



Обгін

На дорозі з аеропорту Будапешту до готелю Форрас є однопалосна, одностороння дорога довжиною L кілометрів.

Протягом IOI 2023 через цю дорогу проїжджають $N + 1$ трансферних автобусів. Автобуси пронумеровані від 0 до N . Автобус i ($0 \leq i < N$) запланований до відправлення з аеропорту у $T[i]$ -у секунду і може проїхати 1 кілометр за $W[i]$ секунд. Автобус N є резервним автобусом, який може проїхати 1 кілометр за X секунд. Час Y , коли він відправиться з аеропорту, ще не визначений.

Обгін на дорозі загалом заборонений, але автобуси можуть обганяти один одного на **станціях сортування**. Існує M ($M > 1$) станцій сортування, пронумерованих від 0 до $M - 1$, розташованих на різних позиціях на дорозі. Станція сортування j ($0 \leq j < M$) знаходиться на відстані $S[j]$ кілометрів від аеропорту вздовж дороги. Станції сортування впорядковані за зростанням відстані від аеропорту, тобто $S[j] < S[j + 1]$ для кожного $0 \leq j \leq M - 2$. Перша станція сортування - це аеропорт, а остання - готель, тобто $S[0] = 0$ і $S[M - 1] = L$.

Кожен автобус рухається з максимальною швидкістю, поки не наздожене повільніший автобус, що їде попереду на дорозі. В такому випадку вони утворюють групу та змушені рухатися зі швидкістю найповільнішого автобуса, поки не досягнуть наступної станції сортування. Лише там швидші автобуси випередять повільніші.

Формально, для кожних i та j , де $0 \leq i \leq N$ та $0 \leq j < M$, час $t_{i,j}$ (у секундах), коли автобус i **прибуває до** станції сортування j , визначається наступним чином. Для кожного $i < N$ встановлюємо $t_{i,0} = T[i]$, а для N встановлюємо $t_{N,0} = Y$. Для кожного j такого, що $0 < j < M$:

- Визначимо **очікуваний час прибуття** (в секундах) автобуса i до станції сортування j (позначимо як $e_{i,j}$) як час, коли автобус i прибуде до станції сортування j , якби він рухався з повною швидкістю з часу прибуття на станцію сортування $j - 1$. Іншими словами, встановлюємо
 - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j - 1])$ для кожного $0 \leq i < N$, і
 - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j - 1])$.
- Автобус i прибуває до станції сортування j в **максимальний** із очікуваних часів прибуття автобуса i та всіх інших автобусів, які прибули на станцію $j - 1$ раніше, ніж автобус i . Формально, встановлюємо $t_{i,j}$ як максимальне значення $e_{i,j}$ та всіх $e_{k,j}$, для яких $0 \leq k \leq N$ і $t_{k,j-1} < t_{i,j-1}$.

Організатори IOI хочуть спланувати резервний автобус (автобус N). Ваше завдання - відповісти на Q запитань організаторів, які мають наступний вигляд: якщо дано час Y (у секундах), коли резервний автобус повинен виїхати з аеропорту, в який час він прибуде до готелю?

Деталі імплементації

Вам потрібно імплементувати наступні функції:

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

- L : довжина дороги.
- N : кількість нерезервних автобусів.
- T : масив довжиною N , що описує часи, коли нерезервні автобуси мають виїхати з аеропорту.
- W : масив довжиною N , що описує максимальні швидкості нерезервних автобусів.
- X : час, який потрібен резервному автобусу подолати 1 кілометр.
- M : кількість станцій сортування.
- S : масив довжиною M , що описує відстані станцій сортування від аеропорту.
- Цю функцію викликають рівно один раз для кожного тестового випадку, перед будь-якими викликами `arrival_time`.

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

- Y : час, коли резервний автобус (автобус N) має виїхати з аеропорту.
- Ця функція повинна повернути час, коли резервний автобус прибуде до готелю.
- Цю функцію викликають рівно Q разів.

