



เลขอีกแล้ว

เมืองแห่งหนึ่งได้มีการจัดแข่งขันเพื่อใช้สำหรับคัดตัวผู้ที่มีความสามารถเข้ามาทำงานเบื้องสูง ซึ่งต้องใช้ความฉลาดและไหวพริบที่สูงกว่าประชากรทั่วไปในเมืองนี้ การแข่งขันครั้งนี้ทางคณะกรรมการแข่งขันได้สุ่มจำนวนเต็มบวกมาทั้งหมด N จำนวน มีค่าอยู่ในช่วง $[1, 2000]$ ตัวที่ i จะมีค่า A_i โดยที่มีการจำลองการแข่งขันดังนี้

1. ในแต่ละรอบจะมีการสั่งให้ทำการเก็บเลข 1 ตัว หากเก็บครบ 1 ตัวแล้วจะขึ้นรอบถัดไป
2. หากเก็บ 1 ตัวครบทุกรูปแบบแล้ว จะมีการสั่งให้ทำการเก็บเลข 2 ตัว หากเก็บครบ 2 ตัวแล้วจะขึ้นรอบถัดไป
3. หากเก็บ 2 ตัวครบทุกรูปแบบแล้ว จะมีการสั่งให้ทำการเก็บเลข 3 ตัว หากเก็บครบ 3 ตัวแล้วจะขึ้นรอบถัดไป
4. หากเก็บ 3 ตัวครบทุกรูปแบบแล้ว จะมีการสั่งให้ทำการเก็บเลข 4 ตัว หากเก็บครบ 4 ตัวแล้วจะขึ้นรอบถัดไป

⋮

หลังจากเก็บเลขตามขั้นตอนดังกล่าวเสร็จแล้ว ทางผู้จัดการแข่งขันได้นิยามค่าคะแนนที่ได้ในแต่ละรอบการเก็บเลขคือ $C(S)$ ซึ่งมีค่าเท่ากับ $(\max(S) - \min(S)) \cdot D(S)$ เมื่อ

- S คือเซตที่เก็บเลขที่เก็บในแต่ละรอบ
- $D(S)$ คือจำนวนเลขที่แตกต่างกันในเซตนั้น
- คะแนนในตอนสุดท้ายคือผลรวมของค่า $C(S)$ ของทุกรอบการเก็บเลข

ยกตัวอย่างเช่น

- $D(\{7, 1, 10, 10, 7, 5, 1\}) = 4$
- $C(\{7, 1, 10, 10, 7, 5, 1\}) = 36$

เนื่องจากทางทีมงานมีข้อมูลของผลบวกเรียบร้อยแล้ว แต่ไม่แน่ใจว่าจะมีโอกาสที่ค่าที่คำนวณได้ของทางผู้เข้าแข่งขันจะตรงกับทางทีมงานหรือไม่ พวกเขาจึงฝากพวกคุณในการเป็นผู้ทดสอบเบื้องต้นให้หน่อย

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้:

```
int find_ans(std::vector<int> number)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว
- ฟังก์ชันนี้จะส่งค่า vector number ซึ่งมีขนาด N ในช่องที่ i มีค่าเท่ากับ A_i ให้กับทางผู้ใช้งาน
- ทางผู้ใช้งานจะต้องคืนค่าผลรวมของ $C(S)$ โดยให้คืนค่าเป็นจำนวนเต็ม 1 จำนวนแทน เศษจากการหารค่าตอบด้วย $10^9 + 7$

ขอบเขต

- $1 \leq N \leq 10^6$
- $1 \leq A_i \leq 2000$

ปัญหาย่อย

1. (2 คะแนน) A_i เท่ากันทุกค่า
2. (9 คะแนน) $N \leq 20$
3. (17 คะแนน) $N \leq 2000$ และ A_i เป็น permutation ของ 1 ถึง N
4. (26 คะแนน) $A_i \leq 100$
5. (19 คะแนน) $A_i \leq 500$
6. (27 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

```
find_ans([1, 1, 8, 8])
```

จะต้องคืนค่า 126 เพราะมีวิธีเลือกที่เป็นไปได้ดังนี้

1. $[1], [1], [8], [8]$ ซึ่งมีค่า $C(S)$ คือ 0, 0, 0, 0
2. $[1, 1], [1, 8], [1, 8], [1, 8], [8, 8]$ ซึ่งมีค่า $C(S)$ คือ 0, 14, 14, 14, 0
3. $[1, 1, 8], [1, 1, 8], [1, 8, 8], [1, 8, 8]$ ซึ่งมีค่า $C(S)$ คือ 14, 14, 14, 14
4. $[1, 1, 8, 8]$ ซึ่งมีค่า $C(S)$ คือ 14

```
find_ans([4, 8, 14])
```

จะต้องคืนค่า 70 เพราะมีวิธีเลือกที่เป็นไปได้ดังนี้

1. $[4], [8], [14]$ ซึ่งมีค่า $C(S)$ คือ 0, 0, 0
2. $[4, 8], [4, 14], [8, 14]$ ซึ่งมีค่า $C(S)$ คือ 8, 20, 12
3. $[4, 8, 14]$ ซึ่งมีค่า $C(S)$ คือ 30

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านข้อมูลนำเข้าดังรูปแบบต่อไปนี้:

- บรรทัดที่ 1: N
- บรรทัดที่ 2: $A_0 A_1 A_2 \dots A_{N-1}$

เกรตเตอร์ตัวอย่างจะส่งออกข้อมูลที่คืนค่ามาจากฟังก์ชัน `find_ans`

ข้อจำกัด

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