Overtaking (Προσπέρασμα)

Από το αεροδρόμιο της Βουδαπέστης μέχρι το Hotel Forrás υπάρχει δρόμος μίας λωρίδας και μονής κατεύθυνσης. Ο δρόμος έχει μήκος L χιλιόμετρα.

Κατά τη διάρκεια της IOI 2023, $N + 1$ λεωφορεία διασχίζουν αυτόν τον δρόμο. Τα λεωφορεία αριθμούνται από το 0 έως το N. Το λεωφορείο i ($0 \leq i < N$) είναι προγραμματισμένο να αναχωρήσει από το αεροδρόμιο στο $T[i]$-οστό δευτερόλεπτο και μπορεί να διανύσει 1 χιλιόμετρο σε $W[i]$ δευτερόλεπτα. Το λεωφορείο N είναι ένα εφεδρικό λεωφορείο που μπορεί να διανύσει 1 χιλιόμετρο σε X δευτερόλεπτα. Η ώρα Y κατά την οποία θα αναχωρήσει από το αεροδρόμιο δεν έχει ακόμη αποφασιστεί.

Τα προσπεράσματα δεν επιτρέπονται γενικά στον δρόμο, αλλά τα λεωφορεία επιτρέπεται να προσπερνούν το ένα το άλλο στους **σταθμούς διαλογής**. Υπάρχουν M ($M > 1$) σταθμοί διαλογής, αριθμημένοι από 0 έως $M - 1$, σε διαφορετικά σημεία του δρόμου. Ο σταθμός διαλογής j ($0 \leq j < M$) βρίσκεται $S[j]$ χιλιόμετρα από το αεροδρόμιο κατά μήκος του δρόμου. Οι σταθμοί διαλογής δίνονται ταξινομημένοι σε αύξουσα απόσταση από το αεροδρόμιο, δηλαδή $S[j] < S[j + 1]$ για κάθε $0 \leq j \leq M - 2$. Ο πρώτος σταθμός διαλογής είναι το αεροδρόμιο και ο τελευταίος το ξενοδοχείο, δηλαδή $S[0] = 0$ και $S[M - 1] = L$.

Κάθε λεωφορείο ταξιδεύει με τη μέγιστη ταχύτητα, εκτός αν προλάβει ένα πιο αργό λεωφορείο που ταξιδεύει μπροστά του στον δρόμο, οπότε τα λεωφορεία αναγκάζονται να ταξιδεύουν με την ταχύτητα του πιο αργού λεωφορείου, μέχρι να φτάσουν στον επόμενο σταθμό διαλογής. Εκεί, τα ταχύτερα λεωφορεία θα προσπεράσουν τα πιο αργά λεωφορεία.

Τυπικά, για κάθε i και j τέτοια ώστε $0 \leq i \leq N$ και $0 \leq j < M$, ο χρόνος $t_{i,j}$ (σε δευτερόλεπτα) όταν το λεωφορείο i **φτάνει** στον σταθμό διαλογής j ορίζεται ως εξής. Έστω $t_{i,0} = T[i]$ για κάθε $0 \leq i < N$, και έστω $t_{N,0} = Y$. Για κάθε j τέτοιο ώστε $0 < j < M$:

- Ορίστε τον **αναμενόμενο χρόνο άφιξης** (σε δευτερόλεπτα) του λεωφορείου i στον σταθμό διαλογής j, που συμβολίζεται με $e_{i,j}$, ως τον χρόνο που το λεωφορείο i θα έφτανε στον σταθμό διαλογής j εάν ταξίδευε με πλήρη ταχύτητα από τη στιγμή που έφτασε στον σταθμό διαλογής $j - 1$. Δηλαδή, έστω
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j - 1])$ για κάθε $0 \leq i < N$, και
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j - 1])$.
- Το λεωφορείο i φτάνει στον σταθμό διαλογής j στο *μέγιστο* των αναμενόμενων χρόνων άφιξης του λεωφορείου i και κάθε άλλου λεωφορείου που έφτασε στον σταθμό $j - 1$

νωρίτερα από το λεωφορείο i. Τυπικά, έστω $t_{i,j}$ το μέγιστο του $e_{i,j}$ και κάθε $e_{k,j}$ για το οποίο $0 \le k \le N$ και $t_{k,j-1} < t_{i,j-1}$.