Приклад

Розглянемо наступну послідовність викликів:

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

Ігноруючи автобус 4 (який ще не був запланований), наступна таблиця показує очікувані та фактичні часи прибуття для нерезервних автобусів на кожну станцію сортування:

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

Часи прибуття на станцію 0 - це часи, коли автобуси заплановані виїхати з аеропорту. Іншими словами, $t_{i,0} = T[i]$ для $0 \leq i \leq 3$.

Очікувані та фактичні часи прибуття на станцію 1 обчислюються таким чином:

- Очікувані часи прибуття на станцію 1:
 - Автобус 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
 - Автобус 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
 - Автобус 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
 - Автобус 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Часи прибуття на станцію 1:
 - Автобуси 1 та 3 прибувають на станцію 0 раніше, ніж автобус 0, тому $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
 - Автобус 3 прибуває на станцію 0 раніше, ніж автобус 1, тому $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
 - Автобуси 0, 1 та 3 прибувають на станцію 0 раніше, ніж автобус 2, тому $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
 - Ніякий автобус не прибуває на станцію 0 раніше, ніж автобус 3, тому $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Автобусу 4 потрібно 10 секунд, щоб подолати 1 кілометр, і зараз він запланований до виїзду з аеропорту у 0-ву секунду. У цьому випадку наведена таблиця показує часи прибуття для кожного автобуса. Єдине змінене значення щодо очікуваних та фактичних часів прибуття нерезервних автобусів **підкреслено**.

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	<u>60</u>
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

Ми бачимо, що автобус 4 прибуває до готелю на 60-ту секунду. Отже, процедура повинна повернути 60.

