



เดอะบิ๊กไพรซ์ (The Big Prize)

เดอะบิ๊กไพรซ์ (The Big Prize) เป็นรายการทีวีที่มีชื่อเสียง คุณเป็นผู้แข่งขันผู้โชคดีเพียงหนึ่งเดียวที่ได้ผ่านเข้าไปถึงรอบสุดท้าย คุณยืนอยู่หน้ากล่องซึ่งเรียงกันเป็นแถวจำนวน n กล่อง แปะป้ายด้วยหมายเลข 0 จนถึง $n - 1$ เรียงจากซ้ายไปขวา ข้างในกล่องแต่ละใบจะมีรางวัลซึ่งไม่สามารถมองเห็นได้จนกว่ากล่องใบนั้นจะถูกเปิด รางวัลมีแตกต่างกันทั้งหมด $v \geq 2$ ชนิด ชนิดของรางวัลจะแทนด้วยหมายเลข 1 ถึง v เรียงตามลำดับมูลค่าจาก มากไปน้อย

รางวัลชนิดที่ 1 คือเพชรซึ่งมีมูลค่ามากที่สุด ในกล่องทั้งหมดจะมีเพชรอยู่หนึ่งกล่องเท่านั้น รางวัลชนิดที่ v คืออมยิ้มซึ่งมีมูลค่าน้อยที่สุด เพื่อให้เกมน่าตื่นเต้นมากขึ้น จำนวนของรางวัลที่มีมูลค่าน้อย จะมีมากกว่าจำนวนของรางวัลที่มีมูลค่ามากกว่ามันอย่างมากมาย กล่าวโดยละเอียดคือ สำหรับทุก ๆ ค่า t ที่ $2 \leq t \leq v$ เราจะทราบว่า ถ้ารางวัลชนิดที่ $t - 1$ มีจำนวน k ชิ้นแล้วรางวัลชนิดที่ t จะมีจำนวน มากกว่า k^2 ชิ้น

เป้าหมายของคุณคือเล่นเกมให้ได้เพชรเป็นรางวัล โดยเมื่อจบเกมคุณจะต้องเปิดกล่องหนึ่งใบและได้รับรางวัลที่อยู่ภายในกล่องใบนั้น ก่อนที่จะต้องเลือกกล่องเพื่อเปิด คุณสามารถถามคำถามแรมบอด (Rambod) ซึ่งเป็นพิธีกรรายการ ในการถามแต่ละคำถาม คุณจะต้องเลือกกล่อง i จากนั้นแรมบอดจะให้คำตอบเป็นอาร์เรย์ a ที่ประกอบด้วยจำนวนเต็มสองตัวซึ่งมีความหมายดังนี้:

- ในบรรดากล่องที่อยู่ทางซ้ายของกล่อง i จะมีกล่อง $a[0]$ ใบที่มีรางวัลมูลค่ามากกว่ารางวัลในกล่อง i
- ในบรรดากล่องที่อยู่ทางขวาของกล่อง i จะมีกล่อง $a[1]$ ใบที่มีรางวัลมูลค่ามากกว่ารางวัลในกล่อง i

ตัวอย่างเช่น สมมติให้ $n = 8$ หากคุณถามโดยเลือกกล่อง $i = 2$ จากนั้นแรมบอดตอบคุณด้วย $a = [1, 2]$ ความหมายของคำตอบคือ:

- มีกล่องหนึ่งใบจากกล่อง 0 และ 1 ที่มีรางวัลมูลค่ามากกว่ารางวัลในกล่อง 2
- มีกล่อง 2 ใบจากกล่อง 3, 4, ..., 7 ที่มีรางวัลมูลค่ามากกว่ารางวัลในกล่อง 2

งานของคุณคือหากกล่องที่มีเพชรโดยถามคำถามแรมบอดเป็นจำนวนน้อยครั้ง

รายละเอียดการเขียนโปรแกรม

คุณต้องเขียนโปรแกรมย่อยต่อไปนี้:

```
int find_best(int n)
```

- โปรแกรมย่อยนี้จะถูกเรียกใช้จากเกรดเดอร์เพียงหนึ่งครั้ง
- n คือจำนวนกล่อง
- โปรแกรมย่อยต้องคืนค่าเป็นหมายเลขของกล่องที่มีเพชรอยู่ นั่นคือ จำนวนเต็ม d ($0 \leq d \leq n - 1$)

