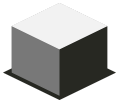




CCC beta #8

Editorial



## 1. Spam Keys by SaBuZa

---

โจทย์ข้อนี้ทดสอบทักษะการเปรียบเทียบสตริงทั่วไป

โดยเราสามารถไล่เปรียบเทียบตั้งแต่ตัวอักษรตัวแรกจนถึงตัวสุดท้ายของสตริงที่รับเข้ามาแล้วเทียบว่าตัวอักษรตัวนั้นอยู่ในตัวอักษรที่ต้องการหรือไม่

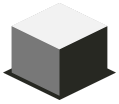
กรณีที่ต้องระวัง คือ กรณีตรวจสอบสตริง “SHIFT” และ “SPACEBAR”

แต่เราสามารถแก้ปัญหาคงความยุ่งยากในการทดสอบได้โดยทุกๆครั้งที่เทียบตัวอักษรแต่ละตัว

หากตัวอักษรตัวนั้นเป็นตัวอักษร ‘S’ ให้ตรวจสอบว่า นับตั้งแต่ตัวอักษรนั้นไป 5 หรือ 8 ตัวตามลำดับ

เป็นคำว่า “SHIFT” หรือ “SPACEBAR” หรือไม่ ได้เลย โดยจำนวนครั้งในการทำงานทั้งหมดอย่างคร่าวๆ

สำหรับสตริงที่ยาว N จะทำงานไม่เกิน  $8 \times N$  ครั้งอย่างแน่นอน



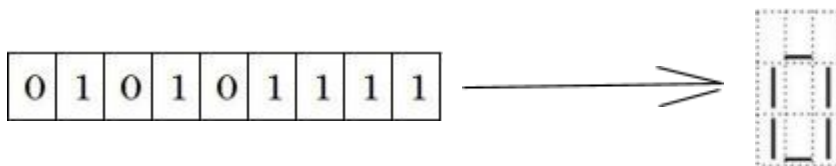
## 2. Underwatch by Nottyking

โจทย์ข้อนี้มีวิธีการทำได้หลายวิธี และวิธีที่ผู้แต่งโจทย์จะเขียนต่อไปนี้เป็นวิธีที่ง่าย เชื่อว่าหลาย ๆ คนอ่านแล้วน่าจะพอเข้าใจ และทำได้

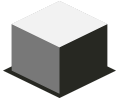
ก่อนอื่นเราต้องสร้างอาร์เรย์สองมิติขนาด  $10 \times 9$  ขึ้นมาก่อนเพื่อ Generate ตัวเลขทุกตัวโดยให้ 0 แทน “ ” (Spacebar) และเลข 1 แทน “|” และ “\_” ดังตารางข้างล่างนี้

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

ตารางข้างบนแสดงรูปแบบการแสดงตัวเลข 0 ถึง 9 โดยในแถวที่  $i$  ( $0 \leq i \leq 10$ ) จะแสดงรูปแบบของเลข  $i$  ดังตัวอย่างเลข 0 ข้างล่างนี้



หลังจากที่เรา Generate ตัวเลขทุกตัวลงอาร์เรย์เสร็จแล้ว ให้เราทำการวนลูปสมมติให้ตัวแปรวนลูปเป็น  $i$  ตั้งแต่เลข 0 ถึง 1440 หรือ  $60 \times 24$  แทนเวลาที่เราจะตรวจสอบ โดย  $i/60$  จะเป็นชั่วโมง และ  $i\%60$  จะเป็นนาที จากนั้นเราจะสามารถตรวจสอบว่าเลขจาก input นั้นสามารถเป็นเวลาใด ๆ ได้บ้างโดยใช้อาร์เรย์ที่เรา Generate ขึ้นมา และนับจำนวนวิธีที่สามารถอ่านได้จาก input ได้



