

บรรจุขนมปังกรอบ (biscuits)

ป้าคง (Auntie Khong) จะจัดการแข่งขันที่มีคนเข้าร่วม x คน และต้องการแจก ขนมปังกรอบบรรจุถุง (bag of biscuits) ให้กับผู้เข้าแข่งขันทุกคน มีขนมปังกรอบแตกต่างกันทั้งสิ้น k ประเภท หมายเลขตั้งแต่ 0 ถึง $k - 1$ ขนมปังกรอบประเภทที่ i ($0 \leq i \leq k - 1$) จะมี ค่าความอร่อย เท่ากับ 2^i ในครัวของป้าคงมีขนมปังกรอบประเภท i อยู่ทั้งสิ้น $a[i]$ ชิ้น ($a[i]$ สามารถเป็นศูนย์ได้)

ในแต่ละถุงที่ป้าคงจะแจกจะมีขนมปังกรอบแต่ละประเภทจำนวนศูนย์ชิ้นหรือมากกว่า จำนวนขนมปังกรอบประเภทที่ i รวมกันทุกถุงจะต้องไม่เกิน $a[i]$ ผลรวมของค่าความอร่อยของขนมปังกรอบทุกชิ้นในถุงใด ๆ จะเรียกว่า ผลรวมความอร่อย ของถุงนั้น

ช่วยป้าคงหาว่ามีค่าของ y ที่เป็นไปได้กี่ค่า ที่ป้าคงสามารถบรรจุขนมปังกรอบจำนวน x ถุง โดยที่ทุกถุงมีผลรวมความอร่อยเท่ากับ y

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int64 count_tastiness(int64 x, int64[] a)
```

- x : จำนวนขนมปังกรอบบรรจุถุงที่ต้องการ
- a : อาร์เรย์ความยาว k สำหรับ i ใด ๆ ที่ $0 \leq i \leq k - 1$ ค่า $a[i]$ จะแทนจำนวนขนมปังกรอบประเภท i ในครัวของป้าคง
- ฟังก์ชันจะต้องคืนจำนวนค่าที่แตกต่างกันของค่า y ที่ป้าคงสามารถบรรจุขนมปังกรอบได้ x ถุง โดยที่ทุกถุงจะมีผลรวมความอร่อยเท่ากับ y
- ฟังก์ชันนี้จะถูกเรียกทั้งสิ้น q ครั้ง (ดูรายละเอียดที่ส่วนข้อจำกัดและปัญหาย่อยสำหรับค่าต่าง ๆ ที่เป็นไปได้) ให้พิจารณาแต่ละการเรียกฟังก์ชันเป็นสถานการณ์ที่แตกต่างกัน

ตัวอย่าง

ตัวอย่างที่ 1

พิจารณาการเรียกต่อไปนี้:

```
count_tastiness(3, [5, 2, 1])
```

ซึ่งหมายความว่าป้าคงต้องการจะบรรจุขนมปังกรอบจำนวน 3 ถุง และมีขนมปังกรอบ 3 ประเภทใน

ครัว

- มีขนมปังกรอบประเภท 0 จำนวน 5 ชิ้น แต่ละชิ้นมีค่าความอร่อยเท่ากับ 1
- มีขนมปังกรอบประเภท 1 จำนวน 2 ชิ้น แต่ละชิ้นมีค่าความอร่อยเท่ากับ 2
- มีขนมปังกรอบประเภท 2 จำนวน 1 ชิ้น แต่ละชิ้นมีค่าความอร่อยเท่ากับ 4

ค่า y ที่เป็นไปได้คือ $[0, 1, 2, 3, 4]$ ยกตัวอย่างเช่น ถ้าต้องการบรรจุขนมปังกรอบ 3 ถุงที่มีค่าความอร่อยรวม 3 ป้าครองสามารถบรรจุ

- ขนมปังหนึ่งถุงที่มีขนมปังกรอบประเภท 0 จำนวนสามชิ้น
- ขนมปังสองถุงที่แต่ละถุงมีขนมปังกรอบประเภท 0 หนึ่งชิ้น และขนมปังกรอบประเภท 1 อีกหนึ่งชิ้น

เนื่องจากมีค่าของ y ที่เป็นไปได้ 5 ค่า ฟังก์ชันจะต้องคืนค่า 5

<div><div></div><div></div><div></div></div> <div>$y = 0$</div>	<div><div>1</div><div>1</div><div>1</div></div> <div>$y = 1$</div>	<div><div>1,1</div><div>2</div><div>2</div></div> <div>$y = 2$</div>
<div><div>1,1,1</div><div>1,2</div><div>1,2</div></div> <div>$y = 3$</div>	<div><div>1,1,2</div><div>1,1,2</div><div>4</div></div> <div>$y = 4$</div>	

ตัวอย่างที่ 2

พิจารณาการเรียกต่อไปนี้:

