



สเปกโตรโฟโตมิเตอร์

การศึกษาวิทยาศาสตร์ได้มีการพัฒนาขึ้นเรื่อย ๆ เราทุกคนต่างก็รู้ว่าแสงอาทิตย์เป็นยังไง ต่อมาก็มีนักประดิษฐ์สร้างหลอดไฟ มนุษย์สร้างอุปกรณ์ที่แปลงแสงออกมาได้ ตลอดระยะเวลาที่ผ่านมา มนุษย์ล้วนต่อสู้กับธรรมชาติ เราต้องการค้นคว้าหาความจริง อธิบายสิ่งที่เกิดขึ้น และทุกอย่างนั้นเริ่มจากการตั้งคำถาม คำถามง่าย ๆ เช่น ทำไมดวงอาทิตย์ถึงสว่าง หรือทำไมวัตถุบางอย่างจึงเปล่งแสงออกมาได้ เป็นต้น เมื่อถามไปเรื่อย ๆ วันหนึ่งมนุษย์ก็ติดสอยห้อยตามคำถามที่ว่า "แสงคืออะไรกันแน่?" บางคนบอกว่าแสงเป็นคลื่น บางคนบอกว่าแสงเป็นอนุภาค จนในที่สุดก็เป็นที่ยอมรับกันว่าเราสามารถมองแสงได้ว่าเป็นทั้งคลื่นและอนุภาค ความทะเยอทะยาน ความไม่หยุดที่จะค้นคว้านี้เองนั้นแหละ เป็นสิ่งที่ทำให้มนุษย์ก้าวหน้าขึ้นไป สร้างเทคโนโลยีต่อเติมมากขึ้นเรื่อย ๆ ในทุกทุกวัน

สเปกโตรโฟโตมิเตอร์ (spectrophotometer) นั้นก็เป็นอุปกรณ์ที่มีหน้าที่ไม่ซับซ้อนเลย คือใช้ในการวัดองค์ประกอบของแสง (ในที่นี้คือวัดองค์ประกอบของแสงที่มองเห็นได้ด้วยตาเปล่า นั่นคือแสงที่มีความยาวคลื่นประมาณ 400 ถึง 700 นาโนเมตร) เราสามารถประดิษฐ์อุปกรณ์สเปกโตรโฟโตมิเตอร์คร่าว ๆ เองได้จากเลนส์และปริซึม เพราะโดยปกติปริซึมจะทำการหักเหแสงที่มีความยาวคลื่นแตกต่างกันออกเป็นสเปกตรัม

สำหรับการทดลองนี้ เรามีแหล่งกำเนิดแสง $3N$ ขึ้น (สามารถคิดได้ว่าเป็นแท่งฉายแสงเลเซอร์ $3N$ อัน) ที่มีขนาดเล็กมาก ระดับไมโครเมตร เราจึงไม่สามารถไปหยิบจับด้วยมือเปล่าได้ นอกจากนี้แสงที่ส่องออกมาก็จางและเล็กน้อยด้วย ไม่สามารถมองเห็นได้ด้วยตาเปล่า แต่เราทราบมาก่อนว่าแสงจากแหล่งกำเนิดแสงทั้ง $3N$ ขึ้น นั้นเป็นแสงสีเดียว (monochromatic light) โดยประกอบด้วยแหล่งกำเนิดแสงความยาวคลื่น 405nm, 520nm และ 660nm อย่างละ N ขึ้น (แต่อาจไม่ได้ยู่ติดกัน) อย่างไรก็ดี แหล่งกำเนิดแสงทั้ง $3N$ ขึ้นนี้ เรียงต่อกันเป็นแถวลำดับ เรียงกระบุดำแหน่งได้ตั้งแต่ในตำแหน่งที่ 0 ถึงตำแหน่งที่ $3N - 1$

