



ภาษาแพะ

แพะนั้นเป็นสิ่งมีชีวิตที่มหัศจรรย์ แพะชอบวิ่งเล่นไปมาในสนามหญ้า แต่แพะมีศัตรูมาก แพะจึงมีความสามารถพิเศษในการวิ่งไปซ่อนในสนามหญ้า กล่าวคือ เมื่อแพะมองเห็นสนามหญ้า (กราฟ) แล้ว แพะจะสามารถนับจำนวนเส้นทางทั้งหมดที่เริ่มจากจุดยอด s และไปจบที่จุดยอด t ที่มีความยาว n ได้ (ภายใน $\text{mod } 10^9 + 7$) ทันที

คุณกำลังสนทนากับต้นมะนาวอยู่ การสนทนาเป็นการตั้งคำถามกันไปมา คุณอยากจะถามต้นมะนาวว่า "ต้นมะนาวเคยเข้าไปนอนในถังน้ำแข็งหรือเปล่า?" แต่ต้นมะนาวตั้งเงื่อนไขกลับมาว่า ต้นมะนาวจะตอบก็ต่อเมื่อคุณแก้ปัญหาคณิตศาสตร์บางอย่าง (หลายอย่าง) ให้กับต้นมะนาวเสียก่อน คำถามของต้นมะนาว จะอยู่ในรูปแบบ "สำหรับแต่ละ n จงหาค่า..." เสมอ โดยต้นมะนาวต้องการให้คุณตอบคำถามเหล่านี้สำหรับค่า n ตั้งแต่ 1 ถึง 100 แน่นอนว่าคำถามเหล่านั้นคุณสามารถเขียนโปรแกรมได้ แต่น่าเสียดายเหลือเกินที่อยู่ดี ๆ เกิดไฟดับขึ้นมา! แต่ไม่ต้องห่วง แพะนั้นอยู่เคียงข้างคุณเสมอ!

คุณสามารถวาดสนามหญ้า (กราฟระบุทิศทาง) โดยอาจมีวงวน (self-loop) หรือเส้นเชื่อมซ้ำ (parallel edge) ได้ และส่งภาพวาดนี้ให้กับแพะ แล้วแพะจะตอบว่า สำหรับจำนวนนับ n ตั้งแต่ 1 ถึง 100 นั้นมีเส้นทางความยาว n ที่เริ่มจาก s แล้วไปจบที่ t ทั้งหมดกี่เส้นทาง

อย่างไรก็ตาม แพะติดประชุมด่วนกับกลุ่มสมาคมการค้าครัวของแห่งประเทศไทยกะลาแลนด์ เพราะแพะได้รับแจ้งมาว่าครัวของชาติตลาด! โอ้ว น่าเศร้ายิ่งนัก นอกจากไฟดับแล้ว ครัวของยังชาติตลาด ต้องรีบหารือกับสมาคมการค้าครัวของอีกทำให้แพะโผล่มาช่วยคุณได้แค่เพียงช่วงสั้น ๆ เท่านั้น คุณจึงต้องรีบส่งแปลนสนามหญ้า จุดเริ่มต้น และจุดจบ ซึ่งก็คือสามสิ่งอันดับ $\langle G, s, t \rangle$ ให้กับแพะ แล้วแพะจะคำนวณคำตอบตามข้างต้น แต่แพะมีเวลาไม่มากพอที่จะมานั่งคุยกับคุณ แพะจึงจะส่งคำตอบนี้ไปยังต้นมะนาววิเศษโดยตรง แล้วหนีหายไปเช่นเคยในพริบตา

คุณไม่อยากให้คำตอบผิด ๆ ให้กับต้นมะนาววิเศษ งานของคุณจึงเป็นการออกแบบ $\langle G, s, t \rangle$ ที่สอดคล้องกับเงื่อนไข คุณสามารถออกแบบ G อย่างไม่รู้ขีดจำกัดจำนวนจุดยอดของ G ต้องมีค่าไม่เกิน 40 ส่วนจำนวนเส้นเชื่อมนั้นเป็นเท่าไรก็ได้ (ที่สามารถใส่ในตัวแปรชนิด `int` ได้ และห้ามติดลบ) เพื่อความง่ายเราจะเรียกจุดยอดเป็นหมายเลขตั้งแต่ 0 ถึง $N - 1$ เมื่อ N แทนจำนวนจุดยอดทั้งหมด นั่นทำให้ s และ t จะต้องเป็นจำนวนเต็มระหว่าง 0 ถึง $N - 1$ เช่นกัน (เป็นไปได้ว่า $s = t$)

