุทของเกรดเดอร์ตัวอย่างจะเป็น Protocol Violation: <MSG> โดยที่ <MSG> มีค่าใดค่าหนึ่งต่อไปนี้:

- invalid parameter: ในการเรียกใช้ move_inside หรือ move_outside ค่าของ i ไม่อยู่ในช่วงตั้งแต่ 0 ถึง $N-1$
- too many calls: มีการเรียกใช้ฟังก์ชันใดฟังก์ชันหนึ่งจาก move_inside, move_outside หรือ press_button มากกว่า 40 000 ครั้ง

หากไม่พบปัญหา เอาท์พุทของของเกรดเดอร์ตัวอย่างจะอยู่ในรูปแบบต่อไปนี้:

- บรรทัดที่ 1: ค่าที่ได้จากการเรียก min_cardinality
- บรรทัดที่ 2: q