เพื่อที่จะศึกษา แยกแยะ แหล่งกำเนิดแสงทั้ง $3N$ ขึ้น เราจึงทำการทดลองโดยใช้อุปกรณ์สเปกโตรโฟโตมิเตอร์ที่ได้เตรียมไว้ ทำการวัดองค์ประกอบแสงจากแหล่งกำเนิดแสงทั้ง $3N$ ขึ้น อย่างไรก็ตาม สเปกโตรโฟโตมิเตอร์นี้ เกิดข้อจำกัด ไม่สามารถบอกค่าสีของแหล่งกำเนิดแสงแต่ละอันได้ แต่ถึงอย่างนั้น เรามีอุปกรณ์สำรอง* ที่หากทำการสลับแหล่งกำเนิดแสงสองชิ้น แล้วจะศึกษาค่าความต่างของความยาวคลื่นนั้นออกมาได้ โดยจะระบุว่าแสงที่เกิดจากแหล่งกำเนิดแสงทั้งสองนั้น มีความยาวคลื่นเท่ากันหรือไม่ เป้าหมายของข้อนี้คือ เราต้องการจัดวางแหล่งกำเนิดแสงทั้ง $3N$ ขึ้นให้แสงที่มีความยาวคลื่นเท่ากันอยู่ติดกัน

*หมายเหตุ อุปกรณ์สำรองนี้ไม่มีอยู่ในชีวิตจริง เป็นเพียงเรื่องสมมติขึ้นมาสำหรับข้อนี้เท่านั้น (แต่สเปกโตรโฟโตมิเตอร์มีอยู่จริง)

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันดังต่อไปนี้

```
void sort_lasers(int N)
```

- ฟังก์ชันนี้จะถูกเรียกเพียงครั้งเดียว โดยจะระบุค่า N แทนจำนวนแหล่งกำเนิดแสงแต่ละชนิด

- ภายในฟังก์ชันนี้ สามารถเรียกใช้งานฟังก์ชันต่อไปนี้ได้

```
bool call_swap(int a, int b)
```

- อุปกรณ์สำรอง จะทำการสลับแหล่งกำเนิดแสงในตำแหน่งที่ a กับตำแหน่งที่ b (เมื่อ $0 \leq a, b < 3N$ และ $a \neq b$)
- หากความยาวคลื่นของแหล่งกำเนิดแสงในตำแหน่ง a และ b มีค่าเท่ากัน จะคืนค่า true แต่หากความยาวคลื่นต่างกันจะคืนค่า false

ขอบเขต

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

การให้คะแนน

สำหรับข้อนี้ จะมีปัญหาย่อย 3 ปัญหาย่อย ดังต่อไปนี้

1. (30 คะแนน) $N \leq 3$
2. (40 คะแนน) $N \leq 30$
3. (30 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

โดยแต่ละปัญหาย่อยจะคิดคะแนนแบบเดียวกัน ดังนี้

ให้ S แทนจำนวนครั้งที่ผู้เข้าแข่งขันเรียก call_swap แล้วให้ $T = \frac{S}{N}$ จะได้คะแนนตามตารางดังต่อไปนี้

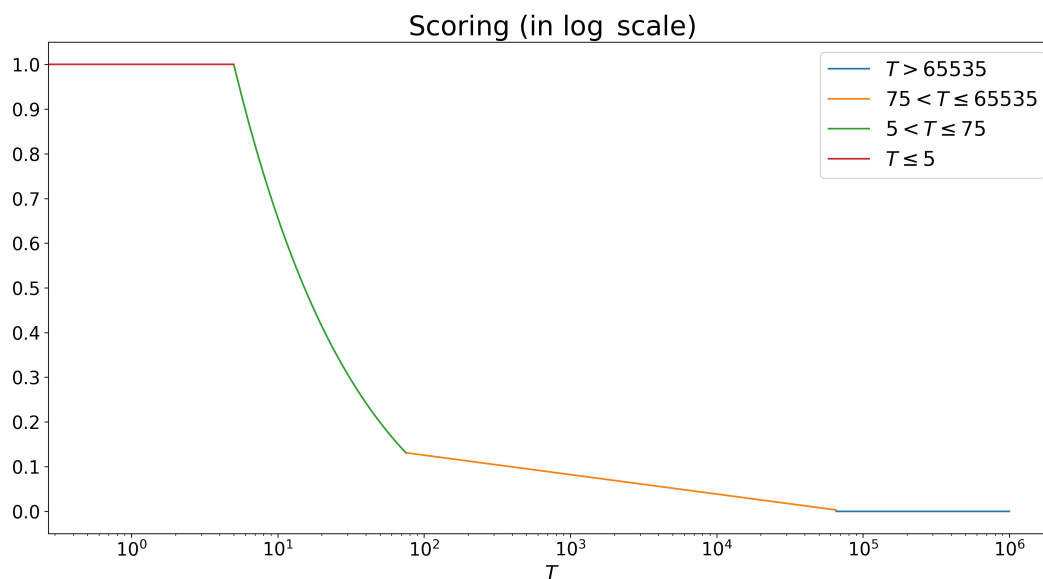
เงื่อนไข	อัตราส่วนคะแนน
$T > 65535$	0
$75 < T \leq 65535$	$0.0131 \cdot (10 - \log_2(T) + \log_2 75)$
$5 < T \leq 75$	$1.171472\sqrt{\frac{5}{T}} - 0.171472$
$T \leq 5$	1