### 3. Number Operation by PalmPTSJ

---

สำหรับข้อนี้มีวิธีการทำได้หลายวิธี โดยวิธีที่ผู้แต่งโจทย์ใช้คือการสังเกตว่า 0 คูณอะไรก็ได้ 0 และ  $X \leq N$  ดังนั้น ถ้าเราสามารถทำให้เกิดเลข 0 ขึ้นได้ เราสามารถนำมันไปคูณกับทุกตัวที่เราไม่ต้องการ แล้วสุดท้ายเราก็แค่บวกตัว X เข้าไป เช่น  $N=10$   $X=5$  เราสามารถทำเป็น  $3-2-1*4*6*7*8*9*10+5$  ได้ โดย  $3-2-1$  จะมีค่าเท่ากับ 0 แล้วจากนั้นเราก็คูณทุกตัวที่เราไม่ต้องการ แล้วก็จบด้วย  $+5$  ก็จะได้คำตอบ แต่เนื่องจากถ้า  $N$  มีค่าน้อย ๆ เราอาจจะทำเป็นเลข 0 ไม่ได้ จึงต้องมีการดักแยกเป็นกรณีไป

กรณีที่  $X = 1$

$N = 2$  ให้ตอบว่า  $2-1$

$N = 3$  ให้ตอบว่า  $3-2*1$

$N = 4$  ให้ตอบว่า  $2+3-4*1$

$N \geq 5$  ให้ตอบว่า  $5-4+2-3*...+1$  โดยที่ ... คือเลขอื่น ๆ ที่ยังไม่ได้ใช้

กรณีที่  $X = 2$

$N = 2$  ให้ตอบว่า  $2*1$

$N = 3$  ให้ตอบว่า  $3-2+1$

$N \geq 4$  ให้ตอบว่า  $4-3-1*...+2$  โดยที่ ... คือเลขอื่น ๆ ที่ยังไม่ได้ใช้

กรณีที่  $X = 3$

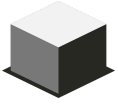
$N = 3$  ให้ตอบว่า  $2-1*3$

$N = 4$  ให้ตอบว่า  $4-2-1*3$

$N \geq 5$  ให้ตอบว่า  $5+1-4-2*...+3$  โดยที่ ... คือเลขอื่น ๆ ที่ยังไม่ได้ใช้

กรณีที่  $X \geq 4$

ให้ตอบว่า  $3-2-1*...+X$  โดยที่ ... คือเลขอื่น ๆ ที่ยังไม่ได้ใช้



#### 4. Water Level by PanTA (Editorial by PalmPTSJ)

---

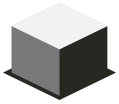
สำหรับช่องใด ๆ ความสูงของน้ำจะมีค่าเท่ากับ  $\min(\max(\text{ความสูงกำแพงทางซ้ายทั้งหมด}), \max(\text{ความสูงกำแพงทางขวาทั้งหมด}))$

เราสามารถทำ preprocess เพื่อหา  $\max(\text{ความสูงกำแพงทางซ้ายทั้งหมด})$  ของช่องที่  $i$  ใด ๆ ได้โดยใช้ Dynamic Programming ได้ โดยใช้สมการ

$$DP[i] = \max(DP[i-1], arr[i])$$

และสำหรับ  $\max(\text{ความสูงกำแพงทางขวาทั้งหมด})$  เราก็ทำได้คล้าย ๆ กันแค่กลับด้าน

หลังจากทำ preprocess ทั้งสองด้านเสร็จแล้วเราก็สามารถไล่หาความสูงน้ำในแต่ละช่องได้ ซึ่งแต่ละช่องจะใช้เวลา  $O(1)$  และโดยรวมแล้วก็จะใช้เวลา  $O(N)$



## 5. A Kid and Dominoes by bT33

สำหรับโจทย์ข้อนี้ ผู้แต่งโจทย์ได้คิดวิธีทำไว้ 2 วิธีดังนี้

### วิธีที่ 1

ในวิธีแรก เราจะแบ่งอาร์เรย์ขนาด  $N$  ออกเป็นช่วงหลาย ๆ ช่วง

ให้น้อยที่สุดโดยในแต่ละช่วงจะเรียงจาก น้อยไปมากเสมอ แล้วจะสามารถแบ่งเป็นกรณีต่าง ๆ

