



แข่งแกร่งด้วยระเบิด (bombs)

คุณอยู่ในโลก 1 มิติ สามารถมองเป็นเส้นจำนวนจาก $-\infty$ ถึง ∞ เมื่อเวลาเริ่มต้น (วินาทีที่ 0) คุณสามารถเลือกเกิด ณ ตำแหน่งไหนก็ได้และคุณจะถูกพิชิตเป็นจำนวนเต็มเท่านั้น

ในโลกนี้มีระเบิดที่ค่อย ๆ ปรากฏขึ้นจำนวน N ลูก สำหรับ $1 \leq i \leq N$, ระเบิดลูกที่ i จะปรากฏที่พิกัด x_i จะระเบิดในเวลา t_i และมีความรุนแรง a_i หน่วย ถ้าในเวลาที่เราเปิดลูกนี้ระเบิดขึ้น คุณอยู่ที่ตำแหน่ง k คุณจะเสีย HP เท่ากับ $a_i - |x_i - k|$ หน่วย สังเกตว่าค่านี้อาจจะเป็นลบได้ นั่นคือ ถ้า $|x_i - k|$ มีค่ามากพอ แทนที่คุณจะเสีย HP คุณจะได้ HP เพิ่มจากระเบิดลูกนี้ (แต่นั่นไม่ใช่เป้าหมายที่คุณต้องการ!!)

คุณเป็นคนที่มีความมุ่งมั่นอยากเป็นผู้แข็งแกร่ง แต่คุณมีความเชื่อที่ประหลาด กล่าวคือ คุณเชื่อว่ายิ่งบาดเจ็บจะยิ่งแข็งแกร่ง เป้าหมายชีวิตคุณก็คืออยากเสีย HP ให้ **มากที่สุด** เท่าที่จะทำได้

ทุกครั้งที่ระเบิดปรากฏขึ้น ให้คุณตอบ HP ที่เสียรวม **มากที่สุด** เท่าที่เป็นไปได้ รับประกันว่าระเบิดลูกที่ทยอยเพิ่มขึ้นมาจะระเบิดก่อนระเบิดลูกก่อนหน้า นั่นคือ $t_i \leq t_{i+1}$ สำหรับทุก ๆ $1 \leq i < N$

ระหว่างการถาม จะมีการระบุความเร็ว p_i ในการเคลื่อนที่ของคุณ ความเร็วดังกล่าวจะเป็นความเร็วในการเคลื่อนที่ของคุณจากช่วงเวลาที่คุณถามก่อนหน้าถึงเวลา t_i กล่าวคือ สมมติให้ j เป็นดัชนีที่น้อยกว่า i ที่มากที่สุดที่ $t_j \neq t_i$ คุณสามารถเดินระหว่างเวลา t_j ถึง t_i ได้เท่ากับ $p_i \cdot (t_i - t_j)$ หน่วย

ความเร็วนี้อาจจะเปลี่ยนไปช่วงเวลาได้ด้วย แต่รับประกันว่าถ้า $t_i = t_{i+1}$ แล้ว $p_i = p_{i+1}$

พิจารณาตัวอย่างต่อไปนี้ สมมติว่า $N = 6$ และมีลูกระเบิดดังนี้

ลูกที่	x_i	t_i	a_i	p_i
1	30	0	6	1
2	35	0	10	1
3	25	1	5	1
4	10	3	10	20
5	15	3	7	20
6	20	3	8	20

เมื่อมีระเบิดลูกที่ 1 ตำแหน่งที่ดีที่สุดที่คุณควรจะอยู่คือตำแหน่ง 30 ซึ่งทำให้คุณเสีย HP เท่ากับ

$$6 - |30 - 30| = 6$$

หน่วย

เมื่อเพิ่มระเบิดลูกที่ 2 ซึ่งระเบิดในเวลาเดียวกัน ตำแหน่งที่ดีที่สุดคือตำแหน่งใดก็ได้ระหว่าง 30 - 35 ซึ่งจะทำให้

คุณเสีย HP รวมเท่ากับ 11 หน่วย

เมื่อมีระเบิด 3 ลูก โดยที่ลูกที่ 3 ระเบิดที่เวลา $t_3 = 1$ คุณอาจจะเริ่มต้นที่จุด 30 และเสีย HP 11 หน่วย จากนั้นเนื่องจาก $p_3 = 1$ คุณสามารถเดินมาที่ตำแหน่ง 29 ในวินาทีที่ 1 เพื่อเสีย HP เท่ากับ $5 - |25 - 29| = 1$ รวมเสีย HP เท่ากับ 12 หน่วย

เมื่อมีระเบิด 4 ลูก สังเกตว่าลูกนี้ระเบิดที่เวลา 2 และ $p_4 = 20$ และ $t_4 = 3$ คุณสามารถเดินระหว่างเวลาที่ 1 ถึง 3 ได้เท่ากับ $2 \cdot 20 = 40$ หน่วย ทางเลือกที่ดีที่สุดนั้นคือในช่วงแรกให้ทำแบบเดิมและในวินาทีที่ 3 เดินมาที่ตำแหน่ง 10 แต่ครั้งนี้คุณจะเสีย HP เท่ากับ $10 - |10 - 10| = 10$ หน่วย รวมเสีย HP เท่ากับ 22 หน่วย

