Wyprzedzanie

Z lotniska w Budapeszcie do Hotelu Forrás prowadzi prosta jednokierunkowa droga. Ma ona L kilometrów długości.

Podczas IOI 2023, tą drogą przejedzie $N+1$ busów. Busy są ponumerowane od 0 do N. Bus numer i ($0 \leq i < N$) planowo opuści lotnisko po $T[i]$ sekundach od rozpoczęcia IOI 2023 i jest w stanie przejechać 1 kilometr w $W[i]$ sekund. Bus N jest busem **rezerwowym** i jest w stanie przejechać 1 kilometr w X sekund. Czas wyjazdu tego busa z lotniska, który oznaczamy przez Y, nie jest jeszcze ustalony. Busy o numerach od 0 do $N-1$ włącznie nazywamy **zaplanowanymi**.

Na rozważanej drodze nie jest dozwolone wyprzedzanie, jednak busy mogą się wyprzedzać na **stacjach**. Na drodze jest M stacji ($M > 1$), ponumerowanych od 0 do $M-1$. Stacja j ($0 \leq j < M$) znajduje się $S[j]$ kilometrów od lotniska. Stacje są posortowane rosnąco względem odległości od lotniska, czyli $S[j] < S[j+1]$ dla każdego $0 \leq j \leq M-2$. Pierwsza stacja znajduje się na lotnisku, a ostatnia przy hotelu, czyli $S[0] = 0$ oraz $S[M-1] = L$.

Każdy bus jedzie z maksymalną prędkością, chyba że dogoni wolniejszy bus jadący przed nim. Wtedy przyjmuje prędkość wolniejszego busa i razem jadą aż do następnej stacji. Tam szybsze busy wyprzedzą wolniejsze busy.

Formalnie, dla każdego i oraz j, takich że $0 \leq i \leq N$ oraz $0 \leq j < M$, czas $t_{i,j}$ (w sekundach) kiedy bus i **przyjedzie** na stację j jest zdefiniowany następująco. Niech $t_{i,0} = T[i]$ dla każdego $0 \leq i < N$ oraz niech $t_{N,0} = Y$. Dla każdego j, takiego że $0 < j < M$:

- Oznaczmy **spodziewany czas przyjazdu** (w sekundach) busa i na stację j jako $e_{i,j}$. Jest to czas przyjazdu busa i na stację j, przy (hipotetycznym) założeniu, że od razu po przyjechaniu na stację $j-1$ pojechał on z maksymalną prędkością do stacji j. To oznacza, że:
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ dla każdego $0 \leq i < N$, oraz
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- Czas przyjazdu busa i na stację j to *maksimum* z spodziewanych czasów przyjazdu na stację j busa i oraz busów, które przyjechały na stację $j-1$ przed nim. Formalnie, $t_{i,j}$ jest równe maksimum spośród $e_{i,j}$ oraz wszystkich $e_{k,j}$ spełniających $0 \leq k \leq N$ oraz $t_{k,j-1} < t_{i,j-1}$.

Organizatorzy IOI chcą zaplanować przejazd rezerwowego busa (busa N). Twoim zadaniem jest odpowiedzieć na Q pytań organizatorów, które są następującej postaci: mając dany czas Y (w sekundach) wyjazdu busa N z lotniska, jaki jest jego czas przyjazdu do hotelu?

Szczegóły implementacji

Twoim zadaniem jest zaimplementować następujące procedury.

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

- L: długość drogi.
- N: liczba zaplanowanych (tj. nie rezerwowych) busów.
- T: tablica długości N opisująca czasy wyjazdu z lotniska zaplanowanych busów.
- W: tablica długości N opisująca maksymalne prędkości zaplanowanych busów.
- X: liczbą sekund, które rezerwowy bus potrzebuje na przejechanie 1 kilometra.
- M: liczba stacji.
- S: tablica długości M opisująca odległości stacji od lotniska.
- Ta procedura jest wywołana dokładnie raz w każdym przypadku testowym zanim wystąpią jakiekolwiek wywołania procedury `arrival_time`.

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

- Y: czas wyjazdu rezerwowego busa z lotniska
- Ta procedura powinna zwrócić czas przyjazdu busa N do hotelu.
- Ta procedura zostanie wywołana dokładnie Q razy.

Przykład

Rozważmy następujący ciąg wywołań procedur:

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

Pomijając busa 4 (rezerwowego), poniższa tabela przedstawia spodziewane i faktyczne czasy przyjazdu zaplanowanych busów na każdą ze stacji:

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

Czasy przyjazdów na stację 0 są równe czasom wyjazdów busów z lotniska, tj. $t_{i,0} = T[i]$ dla $0 \le i \le 3$.

