Ultrapassagem

Há uma via de mão única, com uma única faixa, do aeroporto de Budapeste para o Hotel Forrás. A via possui L quilômetros.

Durante o evento da IOI 2023, $N + 1$ ônibus de transporte atravessam essa via. Os ônibus são numerados de 0 a N. O ônibus i ($0 \le i < N$) está programado para sair do aeroporto no $T[i]$-ésimo segundo do evento, e pode percorrer 1 quilômetro em $W[i]$ segundos. O ônibus N é um ônibus reserva que pode percorrer 1 quilômetro em X segundos. O tempo Y no qual ele vai sair do aeroporto ainda não foi decidido.

Realizar ultrapassagens não é permitido na via em geral, mas os ônibus podem ultrapassar uns aos outros nas **estações de ordenação**. Existem M ($M > 1$) estações de ordenação, numeradas de 0 a $M - 1$, em posições diferentes da via. A estação de ordenação j ($0 \le j < M$) está localizada a $S[j]$ quilômetros do aeroporto ao longo da via. As estações de ordenação estão ordenadas pela distância ao aeroporto, isto é, $S[j] < S[j + 1]$ para cada $0 \le j \le M - 2$. A primeira estação de ordenação é o aeroporto e a última é o hotel, isto é, $S[0] = 0$ e $S[M - 1] = L$.

Cada ônibus viaja em sua velocidade máxima a menos que seja bloqueado por um ônibus mais lento em sua frente na via, neste caso eles andam juntos e são forçados a viajarem na velocidade do ônibus mais lento, até alcançarem a próxima estação de ordenação. Lá, os ônibus mais rápidos vão ultrapassar os ônibus mais lentos.

Formalmente, para cada i e j tais que $0 \le i \le N$ e $0 \le j < M$, o tempo $t_{i,j}$ (em segundos) quando o ônibus i **chega** à estação j é definido da seguinte maneira. Seja $t_{i,0} = T[i]$ para cada $0 \le i < N$ e seja $t_{N,0} = Y$. Para cada j tal que $0 < j < M$:

- Definimos o **tempo esperado de chegada** (em segundos) do ônibus i na estação j, denotado por $e_{i,j}$, como o tempo em que o ônibus i chegaria à estação j se viajasse sempre em sua velocidade máxima desde o momento em que chegou à estação $j - 1$. Isto é:
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ para todo $0 \le i < N$, e
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- O ônibus i chega à estação j no *máximo* entre os tempos esperados de chegada do ônibus i e de todos os outros ônibus que chegaram à estação $j - 1$ antes do ônibus i. Formalmente, seja $t_{i,j}$ o máximo entre $e_{i,j}$ e todos os $e_{k,j}$ para os quais $0 \le k \le N$ e $t_{k,j-1} < t_{i,j-1}$.

Os organizadores das IOI querem agendar a saída do ônibus reserva (o ônibus N). Sua tarefa é responder a Q questões dos organizadores, cada uma no seguinte formato: dado o tempo Y (em segundos) em que o ônibus reserva supostamente irá sair do aeroporto, em qual momento ele chegará ao hotel?

Detalhes de Implementação

Sua tarefa é implementar os seguintes procedimentos.

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

- L: o comprimento da via.
- N: o número de ônibus que não são reserva.
- T: um vetor de tamanho N descrevendo os tempos em que os ônibus que não são reserva estão programados para saírem do aeroporto.
- W: um vetor de tamanho N descrevendo as velocidades máximas dos ônibus que não são reserva.
- X: o tempo que o ônibus reserva demora para percorrer 1 km.
- M: o número de estações de ordenação.
- S: um vetor de tamanho M descrevendo as distâncias das estações ao aeroporto.
- Este procedimento será chamado exatamente uma vez para cada caso de teste, antes de qualquer chamada a `arrival_time`.

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

- Y: o tempo em que o ônibus reserva (ônibus N) supostamente irá sair do aeroporto.
- Este procedimento deve retornar o tempo no qual o ônibus reserva chegaria ao hotel.
- Este procedimento será chamado exatamente Q vezes.

Exemplos

Considere a seguinte sequência de chamadas:

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

Ignorando o ônibus 4 (que ainda não possui saída programada), a seguinte tabela mostra os tempos esperados e os tempos reais de chegada para todos os ônibus que não são reserva em cada uma das estações:

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

Os tempos de chegada à estação 0 são os tempos em que os ônibus estão programados para saírem do aeroporto. Isto é $t_{i,0} = T[i]$ para $0 \le i \le 3$.

Os tempos esperados e reais de chegada à estação 1 são calculados da seguinte maneira:

- Os tempos esperados de chegada à estação 1:
  - Ônibus 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - Ônibus 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - Ônibus 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - Ônibus 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Os tempos reais de chegada à estação 1:
  - Os ônibus 1 e 3 chegam à estação 0 antes do ônibus 0, portanto $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - O ônibus 3 chega à estação 0 antes do ônibus 1, portanto $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Os ônibus 0, 1 e 3 chegam à estação 0 antes do ônibus 2, portanto $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Nenhum ônibus chega à estação 0 antes do ônibus 3, portanto $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

O ônibus 4 leva 10 segundos para percorrer 1 km e agora está programado para sair no segundo 0. Neste caso, a tabela seguinte mostra os tempos de chegada de cada ônibus. A única alteração em relação aos tempos esperados e reais dos ônibus que não são reserva está sublinhada.

