Adelantamiento

Hay una ruta de un solo sentido y un solo carril desde el aeropuerto de Budapest hasta el Hotel Forrás. La ruta tiene L kilómetros de largo. Durante la IOI 2023, $N+1$ micros de traslado atraviesan esta ruta. Los micros están enumerados desde 0 a N. El micro i ($0 \leq i < N$) deja el aeropuerto en el instante $T[i]$ (medido en segundos desde el inicio del evento), y puede viajar 1 kilómetro en $W[i]$ segundos. El micro N es un micro de reserva que puede viajar 1 kilómetro en X segundos. Todavía no se decidió el instante Y en el que este micro debe dejar el aeropuerto.

En general, no está permitido adelantar en la ruta, pero los micros tienen permitido adelantarse entre sí en **estaciones de ordenamiento**. Hay M ($M > 1$) estaciones de ordenamiento, enumeradas desde 0 a $M-1$, en diferentes posiciones de la ruta. La estación de ordenamiento j ($0 \leq j < M$) está ubicada a $S[j]$ kilómetros del aeropuerto a lo largo de la ruta. Las estaciones de ordenamiento se ordenan de forma creciente por la distancia desde el aeropuerto, esto es, $S[j] < S[j+1]$ para cada $0 \leq j \leq M-2$. La primera estación de ordenamiento está ubicada en el aeropuerto y la última es el hotel, esto es $S[0] = 0$ y $S[M-1] = L$

Cada micro viaja a máxima velocidad a menos que este se encuentre con un micro más lento viajando delante de él en la ruta, en cuyo caso se amontonan y se ven forzados a viajar a la velocidad del micro más lento hasta alcanzar la siguiente estación de ordenamiento. Allí, los micros más rápidos se adelantarán a los micros más lentos.

Formalmente, para cada i y j tal que $0 \leq i \leq N$ y $0 \leq j < M$, el instante $t_{i,j}$ (en segundos desde el inicio del evento) en el que el micro i **llega a la** estación de ordenamiento j se define como sigue: Sea $t_{i,0} = T[i]$ para cada $0 \leq i < N$, y sea $t_{N,0} = Y$. Para cada j tal que $0 < j < M$:

- Se define el **instante esperado de llegada** (en segundos desde el inicio del evento) del micro i en la estación de ordenamiento j, denotado por $e_{i,j}$, como el instante en el que el micro i debería llegar a la estación j si estaba viajando a máxima velocidad desde el instante en el que llegó a la estación de ordenamiento $j-1$. Esto es, sea
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ para cada $0 \leq i < N$, y
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$. El micro i llega a la estación de ordenamiento j en el *máximo* de los instantes esperados de llegada del micro i y de todos los otros micros que llegaron a la estación $j-1$ antes que el micro i. Formalmente, sea $t_{i,j}$ el máximo entre $e_{i,j}$ y todos los $e_{k,j}$ con k tal que $0 \leq k \leq N$ y $t_{k,j-1} < t_{i,j-1}$.

Los organizadores de la IOI quieren planificar la salida del micro de reserva (micro N). Tu tarea es responder las Q preguntas de los organizadores, que son de la siguiente forma: dado el instante Y

(en segundos desde el inicio del evento) en el que se supone que el micro de reserva debe salir del aeropuerto, ¿en qué instante llegaría al hotel?

Detalles de Implementación

Tu tarea es implementar las siguientes funciones.

```
void init(int L, int N, int64[] T, int[] W, int X, int M, int[] S)
```

- L: el largo de la ruta.
- N: el número de micros que no son de reserva.
- T: un arreglo de tamaño N describiendo los instantes en que los micros salen del aeropuerto.
- W: un arreglo de tamaño N describiendo la máxima velocidad de los micros que no son de reserva.
- X: el tiempo que toma para el micro de reserva viajar 1 kilómetro.
- M: el número de estaciones de ordenamiento.
- S: un arreglo de tamaño M describiendo las distancias de las estaciones de ordenamiento desde el aeropuerto.
- Esta función es llamada exactamente una vez para cada caso de prueba, antes de cualquier llamada a `arrival_time`.

```
int64 arrival_time(int64 Y)
```

- Y: el instante en que se supone que el micro (micro N) de reserva debe salir del aeropuerto.
- Esta función deberá devolver el instante en que el micro de reserva llegaría al hotel.
- Esta función es llamada exactamente Q veces.

