

DNA กลายพันธุ์

เกรซเป็นนักชีววิทยาที่ทำงานอยู่ในบริษัททางชีวสารสนเทศศาสตร์ในประเทศสิงคโปร์ หน้าตัวอย่างหนึ่งของเธอคือการวิเคราะห์สายลำดับ DNA ของสิ่งมีชีวิตหลากหลายประเภท สายลำดับ DNA ถูกจำกัดความเป็นสายอักขระที่ประกอบไปด้วยอักขระ "A", "T", และ "C" สังเกตว่าสายลำดับ DNA สำหรับโจทย์นี้ **ไม่มีอักขระ "G"**

เราจำกัดความการกลายพันธุ์ว่า เป็นกระบวนการที่กระทำบนสายลำดับ DNA โดยที่สององค์ประกอบในลำดับนั้นถูกสลับที่ ตัวอย่างเช่น การกลายพันธุ์หนึ่งครั้งสามารถเปลี่ยน "ACTA" ให้เป็น "AATC" โดยสลับอักขระ "A" และ "C" ที่เน้นไว้

ระยะห่างการกลายพันธุ์ระหว่างสองลำดับ คือจำนวนการกลายพันธุ์ที่น้อยที่สุดที่จำเป็นในการกลายรูปลำดับหนึ่งให้เป็นอีกลำดับหนึ่ง หรือ -1 ถ้าไม่สามารถกลายรูปลำดับหนึ่งให้เป็นอีกลำดับหนึ่งได้ด้วยการกลายพันธุ์

เกรซกำลังวิเคราะห์สายลำดับ DNA สองสาย a และ b โดยทั้งคู่ประกอบด้วย n องค์ประกอบที่ถูกกำกับด้วยเลขตั้งแต่ 0 ถึง $n - 1$ งานของคุณคือการช่วยเกรซตอบคำถาม q คำถามในรูปแบบ ระยะห่างการกลายพันธุ์ระหว่างสายอักขระย่อย $a[x..y]$ และสายอักขระย่อย $b[x..y]$ ว่ามีค่าเป็นเท่าไร ในที่นี้ สายอักขระย่อย $s[x..y]$ ของสายลำดับ DNA s ถูกจำกัดความเป็นลำดับของอักขระที่ติดกันของ s โดยที่มีตำแหน่งตั้งแต่ x ถึง y รวมหัวท้าย หรือในอีกความหมายคือ $s[x..y]$ เป็นลำดับ $s[x]s[x + 1] \dots s[y]$

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันต่อไปนี้

```
void init(string a, string b)
```

- a, b : สายอักขระความยาว n อธิบายถึงสายลำดับ DNA สองสายที่จะนำไปวิเคราะห์
- ฟังก์ชันนี้ถูกเรียกหนึ่งครั้งเท่านั้น ก่อนที่จะมีการเรียกฟังก์ชัน `get_distance`

```
int get_distance(int x, int y)
```

- x, y : ตำแหน่งเริ่มต้นและสิ้นสุดของสายอักขระย่อยที่จะนำไปวิเคราะห์
- ฟังก์ชันนี้คืนค่าระยะห่างของการกลายพันธุ์ระหว่างสองสายอักขระย่อย $a[x..y]$ และ $b[x..y]$
- ฟังก์ชันนี้ถูกเรียกเป็นจำนวน q ครั้งพอดี

ตัวอย่าง

พิจารณาการเรียกต่อไปนี้:

```
init("ATACAT", "ACTATA")
```

.. ..

สมมุติว่าเกรดเดอร์เรียก `get_distance(1, 3)` การเรียกครั้งนี้ควรคืนค่าระยะทางการกลายพันธุ์ระหว่าง $a[1..3]$ และ $b[1..3]$ นั่นคือลำดับ "TAC" และ "CTA" "TAC" สามารถกลายพันธุ์เป็น "CTA" ด้วยการกลายพันธุ์ 2 ครั้ง: TAC \rightarrow CAT ตามด้วย CAT \rightarrow CTA และการกลายรูปไม่สามารถเป็นไปได้อย่างน้อย 2 ครั้ง

ดังนั้น การเรียกครั้งนี้ควรคืนค่า 2

สมมุติว่าเกรดเดอร์เรียก `get_distance(4, 5)` การเรียกครั้งนี้ควรคืนค่าระยะทางการกลายพันธุ์ระหว่าง "AT" และ "TA" "AT" สามารถกลายพันธุ์เป็น "TA" ด้วยการกลายพันธุ์หนึ่งครั้ง และเป็นที่ยืนยันว่าจำเป็นต้องมีการกลายพันธุ์อย่างน้อยหนึ่งครั้ง

ดังนั้น การเรียกครั้งนี้คืนค่า 1

สุดท้าย สมมุติว่าเกรดเดอร์เรียก `get_distance(3, 5)` เนื่องจากไม่มีทางใดสำหรับลำดับ "CAT" ที่จะกลายรูปเป็น "ATA" ด้วยลำดับการกลายพันธุ์ใดๆ การเรียกครั้งนี้คืนค่า -1

ข้อจำกัด

- $1 \leq n, q \leq 100\,000$
- $0 \leq x \leq y \leq n - 1$
- อักขระแต่ละตัวใน a และ b เป็นหนึ่งใน "A", "T", และ "C"

ปัญหาย่อย

1. (21 คะแนน) $y - x \leq 2$
2. (22 คะแนน) $q \leq 500$, $y - x \leq 1000$, อักขระแต่ละตัวใน a และ b เป็น "A" หรือ "T"
3. (13 คะแนน) อักขระแต่ละตัวใน a และ b เป็น "A" หรือ "T"
4. (28 คะแนน) $q \leq 500$, $y - x \leq 1000$
5. (16 คะแนน) ไม่มีเงื่อนไขใด ๆ เพิ่มเติม

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้:

- บรรทัดที่ 1: $n\ q$
- บรรทัดที่ 2: a
- บรรทัดที่ 3: b
- บรรทัดที่ $4 + i$ ($0 \leq i \leq q - 1$): $x\ y$ สำหรับการเรียกฟังก์ชัน `get_distance` ครั้งที่ i

เกรดเดอร์ตัวอย่างเขียนข้อมูลส่งออกในรูปแบบต่อไปนี้:

- บรรทัดที่ $1 + i$ ($0 \leq i \leq q - 1$): ค่าที่คืนจากการเรียกฟังก์ชัน `get_distance` ครั้งที่ i