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 o ônibus 4 chega ao hotel no segundo 60. Portanto, o procedimento deve retornar 60.

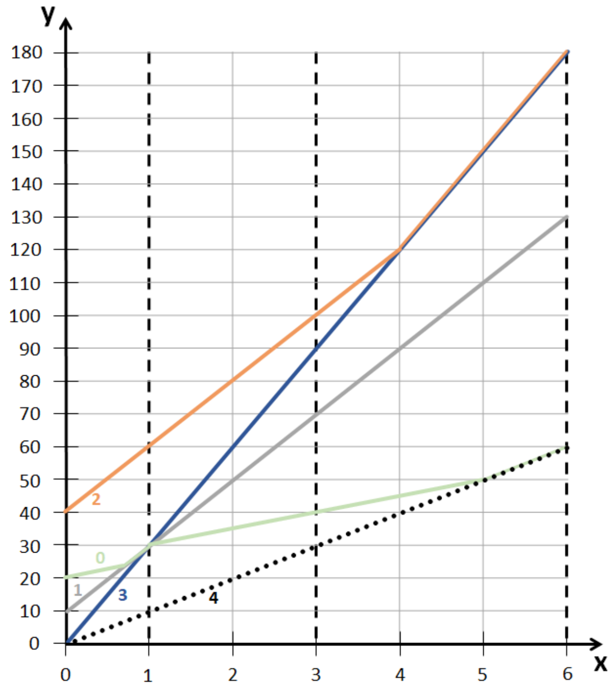
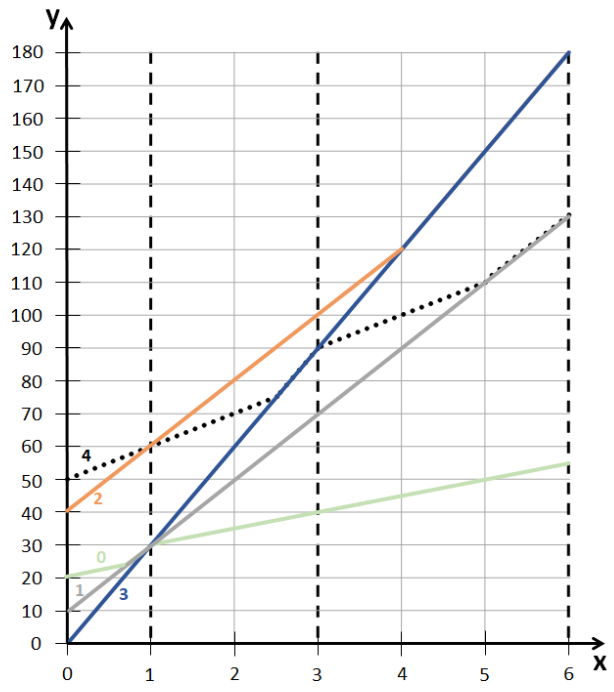
```
arrival_time(50)
```

O ônibus 4 está agora programado para sair do aeroporto no segundo 50. Neste caso, não existem mudaças nos tempos de chegada dos ônibus que não são reserva quando comparados com a tabela inicial. Os tempos de chegada são mostrados na seguinte tabela.

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

O ônibus 4 ultrapassa o ônibus 2, que é mais lento, na estação 1 porque chegam ao mesmo tempo. A seguir, o ônibus 4 fica bloqueado pelo ônibus 3 entre as estações 1 e 2, fazendo com que o ônibus 4 chegue à estação 2 no segundo 90 em vez do segundo 80. Depois de deixar a estação 2, o ônibus 4 fica bloqueado pelo ônibus 1 até chegarem ao hotel. O ônibus 4 chega ao hotel no segundo 130. Por isso, o procedimento deve retornar 130.

Podemos fazer um gráfico do tempo em que cada ônibus chega a cada distância a partir do aeroporto. O eixo x representa a distância ao aeroporto (em quilômetros) e o eixo y representa o tempo (em segundos). As linhas tracejadas verticais marcam as posições das estações. As diferentes linhas contínuas (acompanhadas dos índices dos ônibus) representam os 4 ônibus que não são reserva. A linha pontilhada representa o ônibus reserva.

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

Restrições

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

Subtarefas

1. (9 pontos) $N = 1, Q \leq 1\,000$
2. (10 pontos) $M = 2, Q \leq 1\,000$
3. (20 pontos) $N, M, Q \leq 100$
4. (26 pontos) $Q \leq 5\,000$
5. (35 pontos) Nenhuma restrição adicional

Corretor Exemplo

O corretor exemplo lê a entrada no seguinte formato:

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

O corretor exemplo imprime suas respostas no seguinte formato:

- linha $1+k$ ($0 \leq k < Q$): o valor retornado por `arrival_time` na pergunta k