แล้วคะแนนในชุดทดสอบนั้นจะมีค่าเท่ากับ อัตราส่วนคะแนน (จากตาราง) คูณด้วยคะแนนเต็มของปัญหาย่อยนั้น และคะแนนในแต่ละปัญหาย่อยจะเป็น **ค่าน้อยสุด** ของคะแนนในแต่ละชุดทดสอบในปัญหาย่อยนั้น

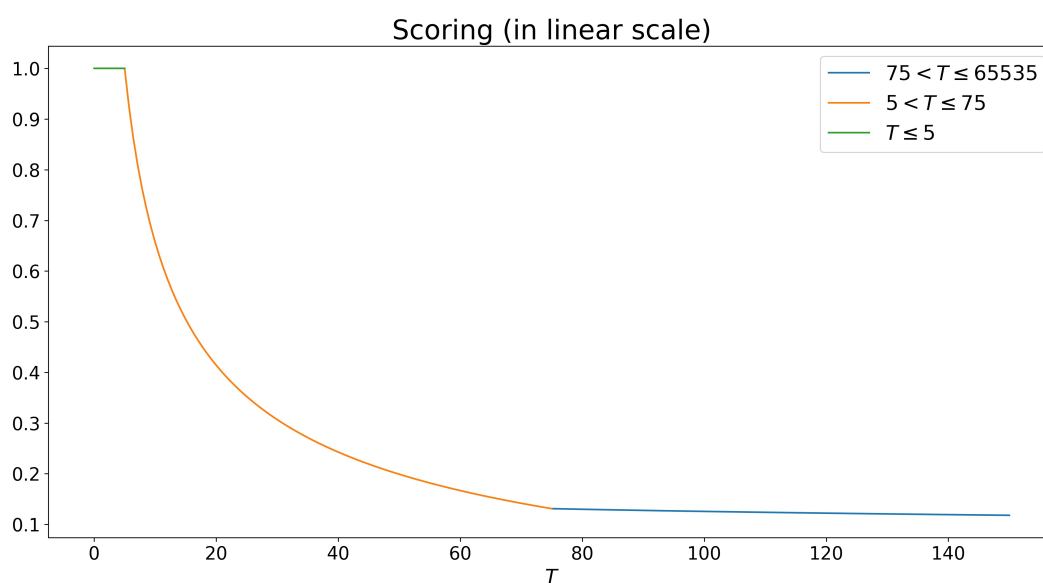
หมายเหตุ

- หากอัตราส่วนคะแนนที่ได้มีค่าไม่เท่ากับ 1 (รวมถึงกรณีที่คำตอบถูกแต่ใช้ $T > 65535$) ตัวตรวจจะแสดงผล Output is partially correct
- หากอัตราส่วนคะแนนที่ได้มีค่าเท่ากับ 1 ตัวตรวจจะแสดงผล Output is correct
- ส่วนกรณีที่ตัวตรวจแสดงผล Output isn't correct จะเกิดขึ้นเมื่อเกิดข้อผิดพลาดในการเรียกฟังก์ชัน call_swap หรือเมื่อ sort_lasers จบการทำงานแล้วแหล่งกำเนิดแสงที่มีความยาวคลื่นเท่ากันกลับไม่อยู่ติดกัน

พิจารณาแผนภาพประกอบการให้คะแนน ดังต่อไปนี้



(ภาพถัดมาจะตัดมาเฉพาะส่วนที่ T มีค่าไม่มากเกินไป)