คุณจะต้องส่งออกข้อมูลของกราฟ G ในรูปแบบ adjacency matrix โดยตัวเลขในแถวที่ i และคอลัมน์ที่ j แทนจำนวนเส้นเชื่อมจาก i ไป j (i อาจเท่ากับ j ได้) สำหรับ $0 \leq i, j < N$ ดังนั้น ในทุกปัญหาย่อย ฟังก์ชันจะต้องคืนค่าชนิด `tuple<vector<vector<int>>, int, int>` ระบุกราฟ (เวกเตอร์ขนาด N ที่ประกอบด้วยเวกเตอร์ขนาด N ในแต่ละช่อง โดยแต่ละช่องระบุจำนวนเส้นเชื่อม), ระบุจุดเริ่มต้น และระบุจุดสิ้นสุด ตามลำดับ

ตัวอย่าง

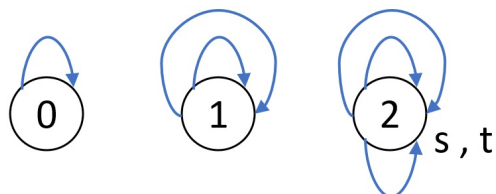
เพื่อความง่ายเราจึงมีตัวอย่างให้สองตัวอย่าง

ตัวอย่าง 1

ตัวอย่างแรกสอดคล้องกับฟังก์ชัน example1 โดยจะตอบคำถามดังนี้

$S :=$ "สำหรับแต่ละ n จงหาค่าของเศษจากการหาร 3^n ด้วย $10^9 + 7$ "

ฟังก์ชันคืนค่า $\left\langle \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}, 2, 2 \right\rangle$ ซึ่งสอดคล้องกับกราฟดังต่อไปนี้



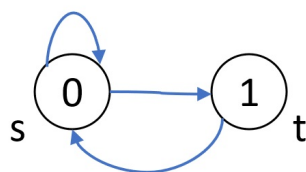
สังเกตว่า ไม่ว่า n จะเป็นอะไรก็ตาม (ตั้งแต่ 1 ถึง 100) แพะจะตอบถูกต้อง

ตัวอย่าง 2

ตัวอย่างที่สองสอดคล้องกับฟังก์ชัน example2 โดยจะตอบคำถามดังนี้

$S :=$ "สำหรับแต่ละ n จงหาค่าของเศษจากการหารจำนวนฟีโบนัชชีตัวที่ n ด้วย $10^9 + 7$ หรือเขียนแทนด้วย $F_n \bmod (10^9 + 7)$ โดยนิยาม $F_0 = 0, F_1 = 1$ และ $F_n = F_{n-1} + F_{n-2}$ สำหรับทุกจำนวนนับ $n \geq 2$ "

ฟังก์ชันคืนค่า $\left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, 0, 1 \right\rangle$ ซึ่งสอดคล้องกับกราฟดังต่อไปนี้



สังเกตว่า ไม่ว่า n จะเป็นอะไรก็ตาม (ตั้งแต่ 1 ถึง 100) แพะจะตอบถูกต้องเสมอ

ปัญหาย่อย

ในข้อนี้จะมีหลายปัญหาย่อยทั้งหมด 6 ปัญหาซึ่งแยกออกจากกันอย่างสิ้นเชิง แต่ภายในปัญหาย่อยแต่ละปัญหา จะมีปัญหาย่อยของปัญหาย่อยที่เกี่ยวข้องกัน พิจารณารายการดังต่อไปนี้

ปัญหาย่อย 1 (2 คะแนน)

$S :=$ "ให้เมทริกซ์จัตุรัส A มา ขนาด $M \times M$ นอกจากนี้ให้ค่า r และ c มาด้วย ($0 \leq r, c < M$) สำหรับแต่ละ n จงหาค่าของ $(A^n)_{r,c} \bmod (10^9 + 7)$ " กล่าวคือ หาค่าของเมทริกซ์ A^n ในช่องแถวที่ r คอลัมน์ที่ c (เราเรียกเมทริกซ์ตั้งแต่แถวที่ 0 ถึงแถวที่ $M - 1$ และคอลัมน์ตั้งแต่ 0 ถึง $M - 1$ เช่นเดียวกัน)

คุณจะต้องเขียนฟังก์ชัน

```
tuple<vector<vector<int>>, int, int> task1(vector<vector<int>> A, int
r, int c)
```