```
count_tastiness(2, [2, 1, 2])
```

ซึ่งหมายความว่าป้าครองต้องการจะบรรจุขนมปังกรอบจำนวน 2 ถุง และมีขนมปังกรอบ 3 ประเภทในครัว

- มีขนมปังกรอบประเภท 0 จำนวน 2 ชิ้น แต่ละชิ้นมีค่าความอร่อยเท่ากับ 1
- มีขนมปังกรอบประเภท 1 จำนวน 1 ชิ้น แต่ละชิ้นมีค่าความอร่อยเท่ากับ 2
- มีขนมปังกรอบประเภท 2 จำนวน 2 ชิ้น แต่ละชิ้นมีค่าความอร่อยเท่ากับ 4

ค่า y ที่เป็นไปได้คือ $[0, 1, 2, 4, 5, 6]$ เนื่องจากมีค่าของ y ที่เป็นไปได้ 6 ค่า ฟังก์ชันจะต้องคืนค่า 6

ข้อจำกัด

- $1 \leq k \leq 60$

- $1 \leq q \leq 1000$
- $1 \leq x \leq 10^{18}$
- $0 \leq a[i] \leq 10^{18}$ (สำหรับทุก ๆ ค่า i ที่ $0 \leq i \leq k - 1$)
- ในการเรียก `count_tastiness` แต่ละครั้ง ผลรวมของค่าความอร่อยของขนมปังกรอบทั้งหมดทุกชิ้นในครั้วจะไม่เกิน 10^{18}

ปัญหาย่อย

1. (9 คะแนน) $q \leq 10$ และสำหรับทุก ๆ การเรียก `count_tastiness` ผลรวมของค่าความอร่อยของขนมปังกรอบทั้งหมดทุกชิ้นในครั้วจะไม่เกิน 100 000
2. (12 คะแนน) $x = 1, q \leq 10$
3. (21 คะแนน) $x \leq 10\,000, q \leq 10$
4. (35 คะแนน) ค่าตอบที่ถูกต้องในการเรียก `count_tastiness` แต่ละครั้งจะมีค่าไม่เกิน 200 000
5. (23 คะแนน) ไม่มีเงื่อนไขเพิ่มเติมอื่น ๆ

เกรตเดอร์ตัวอย่าง

เกรตเดอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้ บรรทัดแรกระบุจำนวนเต็ม q จากนั้นจะมีบรรทัดต่อมาอีก q คู่ แต่ละคู่จะอธิบายสถานการณ์หนึ่ง ๆ ซึ่งจะอยู่ในรูปแบบต่อไปนี้:

- บรรทัดที่ 1: $k \ x$
- บรรทัดที่ 2: $a[0] \ a[1] \ \dots \ a[k - 1]$

ผลลัพธ์ของเกรตเดอร์ตัวอย่างอยู่ในรูปแบบต่อไปนี้:

- บรรทัด i ($1 \leq i \leq q$): ค่าที่คืนจาก `count_tastiness` สำหรับสถานการณ์ที่ i ในข้อมูลนำเข้า