Ejemplo

Considera la siguiente secuencia de llamadas:

```
init(6, 4, [20, 10, 40, 0], [5, 20, 20, 30], 10, 4, [0, 1, 3, 6])
```

Ignorando el micro 4 (dado que todavía no ha sido planificado), la siguiente tabla muestra los instantes de llegada esperados y reales para los micros que no son de reserva a cada una de las estaciones de ordenamiento:

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	55
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180

Los instantes de llegada a la estación 0 son los instantes en los que los micros salen del aeropuerto aeropuerto. Esto es, $t_{i,0} = T[i]$ para $0 \leq i \leq 3$.

Los instantes esperados y reales de llegada a la estación 1 son computados como sigue:

- El instante esperado de llegada a la estación 1:
  - micro 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - micro 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - micro 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - micro 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Los instantes de llegada a la estación 1:
  - Los micros 1 y 3 llegan a la estación 0 antes que el micro 0, por lo tanto $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - El micro 3 llega a la estación 0 antes que el micro 1, por lo tanto $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - El micro 0, micro 1 and micro 3 llegan a la estación de ordenamiento 0 antes que el micro 2, por lo tanto $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Ningún micro llega a la estación 0 antes que el micro 3, por lo tanto $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

El micro 4 toma 10 segundos en viajar 1 kilómetro y está programado para dejar el aeropuerto al 0-ésimo segundo. En este caso, la siguiente tabla muestra los instantes de llegada para cada micro. El único cambio en cuanto a los instantes de llegada esperados y reales de los micros que no son de reserva está subrayado.

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	$\underline{60}$
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180
4		0		10	10		30	30		60	60

Podemos ver que el micro 4 llega al hotel al 60-ésimo segundo. Así, la función debería devolver 60

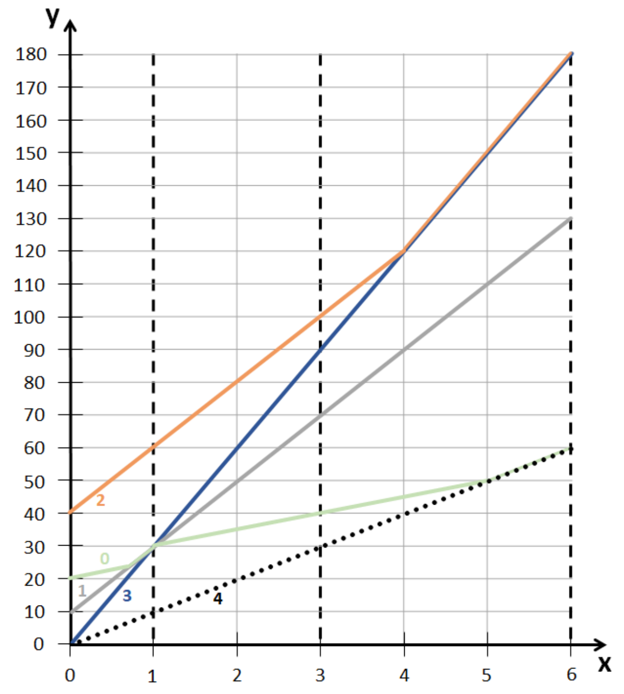
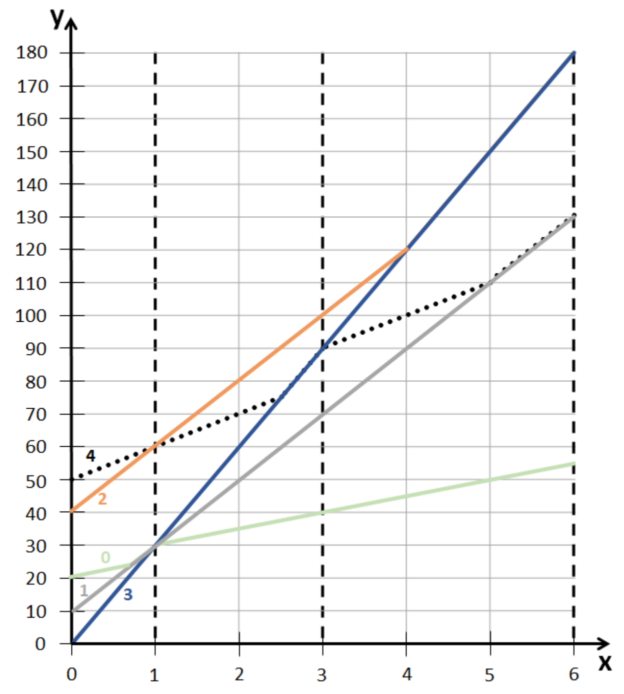
```
arrival_time(50)
```