ซึ่งมีค่าที่เป็นไปได้ค่าเดียว และเป็นหมายเลขของกล่องที่มีรางวัลชนิดที่ 1

โปรแกรมย่อยด้านบนสามารถเรียกใช้โปรแกรมย่อยต่อไปนี้:

```
int[] ask(int i)
```

- i แทนป้ายของกล่องที่คุณเลือกถาม ค่าของ i จะต้องอยู่ระหว่าง 0 ถึง $n - 1$ (รวม 0 และ $n - 1$ ด้วย)
- โปรแกรมย่อยนี้จะคืนค่าเป็นอาร์เรย์ a ที่มีข้อมูลสองตัว ในกรณีนี้ $a[0]$ จะเป็นจำนวนของรางวัลที่อยู่ในกล่องที่อยู่ด้านซ้ายของกล่อง i ที่มีมูลค่ามากกว่ารางวัลในกล่อง i และ $a[1]$ แทนจำนวนของรางวัลที่อยู่ในกล่องที่อยู่ด้านขวาของกล่อง i ที่มีมูลค่ามากกว่ารางวัลในกล่อง i

ตัวอย่าง

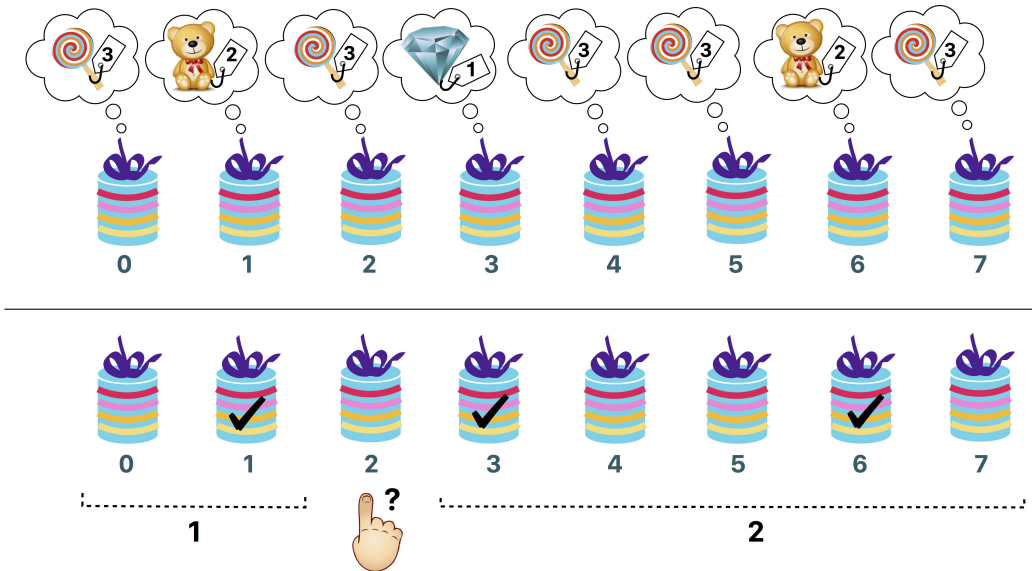
เกรตเตอร์เรียกโปรแกรมย่อยต่อไปนี้:

```
find_best(8)
```

มีกล่อง $n = 8$ ใบ สมมติว่ารางวัลในกล่องแต่ละใบคือ $[3, 2, 3, 1, 3, 3, 2, 3]$ การเรียกโปรแกรมย่อย `ask` ที่เป็นไปได้ทั้งหมด และค่าส่งกลับที่สอดคล้องกัน แสดงในรายการด้านล่าง