ที่อาจเกิดขึ้นได้ ดังต่อไปนี้

- ขยายช่วงใดช่วงหนึ่งที่ปลายด้านใดด้านหนึ่ง คือ สำหรับแต่ละช่วงใด ๆ  
เราจะดูว่าที่ปลายของช่วงด้านใดด้านหนึ่งสามารถหาโดมิโนที่ไม่ได้อยู่ในช่วงนั้นมาต่อที่ปลายด้านนั้นได้หรือไม่
- ขยายช่วงใดช่วงหนึ่งที่ปลายทั้งสองข้าง คือ สำหรับแต่ละช่วงใด ๆ  
เราจะดูว่าเมื่อสลับโดมิโนอันที่ไม่ได้อยู่ในช่วงนั้นและอยู่ติดกับปลายทั้งสองข้าง  
แล้วสามารถช่วงที่ปลายทั้งสองได้หรือไม่
- ผสานสองช่วง คือ การนำสองช่วงใด ๆ ที่อยู่ติดกันมารวมกันเป็นช่วงเดียว  
ซึ่งจะพิจารณาที่ตัวมากที่สุดของช่วงซ้ายกับตัวน้อยสุดของช่วงขวา โดยลองสลับทั้งสองอัน  
หรือสลับอันใดตัวหนึ่งในสอง/อันนั้นกับอีกอันที่ไม่ได้อยู่ในทั้งสองช่วง
- ผสานสองช่วงพร้อมกับขยายปลายด้านใดด้านหนึ่ง คือ การนำสองช่วงใด ๆ มารวมกัน  
แล้วสามารถขยายช่วงที่รวมกันแล้วที่ปลายด้านใดด้านหนึ่งได้  
โดยดูได้จากการลองสลับอันมากที่สุดของช่วงซ้ายกับอันน้อยสุดของช่วงถัดจากช่วงขวา  
หรือลองสลับอันน้อยสุดของช่วงขวากับอันมากที่สุดของช่วงก่อนหน้าช่วงซ้าย
- ผสานสามช่วง คือ การนำสามที่อยู่ติดกันมารวมกันเป็นช่วงเดียว  
โดยดูได้จากการลองสลับอันมากที่สุดของช่วงซ้ายกับอันมากที่สุดของช่วงขวา

โดยวิธีนี้มีเวลาการทำงานเป็น  $O(N)$

### วิธีที่ 2

โดยวิธีนี้ ผู้แต่งโจทย์จะใช้วิธีการค้นหาแบบทวิภาค (Binary Search)

กำหนดให้  $k$  = จำนวนโดมิโนของโดมิโนช่วงหนึ่งที่อยู่ติดกัน ซึ่งถ้า  $k$

มีค่ามากกว่าคำตอบของโจทย์ข้อนี้แล้วจะไม่สามารถทำให้เกิดโดมิโนช่วงหนึ่งที่อยู่ติดกันจำนวน  $k$  อัน  
ที่มีน้ำหนักเรียงจากน้อยไปมากได้ตามที่โจทย์ต้องการ ทำให้เราสามารถ Binary Search บน  $k$   
เพื่อหาคำตอบของโจทย์ข้อนี้ได้

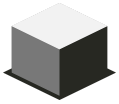
สำหรับค่า  $k$  ใด ๆ เราจะทำการตรวจสอบว่า สามารถทำให้เกิดโดมิโน  $k$

ตัวติดกันที่เรียงจากน้อยไปมาก โดยการสลับไม่เกิน 1 ครั้ง ได้หรือไม่ ถ้าทำได้แล้วเราก็จะไปหาค่าที่มากขึ้น  
แต่ถ้าไม่แล้วเราก็จะไปหาค่า  $k$  ที่น้อยลง

