Dépassement

L'aéroport de Budapest est relié à l'Hotel Forrás par une route à une seule voie et à sens unique de L kilomètres.

À l'occasion de l'IOI 2023, $N+1$ cars empruntent cette route. Ils sont numérotés de 0 à N inclus. Le car i ($0 \leq i < N$) doit partir de l'aéroport à la $T[i]$-ième seconde de l'évènement et peut parcourir 1 kilomètre en $W[i]$ secondes. Le car N est un car de réserve qui peut parcourir 1 kilomètre en X secondes. L'heure Y à laquelle ce car quittera l'aéroport n'est pas encore décidée.

Les dépassements n'étant pas autorisés sur cette route, les cars peuvent uniquement se dépasser dans des **élargissements**. Il y a M ($M > 1$) élargissements, numérotés de 0 à $M-1$, positionnés le long de la route. L'élargissement j ($0 \leq j < M$) est situé à $S[j]$ kilomètres de l'aéroport, le long de la route. Les élargissements sont triés par ordre croissant de distance de l'aéroport, c'est-à-dire $S[j] < S[j+1]$ pour chaque $0 \leq j \leq M-2$. Le premier élargissement est situé à l'aéroport, le dernier à l'hôtel, donc $S[0] = 0$ et $S[M-1] = L$.

Chaque car circule à sa vitesse maximale, sauf s'il rattrape un car plus lent, auquel cas il est obligé de rouler à la vitesse du car plus lent jusqu'au prochain élargissement. À cet élargissement, le car plus rapide dépassera les cars plus lents que lui.

Formellement : pour chaque i et j tels que $0 \leq i \leq N$ et $0 \leq j < M$, l'heure $t_{i,j}$ (en secondes) à laquelle un car i **arrive** à l'élargissement j est définie comme suit. Soit $t_{i,0} = T[i]$ pour tout $0 \leq i < N$ et soit $t_{N,0} = Y$. Pour tout j tel que $0 < j < M$:

- L'**heure attendue d'arrivée** (en secondes) du car i à l'élargissement j est notée $e_{i,j}$ et est définie comme l'heure à laquelle le car i arriverait à l'élargissement j s'il circulait à sa vitesse maximale depuis son passage à l'élargissement $j-1$. Donc :
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ pour chaque $0 \leq i < N$, et
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- Le car i arrive à l'élargissement j à l'heure *maximale* parmi les heures attendues d'arrivée du car i et de tous les autres cars arrivés à l'élargissement $j-1$ avant le car i. Formellement, $t_{i,j}$ est le maximum de $e_{i,j}$ et de tous les $e_{k,j}$ pour lesquels $0 \leq k \leq N$ et $t_{k,j-1} < t_{i,j-1}$.

Les organisateurs de l'IOI veulent fixer l'horaire du car de réserve (car N). Votre tâche est de répondre à Q questions ayant la forme suivante : étant donnée l'heure Y (en secondes) à laquelle le car de réserve partira de l'aéroport, à quelle heure arrivera-t-il à l'hôtel ?

Détails d'implémentation

Votre tâche est d'implémenter les fonctions suivantes.

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

- L: la longueur de la route.
- N: le nombre de cars planifiés ("un car planifié" est un car qui n'est pas le car de réserve).
- T: un tableau de taille N représentant les moments auxquels les cars planifiés doivent partir de l'aéroport.
- W: un tableau de taille N représentant les vitesses maximales des cars planifiés.
- X: le temps que prend le car de réserve pour parcourir 1 kilomètre.
- M: le nombre d'élargissements.
- S: un tableau de longueur M représentant les distances depuis l'aéroport jusqu'à chaque élargissement.
- Cette fonction est appelée exactement une fois pour chaque test, avant tout appel à `arrival_time`.

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

- Y: l'heure à laquelle le car de réserve (car N) doit quitter l'aéroport.
- Cette fonction doit renvoyer l'heure à laquelle le car de réserve arriverait à l'hotel.
- Cette fonction est appelée exactement Q fois.

Exemple

Considérons la séquence d'appels suivants:

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

Si on ignore le car 4 (dont l'heure de départ n'a pas encore été décidée), le tableau suivant montre les heures attendues d'arrivée et les heures réelles d'arrivée à chaque élargissement pour tous les cars (sauf pour le car de réserve) :

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

Les heures d'arrivée à l'élargissement 0 sont les heures auxquelles les cars doivent quitter l'aéroport. C'est-à-dire, $t_{i,0} = T[i]$ for $0 \leq i \leq 3$.

Les heures attendues d'arrivée et les heures réelles d'arrivée à l'élargissement 1 sont calculées ainsi :

- Les heures attendues d'arrivée à l'élargissement 1 :
    - Car 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
    - Car 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
    - Car 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
    - Car 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Les heures réelles d'arrivée à l'élargissement 1 :
    - Les cars 1 et 3 arrivent à l'élargissement 0 avant le car 0, donc $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
    - Le car 3 arrive à l'élargissement 0 avant le car 1, donc $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
    - Les cars 0, 1 et 3 arrivent à l'élargissement 0 avant le car 2, donc $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
    - Aucun car n'arrive à l'élargissement 0 avant le car 3, donc $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Le car 4 prend 10 secondes pour parcourir 1 kilomètre et doit désormais quitter l'aéroport en la seconde 0. Dans ce cas, le tableau suivant montre les heures d'arrivée de chaque car. Le seul changement concernant les heures d'arrivée (attendues et réelles) des cars autres que celui de réserve est souligné.