โดยข้อมูล A , r และ c แทนเมทริกซ์ A , จำนวนเต็ม r และ c ตามลำดับ ($A[i][j]$ คือ $A_{i,j}$ สำหรับทุก $0 \leq i, j < M$) รับประกันว่า $-10 \leq A_{i,j} \leq 10$ และ $A_{i,j}$ เป็นจำนวนเต็ม สำหรับทุก $0 \leq i, j < M$ และรับประกันว่า $1 \leq M \leq 40$

สำหรับปัญหาย่อยนี้ จะมีการให้คะแนนบางส่วน ตามเงื่อนไขดังต่อไปนี้

1. (1 คะแนน) $A_{i,j} \geq 0$ สำหรับทุก $0 \leq i, j < M$
2. (1 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ปัญหาย่อย 2 (6 คะแนน)

$S :=$ "ให้กราฟ H เป็นกราฟระบุทิศทาง โดยอาจมีวงวนหรือเส้นเชื่อมซ้ำได้ โดยมอบมาในรูปแบบ adjacency matrix โดยตัวเลขในแถวที่ i และคอลัมน์ที่ j แทนจำนวนเส้นเชื่อมจาก i ไป j (i อาจเท่ากับ j ได้) สำหรับแต่ละ n จงนับเศษจากการหารจำนวนเส้นทางทั้งหมดจาก a ไป b ที่ ความยาวไม่เกิน n ด้วย $10^9 + 7$ "

คุณจะต้องเขียนฟังก์ชัน

```
tuple<vector<vector<int>>, int, int> task2(vector<vector<int>> H, int
a, int b)
```

โดยข้อมูล H , a และ b แทนเมทริกซ์ H , จำนวนเต็ม a และ b ตามลำดับ ($H[i][j]$ คือ $H_{i,j}$ สำหรับทุก $0 \leq i, j < V$ เมื่อ V แทนจำนวนจุดยอดของ H) รับประกันว่า $0 \leq H_{i,j} \leq 100$ สำหรับทุก $0 \leq i, j < V$ และรับประกันว่า $1 \leq V \leq 40$

ปัญหาย่อย 3 (35 คะแนน)

$S :=$ "ให้พหุนาม $P(x)$ ที่มีสัมประสิทธิ์เป็นจำนวนเต็มระหว่าง 0 ถึง 5 ดีกรีไม่เกิน 10 จงหา $P(n) \bmod (10^9 + 7)$ สำหรับแต่ละ n "

คุณจะต้องเขียนฟังก์ชัน

```
tuple<vector<vector<int>>, int, int> task3(vector<int> a)
```

โดยข้อมูล a แทนสัมประสิทธิ์ของ P ($a[i]$ คือ a_i สำหรับทุก $0 \leq i \leq \deg P$) เมื่อ

$$P(x) = \sum_{i=0}^{\deg P} a_i x^i$$

รับประกันว่าสัมประสิทธิ์ของ $x^{\deg P}$ ซึ่งก็คือ $a_{\deg P}$ จะมีค่าไม่เท่ากับศูนย์

สำหรับปัญหาย่อยนี้ จะมีการให้คะแนนบางส่วน ตามเงื่อนไขดังต่อไปนี้

1. (5 คะแนน) $\deg P = 1; a_1 = 1; a_0 = 0$
2. (5 คะแนน) $\deg P = 1; a_1 = 1$
3. (5 คะแนน) $\deg P = 1; a_0 = 0$
4. (5 คะแนน) $\deg P = 2; a_1 = 0; a_0 = 0$
5. (15 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ปัญหาย่อย 4 (17 คะแนน)

$S :=$ "สำหรับแต่ละ n จงนับจำนวนบิตสตริง (สตริงที่ประกอบด้วย 0 หรือ 1 เท่านั้น) ความยาว n ที่ไม่ปรากฏสตริงย่อย 101 อยู่เลย"

คุณจะต้องเขียนฟังก์ชัน

```
tuple<vector<vector<int>>, int, int> task4()
```

ปัญหาย่อย 5 (25 คะแนน)

$S :=$ "ให้บิตสตริง D ความยาว M ไป เรียกเป็นช่องตั้งแต่ช่องที่ 0 ถึงช่องที่ $M - 1$ สำหรับแต่ละ n จงหา D_{n-1} " สำหรับปัญหาย่อยนี้จะตรวจสอบเพียง n ตั้งแต่ 1 ถึง M เท่านั้น (ต่างจากปัญหาย่อยอื่นที่จะตรวจสอบตั้งแต่ 1 ถึง 100)

คุณจะต้องเขียนฟังก์ชัน

```
tuple<vector<vector<int>>, int, int> task5(vector<int> D)
```

