Rebasando

Hay una carretera de un solo sentido desde el aeropuerto de Budapest hasta el Hotel Forrás. La carretera mide L kilómetros de largo.

Durante el evento IOI 2023, $N+1$ autobuses atraviesan esta carretera. Los autobuses están numerados del 0 al N. El autobús ordinario i ($0 \leq i < N$) planea salir del aeropuerto en el $T[i]$-ésimo segundo del evento, y puede recorrer 1 kilómetro en $W[i]$ segundos. El autobús N es un autobús sustituto que puede recorrer 1 kilómetro en X segundos. El tiempo Y cuando saldrá del aeropuerto aún no se ha decidido.

En general, no está permitido rebasar en esta carretera, pero los autobuses pueden rebasarse entre ellos en **estaciones de rebase**. Existen M ($M > 1$) estaciones de rebase, numeradas del 0 al $M-1$ en diferentes lugares de la carretera. La estación de rebase j ($0 \leq j < M$) se encuentra a $S[j]$ kilometros del aeropuerto por la carretera. Las estaciones de rebase estan ordenadas por distancia creciente desde el aeropuerto, es decir, $S[j] < S[j+1]$ para toda $0 \leq j \leq M-2$. La primera estación de rebase es el aeropuerto y la última es el hotel, es decir, $S[0] = 0$ y $S[M-1] = L$.

Cada autobús viaja a máxima velocidad a menos que alcance a un autobús más lento que viaja delante de él en la carretera, en cuyo caso se amontonan y se ven obligados a viajar a la velocidad del autobús más lento hasta llegar a la siguiente estación de rebase. Ahí, los autobuses más rápidos adelantarán a los más lentos.

Formalmente, para toda i y j tal que $0 \leq i \leq N$ y $0 \leq j < M$, el tiempo $t_{i,j}$ (en segundos) cuando el autobús i **llega a la** estación de rebase j se define así. Sea $t_{i,0} = T[i]$ para toda $0 \leq i < N$, y sea $t_{N,0} = Y$. Para toda j tal que $0 < j < M$:

- Sea el **tiempo previsto de llegada** (en segundos) del autobus i a la estación de rebase j, denotado por $e_{i,j}$, el momento en que el autobús i llegaría a la estación de rebase j si viajase a toda velocidad desde el momento en que llegó a la estación de rebase $j-1$. Es decir, sea
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ para toda $0 \leq i < N$, y
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- El autobús i llega a la estación de rebase j en el *máximo* de los tiempos previstos de llegada del autobús i y de todos los demás autobuses que llegaron a la estación $j-1$ antes que el autobús i. Formalmente, sea $t_{i,j}$ el máximo de $e_{i,j}$ y de todo $e_{k,j}$ para el que $0 \leq k \leq N$ y $t_{k,j-1} < t_{i,j-1}$.

Los organizadores de la IOI quieren programar el autobús sustituto (autobús N). Tu tarea es responder Q preguntas de los organizadores, las cuales serán de la siguiente forma: dado el tiempo Y (en segundos) cuando el autobús sustituto sale del aeropuerto, ¿a qué hora llegará al hotel?

Detalles de implementación

Debes implementar las siguientes funciones:

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

- L: la longitud de la carretera.
- N: el número de autobuses ordinarios.
- T: un arreglo de tamaño N que describe los tiempos en los que los autobuses ordinarios planean salir del aeropuerto.
- W: un arreglo de tamaño N que describe las velocidades máximas de los autobuses ordinarios.
- X: el tiempo que le toma al autobús sustituto viajar 1 kilómetro.
- M: el número de estaciones de rebase.
- S: un arreglo de tamaño M que describe las distancias de las estaciones de rebase desde el aeropuerto.
- Esta función se llamará exactamente una vez para cada caso de prueba, antes de cualquier llamada a `arrival_time`.

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

- Y: el tiempo en el que el autobús sustituto (autobús N) saldrá del aeropuerto.
- Esta función deberá retornar el tiempo en el que el autobús sustituto llegaría al hotel.
- Esta función se llamará 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 autobús 4 (que aún no ha sido programado), la siguiente tabla muestra los tiempos de llegada previstos y reales de los autobuses ordinarios a cada estación de rebase:

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 tiempos de llegada a la estación 0 son los tiempos en los que los autobuses están programados para salir del aeropuerto. Es decir, $t_{i,0} = T[i]$ para $0 \le i \le 3$.

