Adelantando buses

Hay un camino de un sentido y una sola vía desde el Aeropuerto de Budapest al Hotel Forrás. El camino es de L kilómetros de largo.

Durante la IOI 2023, $N+1$ buses de traslado atraviesan esta carretera. Los buses están enumerados de 0 a N. El bus i ($0 \le i < N$) está programado para dejar el aeropuerto en el segundo $T[i]$ del evento, y puede viajar 1 kilómetro en $W[i]$ segundos. El bus N es un bus de reserva que puede viajar 1 kilómetro en X segundos. Aún no se ha decidido el tiempo Y cuando este debe dejar el aeropuerto

En general, no está permitido adelantar a otros autos en la carretera, pero los buses 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 carretera. La estación de ordenamiento j ($0 \le j < M$) está ubicada a $S[j]$ kilómetros del aeropuerto a lo largo de la carretera. Las estaciones de ordenamiento están ordenadas de forma creciente según su distancia desde el aeropuerto, es decir, $S[j] < S[j+1]$ para todo $0 \le j \le M-2$. La primera estación de ordenamiento está ubicada en el aeropuerto y la última es el hotel, o sea $S[0] = 0$ y $S[M-1] = L$

Cada bus viaja a máxima velocidad a menos que éste se encuentre con un bus más lento viajando delante de él en la carretera, en ese caso, se amontonan y se ven forzados a viajar a la velocidad del bus más lento hasta alcanzar la siguiente estación de ordenamiento. Allí, los buses más rápidos adelantarán a los buses más lentos.

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

- Se define el **tiempo esperado de llegada** (en segundos) del bus i a la estación de ordenamiento j, denotado por $e_{i,j}$, como el tiempo en el cual el bus i debería llegar a la estación j si estuviese viajando a máxima velocidad desde el momento(tiempo) que llegó a la estación de ordenamiento $j-1$. Es decir
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ para cada $0 \le i < N$, y
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- El bus i llega a la estación de ordenamiento j al *máximo* de los tiempos esperados de llegada del bus i y de todos los otros buses que llegaron a la estación $j-1$ antes que el bus

i. Formalmente, sea $t_{i,j}$ el máximo de $e_{i,j}$ y de cada $e_{k,j}$ para el cual $0 \leq k \leq N$ y $t_{k,j-1} < t_{i,j-1}$.

Los organizadores de la IOI quieren planificar la salida del bus de reserva (bus N). Tu tarea es responder las Q preguntas de los organizadores, que son de la siguiente forma: dado el tiempo Y (en segundos) en el que se supone que el bus de reserva debe salir del aeropuerto, ¿a qué hora 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 carretera.
- N: el número de buses que no son de reserva.
- T: un arreglo de tamaño N describiendo los tiempos en que los buses que no son de reserva son programados para dejar el aeropuerto.
- W: un arreglo de tamaño N describiendo la máxima velocidad de los buses que no son de reserva.
- X: el tiempo que toma para el bus 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 por cada caso de prueba, antes de cualquier llamada a `arrival_time`.

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

- Y: el tiempo en que se supone que el bus (bus N) de reserva debe dejar el aeropuerto.
- Esta función debe retornar el tiempo en que el bus 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 bus 4 (dado que aún no ha sido planificado), la siguiente tabla muestra los tiempos de llegada esperados y actuales para los buses 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 tiempos de llegada a la estación 0 son los tiempos en los que los buses son programados para dejar el aeropuerto. Esto es, $t_{i,0} = T[i]$ para $0 \le i \le 3$.