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

On remarque que le car 4 arrive à l'hotel à la 60-ième seconde. Donc, la fonction doit renvoyer 60.

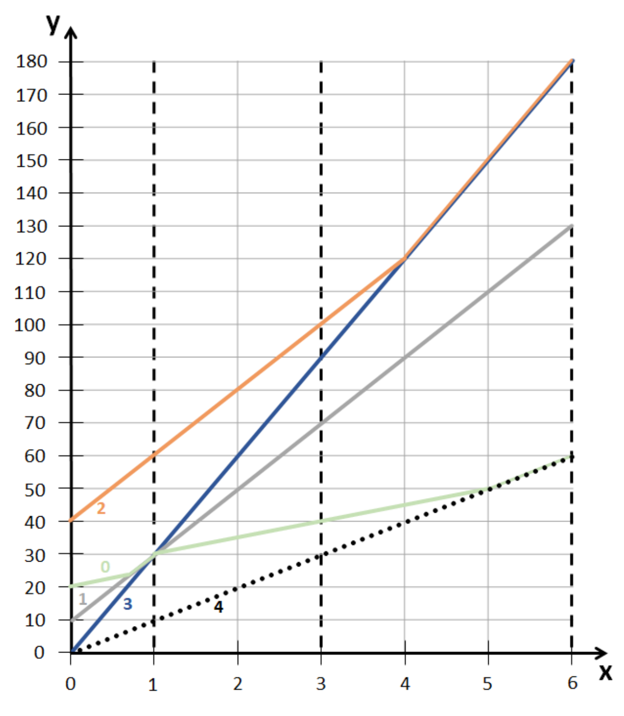
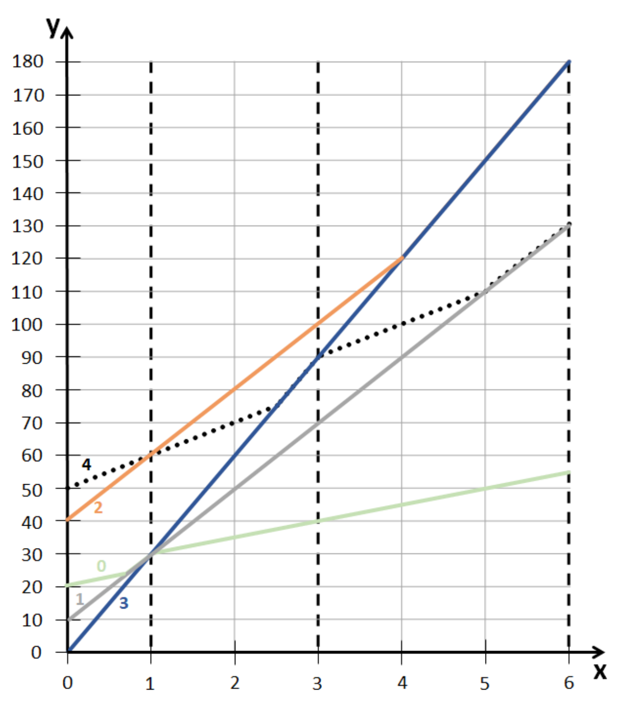
```
arrival_time(50)
```

Le car 4 doit désormais quitter l'aéroport à la 50-ième seconde. Dans ce cas, il n'y a pas de changements dans les heures d'arrivée des cars autres que celui de réserve, comparé au tableau initial. Les heures d'arrivée sont montrées dans le tableau suivant.

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

Le car 4 dépasse le car 2 (qui est plus lent) à l'élargissement 1, où ils arrivent à la même heure. Ensuite, le car 4 se retrouve bloqué derrière le car 3 entre les élargissements 1 et 2, ce qui fait que le car 4 arrive à l'élargissement 2 à la 90-ième seconde au lieu de la 80-ième. Après l'élargissement 2, le car 4 se retrouve bloqué derrière le car 1 jusqu'à ce qu'ils arrivent à l'hôtel. Le car 4 arrive à l'hôtel à la 130-ième seconde. Donc, la fonction doit renvoyer 130.

On peut faire un graphique du temps qu'il faut à chaque car pour arriver à chaque distance de l'aéroport. L'axe x du graphique représente la distance depuis l'aéroport (en kilomètres) et l'axe y représente le temps (en secondes). Les lignes pointillées (tirets) verticales marquent la position des élargissements. Les quatres lignes pleines (accompagnés du numéro de car) représentent les cars, sauf celui de réserve. La ligne noire pointillée (points) représente le car de réserve.

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

Contraintes

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

Sous-tâches

1. (9 points) $N = 1, Q \le 1\,000$
2. (10 points) $M = 2, Q \le 1\,000$
3. (20 points) $N, M, Q \le 100$
4. (26 points) $Q \le 5\,000$
5. (35 points) Aucune contrainte supplémentaire.

Évaluateur d'exemple (grader)

L'évaluateur d'exemple (grader) lit l'entrée au format suivant :

- ligne 1: $L\ N\ X\ M\ Q$
- ligne 2: $T[0]\ T[1]\ \ldots\ T[N-1]$
- ligne 3: $W[0]\ W[1]\ \ldots\ W[N-1]$
- ligne 4: $S[0]\ S[1]\ \ldots\ S[M-1]$
- ligne $5+k$ ($0 \le k < Q$): Y pour la question k

L'évaluateur d'exemple (grader) affiche vos réponses au format suivant :

- ligne $1+k$ ($0 \le k < Q$): la valeur renvoyée par `arrival_time` pour la question k