ตัวอย่าง

สมมติแหล่งกำเนิดแสงมีอยู่ 6 ชิ้น โดยมีความยาวคลื่น 405, 520, 660, 520, 405, 660 ตามลำดับ เกรดเดอร์จะทำการเรียกฟังก์ชัน `sort_lasers(2)` ต่อมา เรายังไม่รู้ความยาวคลื่น แต่ด้วยวิธีการบางอย่าง เราสามารถทดลองสลับได้ดังนี้

```
call_swap(1, 2)
```

จะทำการสลับแหล่งกำเนิดแสงในตำแหน่งที่ 1 กับ 2 และคืนค่า false (เพราะพบว่าความยาวคลื่นต่างกัน) จะได้ความยาวคลื่นของแหล่งกำเนิดแสงแต่ละอันเป็น 405, 660, 520, 520, 405, 660 ตามลำดับ ต่อมาทำการเรียก

```
call_swap(1, 5)
```

จะทำการสลับแหล่งกำเนิดแสงในตำแหน่งที่ 1 กับ 5 และคืนค่า true (เพราะพบว่าความยาวคลื่นเหมือนกัน) จึงไม่เกิดการเปลี่ยนแปลง

```
call_swap(0, 5)
```

จะทำการสลับแหล่งกำเนิดแสงในตำแหน่งที่ 0 กับ 5 และคืนค่า false (เพราะพบว่าความยาวคลื่นต่างกัน) จะได้ความยาวคลื่นของแหล่งกำเนิดแสงแต่ละอันเป็น 660, 660, 520, 520, 405, 405 ตามลำดับ ซึ่งถือว่าเรียงแล้วฟังก์ชัน `sort_lasers` จึงสามารถหยุดการทำงานได้ (โดยไม่ต้องทำอะไรต่อในฟังก์ชัน) แล้วตัวตรวจจะทำการคำนวณคะแนน โดยคำนวณ $T = \frac{S}{N} = \frac{3}{2}$ แล้วเนื่องจากพบว่า $T \leq 5$ จึงได้อัตราส่วนคะแนน 1 (คะแนนเต็มสำหรับข้อมูลทดสอบนั้น)

หมายเหตุ

- หากทำการ `call_swap(a, b)` แล้วทำการ `call_swap(a, b)` ซ้ำ จะทำให้แหล่งกำเนิดแสง a และ b สลับกลับมายู่ตำแหน่งเดิม
- นอกจากนี้ การเรียก `call_swap(a, b)` กับ `call_swap(b, a)` นั้นจะทำงานแบบเดียวกัน

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านข้อมูลดังต่อไปนี้:

- บรรทัดที่ 1: N
- บรรทัดที่ 2: $\lambda_0 \ \lambda_1 \ \dots \ \lambda_{3N-1}$

โดย λ_i แทนความยาวคลื่นของแหล่งกำเนิดแสงที่ตำแหน่ง i ในหน่วยนาโนเมตร สำหรับ $0 \leq i < 3N$

หากการเรียกใช้งาน `call_swap` มีข้อผิดพลาด เกรตเตอร์ตัวอย่างจะระบุ `invalid swap parameters` หากฟังก์ชัน `sort_lasers` จบการทำงานโดยไม่มีปัญหา เกรตเตอร์ตัวอย่างจะทำการตรวจคำตอบ หากแหล่งกำเนิดแสงที่มีความยาวคลื่นเท่ากันไม่ได้ยู่ติดกัน เกรตเตอร์ตัวอย่างจะส่งออก `wrong answer` ตามด้วยค่าความยาวคลื่นของแหล่งกำเนิดแสงแต่ละอันจากตำแหน่งที่ 0 ถึงตำแหน่งที่ $3N - 1$

แต่หากคำตอบถูกต้อง (นั่นคือแหล่งกำเนิดแสงที่มีความยาวคลื่นเท่ากันอยู่ติดกัน) เกรตเตอร์ตัวอย่างจะส่งออก `correct answer` ตามด้วยจำนวนครั้งที่มีการเรียก `call_swap` ในบรรทัดถัดไป

ข้อจำกัด

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