Los tiempos previstos y reales de llegada a la estación de rebase 1 se calculan de la siguiente manera:

- Los tiempos previstos de llegada a la estación de rebase 1:
  - Autobús 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - Autobús 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - Autobús 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - Autobús 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Los tiempos de llegada a la estación de rebase 1:
  - Los autobuses 1 y 3 llegan a la estación 0 antes que el autobus 0, entonces $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - El autobús 3 llega a la estación 0 antes que el autobus 1, entonces $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Los autobuses 0, 1 y 3 llegan a la estación de rebase 0 antes que el autobús 2, entonces $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Ningún autobús llega a la estación 0 antes que el autobús 3, entonces $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

El autobús 4 toma 10 segundos para viajar 1 kilómetro y está programado para dejar el aeropuerto en el segundo 0. En este caso, la siguiente tabla muestra los tiempos de llegada de cada autobús. El único cambio en cuanto a los tiempos de llegada previstos y reales de los autobuses ordinarios 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

Vemos que el autobús 4 llega al hotel en el segundo 60. Por lo tanto, la función debe retornar 60.

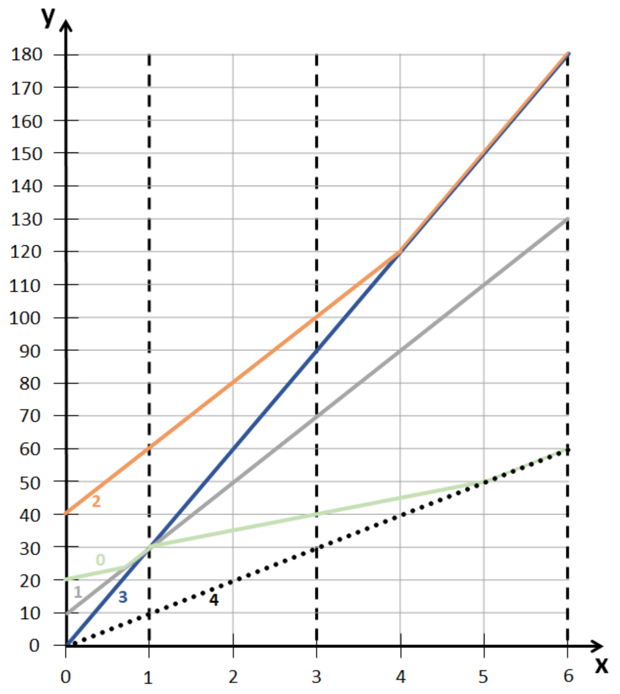
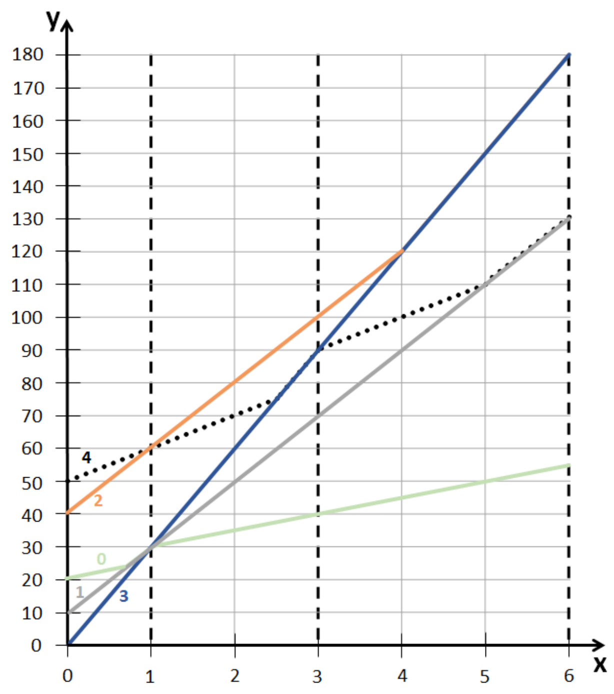
```
arrival_time(50)
```

El autobús 4 ahora está programado para salir del aeropuerto en el segundo 50. En este caso, no hay cambios en los tiempos de llegada de los autobuses ordinarios, comparado con la tabla inicial. Los tiempos 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 autobús 4 rebasa al autobús más lento 2 en la estación de rebase 1 pues llegan al mismo tiempo. Luego, el autobus 4 se amontona con el autobús 3 entre la estacion 1 y la estación 2, haciendo que el autobús 4 llegue a la estación 2 en el segundo 90 en lugar del segundo 80. Después de salir de la estación 2, el autobús 4 se amontona con el autobús 1 hasta llegar al hotel. El autobús 4 llega al hotel en el segundo 130. Por lo tanto, la función debe de regresar 130.

Podemos graficar el tiempo que tarda cada autobús en llegar a cada distancia desde el aeropuerto. El eje x de la gráfica representa la distancia desde el aeropuerto (en kilómetros) y el eje y de la gráfica representa el tiempo (en segundos). Las líneas punteadas verticales marcan las posiciones de las estaciones de rebase. Diferentes líneas sólidas (acompañadas de los índices de los autobuses) representan los cuatro autobuses ordinarios. La línea negra punteada representa el autobús sustituto.

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

Límites

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

Subtareas

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

Evaluador de ejemplo

El evaluador de ejemplo 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 de ejemplo imprime tus respuestas en el siguiente formato:

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