โดยข้อมูล D แทนบิตสตริง D กล่าวคือ $D[i] = D_i$ สำหรับทุกจำนวนเต็ม $0 \leq i < M$

สำหรับปัญหาย่อยนี้ จะมีการให้คะแนนบางส่วน ตามเงื่อนไขดังต่อไปนี้

1. (5 คะแนน) $M = 2$
2. (5 คะแนน) $2 \leq M \leq 4$
3. (15 คะแนน) $2 \leq M \leq 10$

ปัญหาย่อย 6 (15 คะแนน)

$S :=$ "ให้จำนวนนับ $m \geq 2$ ไป สำหรับแต่ละ n จงหาว่า n หารด้วย m ลงตัวหรือไม่ หากลงตัวให้ตอบ 1 หากไม่ลงตัวให้ตอบ 0"

คุณจะต้องเขียนฟังก์ชัน

```
tuple<vector<vector<int>>, int, int> task6(int m)
```

โดยข้อมูล m แทนจำนวนเต็ม m

สำหรับปัญหาย่อยนี้ จะมีการให้คะแนนบางส่วน ตามเงื่อนไขดังต่อไปนี้

1. (5 คะแนน) $m = 2$
2. (5 คะแนน) $m = 7$
3. (5 คะแนน) $2 \leq m \leq 10$

เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างจะรับข้อมูลนำเข้าบรรทัดแรกเป็นชื่อฟังก์ชันที่จะทำการเรียก ซึ่งจะต้องเป็นอย่างใดอย่างหนึ่งในรายการ example1, example2, task1, task2, task3, task4, task5, task6 ต่อมาอาจทำการรับข้อมูลนำเข้าเพิ่มเติม ขึ้นอยู่กับฟังก์ชันที่เรียกใช้งาน โดยมีรายละเอียดดังต่อไปนี้

example1

ไม่มีข้อมูลนำเข้าเพิ่มเติม

example2

ไม่มีข้อมูลนำเข้าเพิ่มเติม

task1

บรรทัดต่อมา ระบุ N, r, c คั่นด้วยช่องว่าง ต่อมาอีก N บรรทัด แต่ละบรรทัดระบุจำนวนเต็ม N ตัว โดยในบรรทัดที่ $2 + i$ จำนวนที่ j แทน $A[i-1][j-1]$

task2

บรรทัดต่อมา ระบุ N, a, b คั่นด้วยช่องว่าง ต่อมาอีก N บรรทัด แต่ละบรรทัดระบุจำนวนเต็ม N ตัว โดยในบรรทัดที่ $2 + i$ จำนวนที่ j แทน $H[i-1][j-1]$

task3

บรรทัดต่อมา ระบุ D แทนดีกรีของพหุนาม P บรรทัดหลังจากนั้น ระบุจำนวนเต็ม $D + 1$ ตัว ตัวที่ i ระบุสัมประสิทธิ์ของเอกนามดีกรี $i - 1$ ของ P

task4

ไม่มีข้อมูลนำเข้าเพิ่มเติม

task5

บรรทัดต่อมา ระบุบิตสตริง D' ความยาวไม่เกิน 10 ในรูปแบบข้อความปกติ ตัวอักษรประกอบด้วย 0 หรือ 1 โดยตัวอักษรติดกันทั้งหมด

task6

บรรทัดต่อมา ระบุจำนวนเต็มตัวเดียวคือ m

การให้คะแนนในระบบ

ภายในระบบจะมีปัญหาย่อยทั้งหมด 15 ปัญหาย่อย ซึ่งสอดคล้องกับการให้คะแนนบางส่วนในแต่ละปัญหาย่อย ตารางดังต่อไปนี้นี้จะแสดงว่าหมายเลขของปัญหาย่อยในระบบสอดคล้องกับปัญหาย่อยใดในโจทย์

ปัญหาย่อยในระบบ	ปัญหาย่อยในโจทย์.การให้คะแนนบางส่วน	คะแนน
Subtask 1	1.1	1
Subtask 2	1.2	1
Subtask 3	2	6
Subtask 4	3.1	5
Subtask 5	3.2	5
Subtask 6	3.3	5
Subtask 7	3.4	5
Subtask 8	3.5	15
Subtask 9	4	17
Subtask 10	5.1	5
Subtask 11	5.2	5
Subtask 12	5.3	15
Subtask 13	6.1	5
Subtask 14	6.2	5
Subtask 15	6.3	5

ขอบเขต

- Time limit: 0.5 seconds
- Memory limit: 512 MB