โดยผู้แต่งจะขอแบ่งการกระทำในการทำให้เกิดช่วงโดมิโน  $k$  ตัวติดกัน เรียงจากน้อยไปมาก เป็น 3 แบบ  
ได้แก่

- แบบที่ 1 : ไม่ต้องสลับ คือ มีช่วงใดช่วงหนึ่งที่ยาว  $k$  เรียงจากน้อยไปมากอยู่แล้ว  
ซึ่งรูปแบบนี้สามารถตรวจสอบได้โดยง่าย
- แบบที่ 2 : สลับภายในช่วง คือ มีช่วงใดช่วงหนึ่งที่มีความยาว  $k$  เมื่อสลับโดมิโนคู่ใดคู่หนึ่ง  
แล้วทำให้ช่วงนั้นเรียงจากน้อยไปมาก  
ซึ่งจะทำได้ก็ต่อเมื่อในช่วงนั้นมีโดมิโนคู่ติดกันที่อันซ้ายหนักกว่าอันขวาอยู่ไม่เกิน 2 คู่  
ซึ่งในกรณีที่มีคู่เดียว เราก็จะสลับคู่ นั้น  
แต่ถ้ามีสองคู่ให้ลองสลับตัวที่หนักมากกว่าของคู่ทางซ้ายกับตัวที่หนักน้อยกว่าของคู่ทางขวา  
แล้วดูว่าเรียงจากน้อยไปมากหรือไม่
- แบบที่ 3 : สลับกับข้างนอกช่วง คือ มีช่วงใดช่วงหนึ่งที่มีความยาว  $k$   
เมื่อสลับอันที่อยู่ในช่วงกับอันที่ไม่ได้อยู่ในช่วง แล้วทำให้ช่วงนั้นเรียงจากน้อยไปมาก  
ซึ่งจะทำได้ก็ต่อเมื่อในช่วงนั้นมีโดมิโนคู่ติดกันที่อันซ้ายหนักกว่าอันขวาอยู่เพียงคู่เดียว  
โดยเราจะต้องเลือกโดมิโนอันใดอันหนึ่งที่ไม่อยู่ในช่วงที่เมื่อนำมาสลับกับอันใดอันหนึ่งในคู่ นั้น  
แล้วเรียงจากน้อยไปมาก

โดยวิธีนี้จะใช้เวลาการทำงานเป็น  $O(N \log N)$  แต่ว่าผู้ออกโจทย์คิดว่าวิธีนี้เขียนได้ง่ายกว่าวิธีที่ 2  
เพราะมีจำนวนกรณีน้อยกว่า



## 6. Mahou Shoujo by Jumpwmk

---

ในโจทย์ข้อนี้เราสามารถแก้ Dynamic Programming มาแก้ได้ โดยจะมีสมการว่า

$$dp[i] = \min_{j=i-k+1 \rightarrow i} \left( dp[j-1] + \max_{d=j \rightarrow i} a_d \right)$$

จากสมการข้างต้น จะเห็นว่าต้องใช้เวลาคำนวณเป็น  $O(N^3)$  แต่สามารถทำให้เหลือ  $O(N^2)$  ได้โดยง่าย แต่ว่าจะไม่เพียงพอสำหรับโจทย์ข้อนี้

สำหรับโจทย์ข้อนี้ เราจำเป็นต้อง optimize การทำ Dynamic Programming โดยอาจใช้ Containers ในภาษา C++ เช่น deque กับ set ซึ่งเราจะใช้ deque ในการหาว่า สำหรับ  $j$  ( $i - k < j \leq i$ ) ค่าของ  $\max\{a_j, a_{j+1}, \dots, a_i\}$  เป็นเท่าไร และใช้ set ในการหาว่าค่า  $j$  ค่าใดที่ทำให้  $dp[j-1] + \max\{a_j, a_{j+1}, \dots, a_i\}$  มีค่าน้อยที่สุด ซึ่งทำให้มีเวลาการทำงานเป็น  $O(N \log N)$