Los tiempos esperados y reales de llegada a la estación 1 son calculados de la siguiente forma:

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

```
arrival_time(0)
```

El bus 4 toma 10 segundos en 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 para cada bus. El único cambio en cuanto a los tiempos de llegada esperados y reales de los buses 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 bus 4 llega al hotel en el segundo 60. Así que, la función debería retornar 60

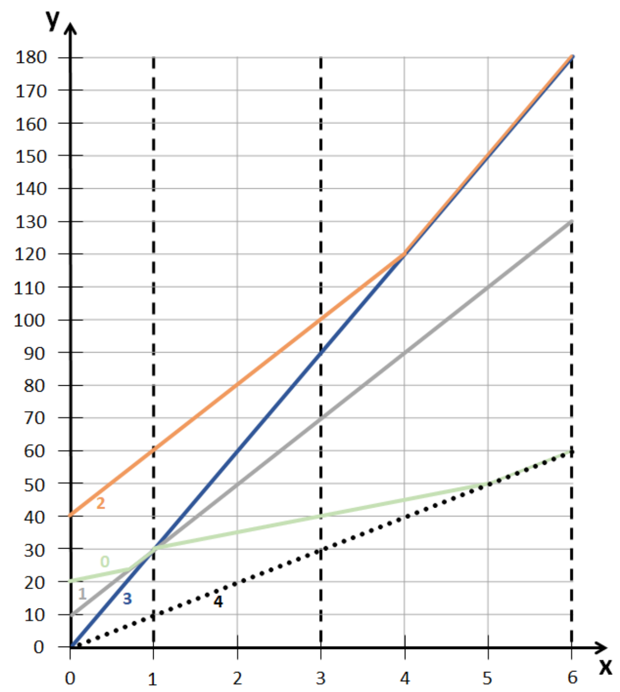
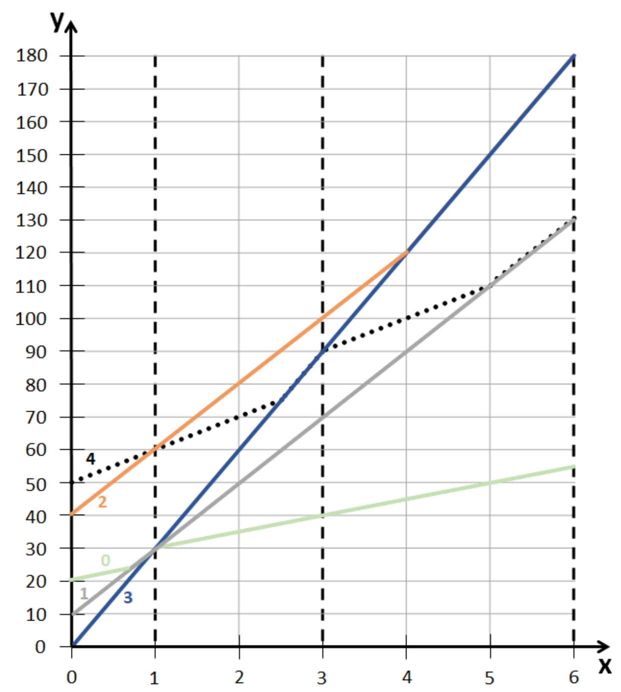
```
arrival_time(50)
```

El bus 4 ahora es programado para dejar el aeropuerto en el segundo 50. En este caso, no hay cambios en los tiempos de llegada para los buses que no son de reserva, comparado con la tabla inicial. Los tiempos de llegada son mostrados 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 bus 4 adelanta al bus 2; que está yendo más lento, en la estación de ordenamiento 1 debido a que llegan al mismo tiempo. Luego, el bus 4 se agrupa con el bus 3 entre las estaciones 1 y 2, haciendo que el bus 4 llegue a la estación 2 al segundo 90 en vez del segundo 80. Luego de dejar la estación 2, el bus 4 se agrupa con el bus 1 hasta que llegan al hotel. El bus 4 llega al hotel en el segundo 130. Así que la función debería devolver 130.

Podemos graficar el tiempo que toma para cada bus 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 discontinuas verticales marcan las posiciones de las estaciones de ordenamiento. Diferentes líneas enteras (acompañadas de los índices de buses) representan los cuatro buses que no son de reserva. La línea negra punteada representa el bus de reserva.

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

Limites

- $1 \leq L \leq 10^9$
- $1 \leq N \leq 1\,000$
- $0 \leq T[i] \leq 10^{18}$ (para todo i tal que $0 \leq i < N$)
- $1 \leq W[i] \leq 10^9$ (para todo 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 retornado de `arrival_time` para la pregunta k.