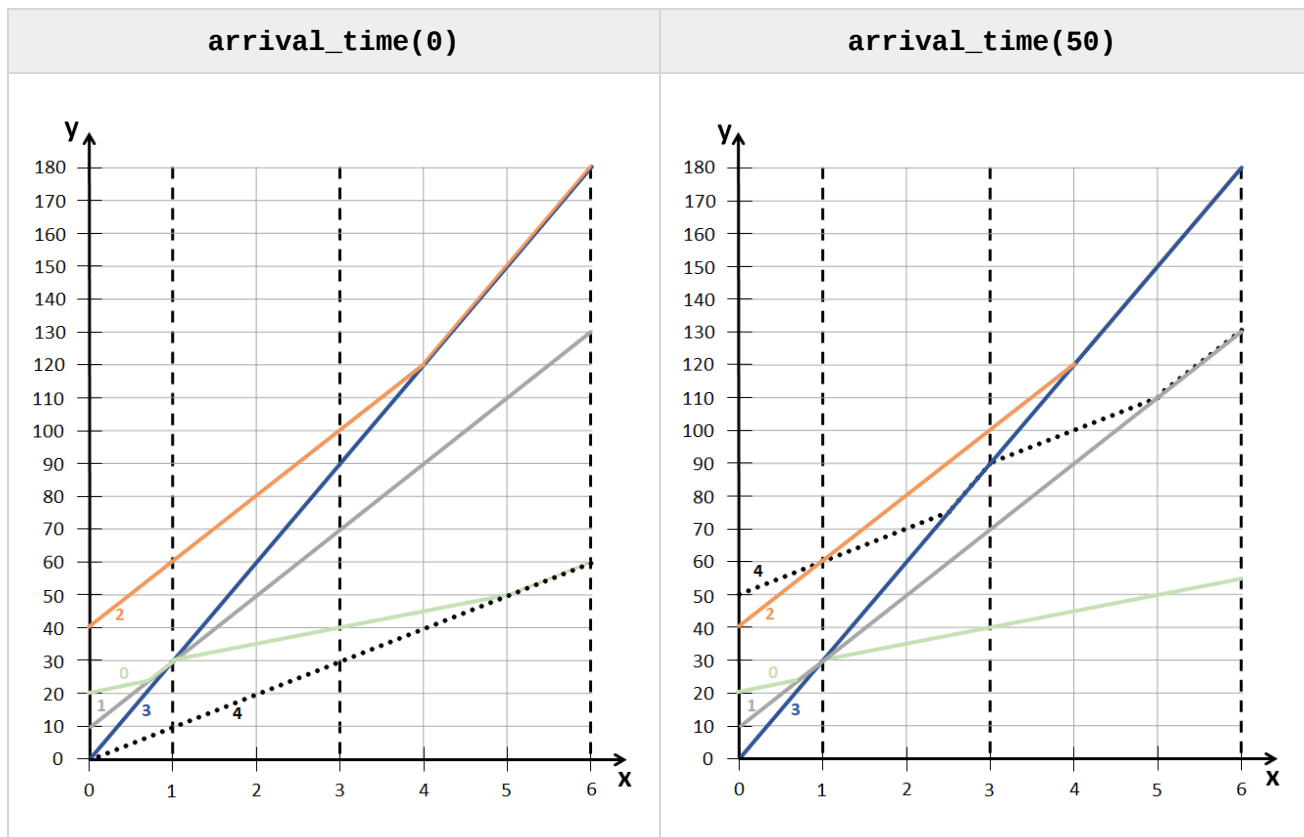
```
arrival_time(50)
```

Автобусу 4 зараз заплановано виїхати з аеропорту на 50-ту секунду. У цьому випадку немає змін в часах прибуття для нерезервних автобусів порівняно зі стартовою таблицею. Часи прибуття показані у наступній таблиці.

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

Автобус 4 випереджає повільний автобус 2 на станції сортування 1, оскільки вони прибувають одночасно. Далі, автобус 4 утворює групу з автобусом 3 між станціями 1 та 2, через що автобус 4 прибуває на станцію 2 на 90-тій секунді замість 80-тої. Після виходу зі станції 2, автобус 4 утворює групу з автобусом 1 до їхнього прибуття до готелю. Автобус 4 прибуває до готелю на 130-тій секунді. Отже, функція повинна повернути 130.

Ми можемо побудувати графік часу, необхідного для прибуття кожного автобуса на кожну відстань від аеропорту. Вісь x графіку представляє відстань від аеропорту (в кілометрах), а вісь y графіку представляє час (в секундах). Вертикальні пунктирні лінії позначають положення станцій сортування. Різні суцільні лінії (разом із позначеннями індексів автобусів) представляють чотири нерезервні автобуси. Пунктирна чорна лінія представляє резервний автобус.



Обмеження

- $1 \leq L \leq 10^9$
- $1 \leq N \leq 1\,000$
- $0 \leq T[i] \leq 10^{18}$ (для кожного i такого, що $0 \leq i < N$)
- $1 \leq W[i] \leq 10^9$ (для кожного i такого, що $0 \leq i < N$)
- $1 \leq X \leq 10^9$
- $2 \leq M \leq 1\,000$
- $0 = S[0] < S[1] < \dots < S[M-1] = L$
- $1 \leq Q \leq 10^6$
- $0 \leq Y \leq 10^{18}$

Підзадачі

1. (9 балів) $N = 1, Q \leq 1\,000$
2. (10 балів) $M = 2, Q \leq 1\,000$
3. (20 балів) $N, M, Q \leq 100$
4. (26 балів) $Q \leq 5\,000$
5. (35 балів) Без додаткових обмежень.

Приклад градера

Градер зчитує вхідні дані наступному форматі:

- Рядок 1: $L \ N \ X \ M \ Q$
- Рядок 2: $T[0] \ T[1] \ \dots \ T[N - 1]$
- Рядок 3: $W[0] \ W[1] \ \dots \ W[N - 1]$
- Рядок 4: $S[0] \ S[1] \ \dots \ S[M - 1]$
- Рядок $5 + k$ ($0 \leq k < Q$): Y для питання k

Градер виводить дані у наступному форматі:

- Рядок $1 + k$ ($0 \leq k < Q$): значення, повернене функцією `arrival_time` для питання k