Οι διοργανωτές της IOI θέλουν να προγραμματίσουν την ώρα του εφεδρικού λεωφορείου (λεωφορείο N). Το καθήκον σας είναι να απαντήσετε σε ερωτήσεις Q των διοργανωτών, οι οποίες έχουν την ακόλουθη μορφή: αν το εφεδρικό λεωφορείο φύγει την ώρα Y (σε δευτερόλεπτα) από το αεροδρόμιο, ποια ώρα θα φτάσει στο ξενοδοχείο;

Λεπτομέρειες Υλοποίησης

Ο στόχος σας είναι να υλοποιήσετε τις ακόλουθες συναρτήσεις.

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

- L: το μήκος του δρόμου.
- N: το πλήθος των μη-εφεδρικών λεωφορείων.
- T: ένας πίνακας μήκους N που περιγράφει τις ώρες κατά τις οποίες τα μη-εφεδρικά λεωφορεία έχουν προγραμματιστεί να αναχωρήσουν από το αεροδρόμιο.
- W: ένας πίνακας μήκους N που περιγράφει τις μέγιστες ταχύτητες των μη-εφεδρικών λεωφορείων.
- X: ο χρόνος που χρειάζεται το εφεδρικό λεωφορείο για να διανύσει 1 χιλιόμετρο.
- M: το πλήθος των σταθμών διαλογής.
- S: ένας πίνακας μήκους M που περιγράφει τις αποστάσεις των σταθμών διαλογής από το αεροδρόμιο.
- Αυτή η συνάρτηση καλείται ακριβώς μία φορά για κάθε test case, πριν από οποιαδήποτε κλήση της `arrival_time`.

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

- Y: η ώρα κατά την οποία το εφεδρικό λεωφορείο (λεωφορείο N) πρέπει να αναχωρήσει από το αεροδρόμιο.
- Αυτή η συνάρτηση πρέπει να επιστρέφει την ώρα κατά την οποία το εφεδρικό λεωφορείο θα φτάσει στο ξενοδοχείο.
- Αυτή η συνάρτηση καλείται ακριβώς Q φορές.

Παράδειγμα

Ας υποθέσουμε την ακόλουθη σειρά κλήσεων:

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

Αγνοώντας το λεωφορείο 4 (το οποίο δεν έχει ακόμη προγραμματιστεί), ο ακόλουθος πίνακας δείχνει τους αναμενόμενους και τους πραγματικούς χρόνους άφιξης των μη-εφεδρικών λεωφορείων σε κάθε σταθμό διαλογής:

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

Οι ώρες άφιξης στον σταθμό 0 είναι οι ώρες κατά τις οποίες τα λεωφορεία είναι προγραμματισμένα να αναχωρήσουν από το αεροδρόμιο. Δηλαδή, $t_{i,0} = T[i]$ για $0 \le i \le 3$.

Οι αναμενόμενοι και οι πραγματικοί χρόνοι των αφίξεων στον σταθμό διαλογής 1 υπολογίζονται ως εξής:

- Οι αναμενόμενοι χρόνοι των αφίξεων στο σταθμό 1:
  - Λεωφορείο 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - Λεωφορείο 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - Λεωφορείο 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - Λεωφορείο 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Οι χρόνοι των αφίξεων στον σταθμό 1:
  - Τα λεωφορεία 1 και 3 φτάνουν στον σταθμό 0 νωρίτερα από το λεωφορείο 0, οπότε $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - Το λεωφορείο 3 φτάνει στον σταθμό 0 νωρίτερα από το λεωφορείο 1, οπότε $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Τα λεωφορεία 0, 1 και 3 φτάνουν στον σταθμό 0 νωρίτερα από το λεωφορείο 2, οπότε $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Κανένα λεωφορείο δεν φτάνει στον σταθμό 0 νωρίτερα από το λεωφορείο 3, οπότε $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Το λεωφορείο 4 χρειάζεται 10 δευτερόλεπτα για να διανύσει 1 χιλιόμετρο και είναι προγραμματισμένο να αναχωρήσει από το αεροδρόμιο στο δευτερόλεπτο 0. Στην περίπτωση αυτή, ο παρακάτω πίνακας δείχνει τους χρόνους άφιξης για κάθε λεωφορείο. Η μόνη αλλαγή όσον αφορά τους αναμενόμενους και πραγματικούς χρόνους άφιξης των μη-εφεδρικών λεωφορείων είναι υπογραμμισμένη.

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

Βλέπουμε ότι το λεωφορείο 4 φτάνει στο ξενοδοχείο το 60-οστό δευτερόλεπτο. Συνεπώς, η συνάρτηση θα επιστρέψει 60.

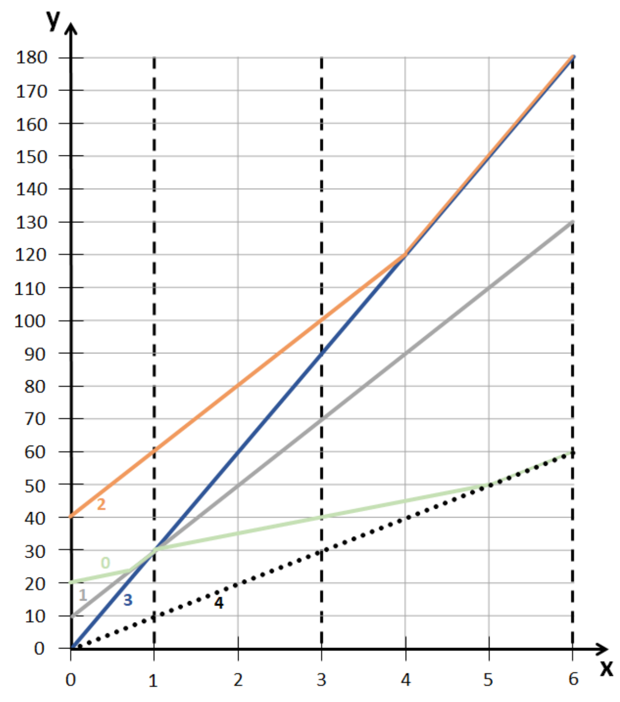
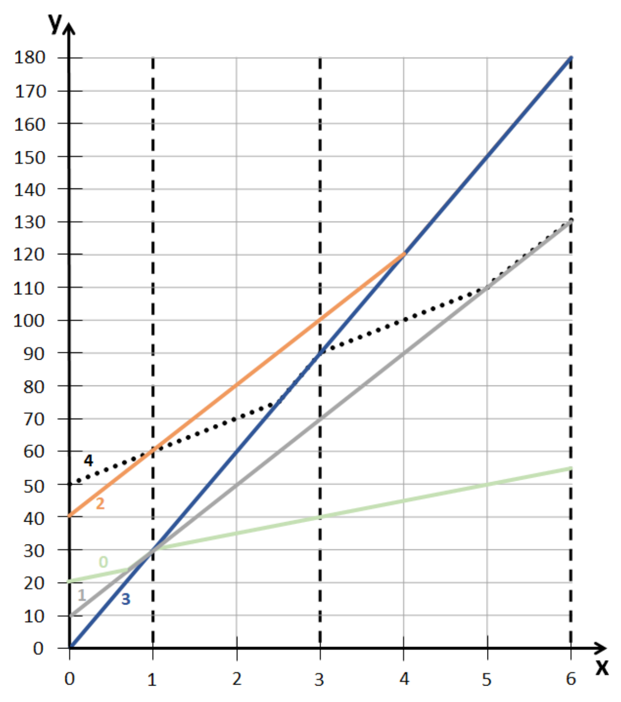
```
arrival_time(50)
```