El micro 4 ahora sale del aeropuerto al 50-ésimo segundo. En este caso, no hay cambios en los instantes de llegada para los micros que no son de reserva, comparado con la tabla inicial. Los instantes de llegada se muestran en la siguiente tabla.

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	55
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180
4		50		60	60		80	90		120	130

El micro 4 adelanta al micro más lento 2 en la estación de ordenamiento 1 dado que llegan al mismo tiempo. Luego, el micro 4 se agrupa con el micro 3 entre las estaciones 1 y 2, haciendo que el micro 4 llegue a la estación 2 al 90-ésimo segundo en vez de al 80-ésimo. Luego de dejar la estación 2, el micro 4 se agrupa con el micro 1 hasta que llegan al hotel. El micro 4 llega al hotel al 130-ésimo segundo. Así, la función debería devolver 130.

Podemos gráficar el tiempo que toma para cada micro llegar a cada distancia desde el aeropuerto. El eje X del gráfico representa la distancia desde el aeropuerto (en kilómetros) y el eje Y de el gráfico representa el tiempo (en segundos). Las líneas segmentadas (a rayitas) verticales marcan las posiciones de las estaciones de ordenamiento. Diferentes líneas sólidas (acompañadas de los índices de micros) representan los cuatro micros que no son de reserva. La línea negra punteada representa el micro de reserva.

`arrival_time(0)`	`arrival_time(50)`
	

Restricciones

- $1 \le L \le 10^9$
- $1 \le N \le 1\,000$
- $0 \le T[i] \le 10^{18}$ (para cada i tal que $0 \le i < N$)
- $1 \le W[i] \le 10^9$ (para cada i tal que $0 \le i < N$)
- $1 \le X \le 10^9$
- $2 \le M \le 1\,000$
- $0 = S[0] < S[1] < \cdots < S[M-1] = L$
- $1 \le Q \le 10^6$
- $0 \le Y \le 10^{18}$

Subtareas

1. (9 puntos) $N = 1, Q \le 1\,000$
2. (10 puntos) $M = 2, Q \le 1\,000$
3. (20 puntos) $N, M, Q \le 100$
4. (26 puntos) $Q \le 5\,000$
5. (35 puntos) Sin restricciones adicionales.

Evaluador local

El evaluador local lee la entrada en el siguiente formato:

- línea 1: $L \ N \ X \ M \ Q$
- línea 2: $T[0] \ T[1] \ \ldots \ T[N-1]$
- línea 3: $W[0] \ W[1] \ \ldots \ W[N-1]$
- línea 4: $S[0] \ S[1] \ \ldots \ S[M-1]$
- línea $5+k$ ($0 \leq k < Q$): Y para la pregunta k

El evaluador local imprime tus respuestas en el siguiente formato:

- línea $1+k$ ($0 \leq k < Q$): el valor retornado por `arrival_time` para la pregunta k.