เมื่อมีระเบิด 5 ลูก สังเกตว่า $t_5 = t_4 = 3$ ระเบิดลูกนี้ระเบิดในเวลาเดียวกับลูกที่แล้ว ในกรณีนี้ ที่เวลา 3 คุณควรจะอยู่ในตำแหน่งใดก็ได้ระหว่าง 10 ถึง 15 (จะอยู่ที่ 10 เหมือนตอนมี 4 ลูกก็ได้) ทางเลือกดังกล่าวทำให้คุณเสีย HP รวมทั้งหมดมากที่สุดได้ 24 หน่วย

เมื่อมีระเบิด 6 ลูก สังเกตว่า $t_6 = t_5 = t_4 = 3$ ระเบิดลูกนี้ระเบิดในเวลาเดียวกับสองลูกที่แล้ว ในกรณีนี้ ที่เวลา 3 คุณควรจะอยู่ในตำแหน่ง 15 ซึ่งความเร็วจากเวลา 1 ถึงเวลา 3 ทำให้คุณสามารถเดินมาได้ ทางเลือกดังกล่าวทำให้คุณเสีย HP รวมทั้งหมดมากที่สุดได้ 27 หน่วย

รายละเอียดการเขียนโปรแกรม

คุณจะต้องเขียนฟังก์ชันสองฟังก์ชันต่อไปนี้

```
void initialize(int N)
```

- ฟังก์ชันนี้จะถูกเรียกหนึ่งครั้ง เพื่อให้คุณเตรียมค่าเริ่มต้น
- ฟังก์ชันจะระบุ N จำนวนลูกระเบิด

จากนั้นเกรดเดอร์จะทยอยเรียกฟังก์ชันด้านล่าง จำนวน N ครั้ง

```
long long max_hp_loss(int X, int T, int A, int P)
```

- จะมีการเรียกทั้งสิ้น N ครั้ง
- ในการเรียกครั้งที่ i จะเป็นการระบุข้อมูลของระเบิดลูกที่ i กล่าวคือ $x_i = X, t_i = T, a_i = A$
- ในคำถามนี้คุณจะเคลื่อนที่ได้ $p_i = P$ หน่วยต่อวินาทีนับจากช่วงเวลาก่อนหน้า (นั่นคือนับจากวินาทีที่ t_j เมื่อ t_j เป็นเวลามากที่สุดที่น้อยกว่า t_i)
- คุณจะต้องคืนค่า HP ที่สูญเสียรวม **มากที่สุด** ที่เป็นไปได้ถ้ามีระเบิดตั้งแต่ลูกที่ 1 ถึงลูกที่ i และระเบิดได้ระเบิดหมดแล้ว

เงื่อนไข

- $2 \leq N \leq 200\,000$
- $1 \leq X \leq 10^9$
- $1 \leq T \leq 10^3$
- $1 \leq A \leq 10^9$

- $1 \leq P \leq 10^3$
- $t_i \leq t_{i+1}$ สำหรับ $1 \leq i \leq N - 1$
- ถ้า $t_i = t_{i+1}$ แล้ว $p_i = p_{i+1}$ สำหรับ $1 \leq i \leq N - 1$

ปัญหาย่อย

1. (3 points) $N \leq 1\,000, T = 1, P = 1$
2. (3 points) $X \leq 100, T = 1, P = 1$
3. (4 points) $N \leq 1\,000, X \leq 100$
4. (4 points) $N \leq 1\,000, T \leq 10$
5. (4 points) $N \leq 1\,000, P \leq 10$
6. (5 points) $T \leq 10, P \leq 10$
7. (5 points) $T = 1$
8. (23 points) $P = 1$
9. (11 points) $N \leq 10$
10. (12 points) $N \leq 100$
11. (26 points) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

จากตัวอย่างข้างต้น เกรดเดอร์จะเริ่มโดยเรียก

```
initialize(4)
```

หลังจากนั้นจะเรียก `max_hp_loss` อีก 6 ครั้งดังนี้

```
max_hp_loss(30, 0, 6, 1)
```

ซึ่งจะต้องคืนค่า 6

ครั้งที่ 2 จะเรียก

```
max_hp_loss(35, 0, 10, 1)
```

ซึ่งจะต้องคืนค่า 11 จากนั้นสำหรับระเบิดลูกที่ 3 จะเรียก

```
max_hp_loss(25, 1, 5, 1)
```

ฟังก์ชันที่ทำงานถูกต้องจะต้องคืนค่า 12 สำหรับระเบิดลูกที่ 4 เกรดเดอร์จะเรียก

```
max_hp_loss(10, 3, 10, 20)
```

ซึ่งจะคืนค่า 22 สำหรับระเบิดลูกที่ 5 เกรดเดอร์จะเรียก

```
max_hp_loss(15, 3, 7, 20)
```

ซึ่งจะคืนค่า 24 สำหรับระเบิดลูกที่ 6 เกรดเดอร์จะเรียก

```
max_hp_loss(20, 3, 8, 20)
```

ซึ่งจะคืนค่า 27

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่าง อ่านข้อมูลนำเข้าในรูปแบบต่อไปนี้:

- บรรทัดที่ 1: N
- บรรทัดที่ $1 + i$ ถึง $1 + N$ สำหรับ $1 \leq i \leq N$: $x_i \ t_i \ a_i \ p_i$

เกรดเดอร์ตัวอย่างจะพิมพ์ผลลัพธ์ที่ได้จาก `max_hp_loss`

ขีดจำกัด

- Time limit: 1 seconds
- Memory limit: 1024 MB