Το λεωφορείο 4 είναι τώρα προγραμματισμένο να αναχωρήσει από το αεροδρόμιο στο 50-οστό δευτερόλεπτο. Σε αυτή την περίπτωση, δεν υπάρχουν αλλαγές στους χρόνους άφιξης των μη-εφεδρικών λεωφορείων σε σύγκριση με τον αρχικό πίνακα. Οι χρόνοι αφίξεων παρουσιάζονται στον ακόλουθο πίνακα.

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

Το λεωφορείο 4 προσπερνά το πιο αργό λεωφορείο 2 στον σταθμό διαλογής 1 καθώς φτάνουν ταυτόχρονα. Στη συνέχεια, το λεωφορείο 4 προχωρά μαζί με το λεωφορείο 3 μεταξύ του σταθμού 1 και του σταθμού 2, με αποτέλεσμα το λεωφορείο 4 να φτάνει στο σταθμό 2 στο 90-οστό δευτερόλεπτο αντί για το 80-οστό. Μετά την αναχώρηση από τον σταθμό 2, το λεωφορείο 4 προχωρά μαζί με το λεωφορείο 1 μέχρι να φτάσουν στο ξενοδοχείο. Το λεωφορείο 4 φτάνει στο ξενοδοχείο στο 130-οστό δευτερόλεπτο. Συνεπώς, η συνάρτηση θα πρέπει να επιστρέψει 130.

Μπορούμε να σχεδιάσουμε τον χρόνο που χρειάζεται κάθε λεωφορείο για να φτάσει σε κάθε απόσταση από το αεροδρόμιο. Ο άξονας x του διαγράμματος αντιπροσωπεύει την απόσταση από το αεροδρόμιο (σε χιλιόμετρα) και ο άξονας y του διαγράμματος αντιπροσωπεύει τον χρόνο (σε δευτερόλεπτα). Οι κατακόρυφες διακεκομμένες γραμμές απεικονίζουν τις θέσεις των σταθμών διαλογής. Οι συμπαγείς γραμμές (μαζί με τους δείκτες των λεωφορείων) αντιπροσωπεύουν τα τέσσερα μη-εφεδρικά λεωφορεία. Η διακεκομμένη μαύρη γραμμή αντιπροσωπεύει το εφεδρικό λεωφορείο.

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

Περιορισμοί

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

Subtasks (Υποπροβλήματα)

1. (9 βαθμοί) $N = 1, Q \leq 1\,000$
2. (10 βαθμοί) $M = 2, Q \leq 1\,000$
3. (20 βαθμοί) $N, M, Q \leq 100$
4. (26 βαθμοί) $Q \leq 5\,000$
5. (35 βαθμοί) Χωρίς πρόσθετους περιορισμούς.

Ενδεικτικός Grader

Ο ενδεικτικός grader διαβάζει την είσοδο με την ακόλουθη μορφή:

- γραμμή 1: $L \ N \ X \ M \ Q$
- γραμμή 2: $T[0] \ T[1] \ \ldots \ T[N-1]$
- γραμμή 3: $W[0] \ W[1] \ \ldots \ W[N-1]$
- γραμμή 4: $S[0] \ S[1] \ \ldots \ S[M-1]$
- γραμμή $5+k$ ($0 \leq k < Q$): Y για την ερώτηση k

Ο ενδεικτικός grader εμφανίζει τις απαντήσεις σας με την ακόλουθη μορφή:

- γραμμή $1+k$ ($0 \leq k < Q$): η επιστρεφόμενη τιμή της `arrival_time` για την ερώτηση k