- `ask(0)` คืนค่าเป็น $[0, 3]$
- `ask(1)` คืนค่าเป็น $[0, 1]$
- `ask(2)` คืนค่าเป็น $[1, 2]$
- `ask(3)` คืนค่าเป็น $[0, 0]$
- `ask(4)` คืนค่าเป็น $[2, 1]$
- `ask(5)` คืนค่าเป็น $[2, 1]$
- `ask(6)` คืนค่าเป็น $[1, 0]$
- `ask(7)` คืนค่าเป็น $[3, 0]$

ในตัวอย่างนี้ เพชรอยู่ในกล่อง 3 ดังนั้น โปรแกรมย่อย `find_best` จะต้องคืนค่าเป็น 3



รูปด้านบนแสดงตัวอย่างดังกล่าว โดยส่วนบนแสดงชนิดของรางวัลในแต่ละกล่อง และส่วนล่างแสดงการเรียก $\text{ask}(2)$ กล่องที่มีเครื่องหมายถูกคือกล่องที่มีรางวัลมูลค่ามากกว่ารางวัลในกล่อง 2

ข้อจำกัด

- $3 \leq n \leq 200\,000$.
- ชนิดของรางวัลในแต่ละกล่องอยู่ในช่วง 1 ถึง v (รวม 1 และ v ด้วย)
- มีหนึ่งกล่องที่มีรางวัลชนิดที่ 1
- สำหรับทุก ๆ $2 \leq t \leq v$, ถ้ารางวัลชนิดที่ $t - 1$ มีอยู่ k ชั้น รางวัลชนิดที่ t จะมี มากกว่า k^2 ชั้น

ปัญหาย่อยและการให้คะแนน

ในบางกรณีทดสอบ พฤติกรรมของเกรดเดอร์จะปรับเปลี่ยนได้ นั่นหมายความว่าในกรณีทดสอบเหล่านี้ เกรดเดอร์ไม่ได้มีลำดับของรางวัลที่ระบุไว้คงที่ก่อน แต่คำตอบที่เกรดเดอร์ให้กับโปรแกรมอาจจะขึ้นกับคำถามที่โปรแกรมของคุณถาม เกรดเดอร์จะตอบคำถามโดยที่รับประกันว่าหลังจากตอบแต่ละคำถามแล้ว จะยังมีลำดับของรางวัลที่สอดคล้องกับคำตอบทั้งหมดที่ผ่านมาน้อยหนึ่งลำดับ

1. (20 คะแนน) จะมีเพชร 1 ชั้น และมียัมจำนวน $n - 1$ อัน (นั่นคือ $v = 2$) คุณสามารถเรียกโปรแกรมย่อย ask ได้ไม่เกิน 10 000 ครั้ง
2. (80 คะแนน) ไม่มีเงื่อนไขเพิ่มเติมอื่น ๆ

ในปัญหาย่อยที่ 2 คุณสามารถได้รับคะแนนบางส่วนได้ ให้ q แทนจำนวนการเรียกโปรแกรมย่อย ask ที่มากที่สุดที่โปรแกรมของคุณใช้ในการตอบข้อมูลทดสอบในปัญหาย่อยนี้ คะแนนที่คุณได้จะคิดตามตารางต่อไปนี้:

จำนวนคำถาม	คะแนน
$10\,000 < q$	0 (แสดงใน CMS ว่า 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างไม่ได้ทำงานแบบปรับเปลี่ยนได้ (adaptive) แต่อ่านข้อมูลอาร์เรย์ p ที่ระบุชนิดของรางวัล สำหรับทุก ๆ $0 \leq b \leq n - 1$, ชนิดของรางวัลในกล่อง b จะระบุด้วย $p[b]$ เกรดเดอร์ตัวอย่างอ่านข้อมูลนำเข้าในรูปแบบดังนี้

- บรรทัดที่ 1: n
- บรรทัดที่ 2: $p[0] \ p[1] \ \dots \ p[n - 1]$

เกรดเดอร์ตัวอย่างพิมพ์คำตอบบรรทัดเดียวเป็นค่าที่คืนจากโปรแกรมย่อย `find_best` และจำนวนการเรียกใช้โปรแกรมย่อย `ask`