Spodziewane i faktyczne czasy przyjazdów na stację 1 są obliczane w następujący sposób:

- Spodziewane czasy przyjazdu na stację 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: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Faktyczne czasy przyjazdu na stację 1:
  - Busy 1 i 3 przyjeżdżają na stację 0 przed busem 0, więc $t_{0,1} = \max(e_{0,1}, e_{1,1}, e_{3,1}) = 30$.
  - Bus 3 przyjeżdża na stację 0 przed busem 1, więc $t_{1,1} = \max(e_{1,1}, e_{3,1}) = 30$.
  - Bus 0, bus 1 i bus 3 przyjeżdżają na stację 0 przed busem 2, więc $t_{2,1} = \max(e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}) = 60$.
  - Żaden bus nie przyjeżdża na stację 0 przed busem 3, więc $t_{3,1} = \max(e_{3,1}) = 30$.

```
arrival_time(0)
```

Bus 4 przejeżdża 1 kilometr w 10 sekund i wyjeżdża z lotniska o czasie 0. W tym przypadku, poniższa tabela pokazuje czasy przyjazdów każdego busa. Jedyna różnica w czasach przyjazdu zaplanowanych busów została podkreślona.

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

Bus 4 przyjeżdża do hotelu o czasie 60, więc procedura powinna zwrócić 60.

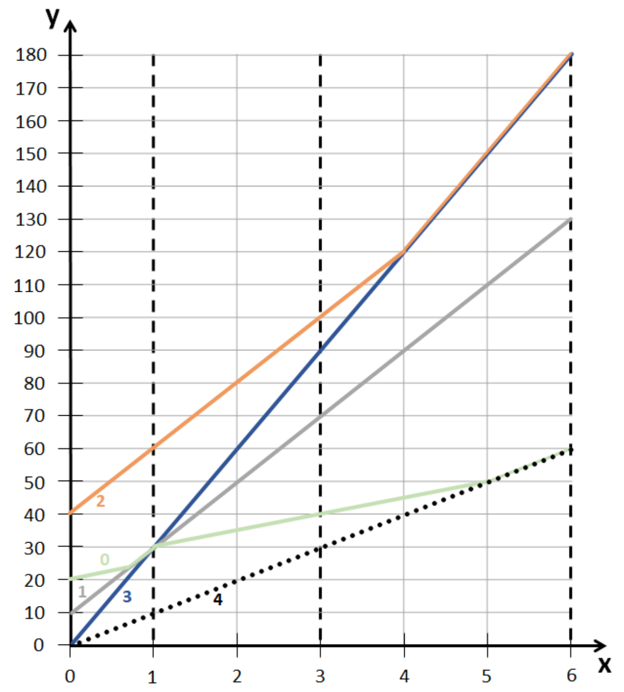
```
arrival_time(50)
```

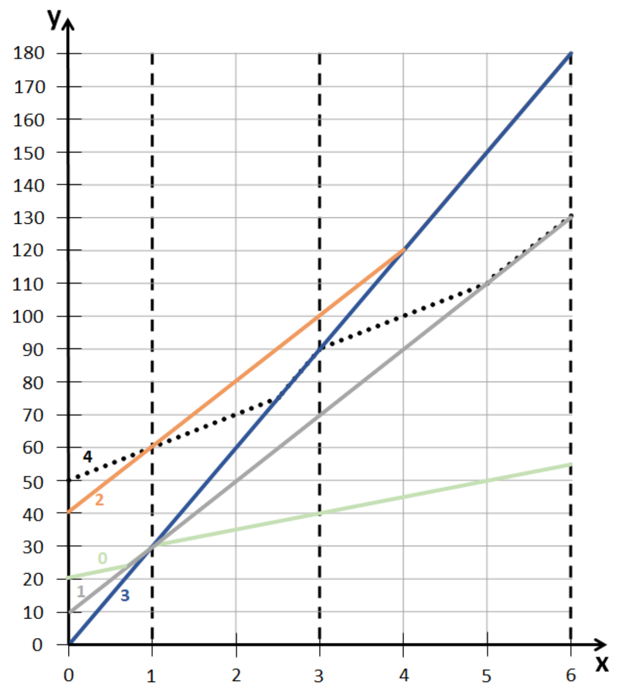
Bus 4 teraz wyjeżdża z lotniska w momencie 50. W tym przypadku czasy przyjazdów zaplanowanych busów nie zmieniają się. Wszystkie czasy przyjazdu znajdują się w poniższej tabeli.

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

Bus 4 wyprzedza wolniejszy bus 2 na stacji 1, ponieważ przyjeżdżają na nią w tym samym momencie. Następnie bus 4 spotyka się z busem 3 pomiędzy stacją 1 a stacją 2, co sprawia, że bus 4 przyjeżdża na stację 2 o czasie 90, a nie 80. Po wyjechaniu ze stacji 2, bus 4 spotyka się z busem 1 i razem jadą aż do hotelu. Bus 4 przyjeżdża do hotelu w momencie 130. Zatem, procedura powinna zwrócić 130.

Na poniższym wykresie narysowano czas, jaki zajmie każdemu busowi przejechanie odpowiedniego dystansu od lotniska. Oś OX reprezentuje dystans od lotniska (w kilometrach), a oś OY reprezentuje czas (w sekundach). Pionowe, przerywane linie oznaczają miejsca, w których znajdują się stacje. Kolorowe, nieprzerywane linie (i towarzyszące im numery busów) reprezentują cztery zaplanowane busy. Kropkowane, czarne linie reprezentują rezerwowy bus.

<table>
<tr><th><code>arrival_time(0)</code></th><th><code>arrival_time(50)</code></th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Ograniczenia

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

Podzadania

1. (9 punktów) $N = 1, Q \leq 1\,000$
2. (10 punktów) $M = 2, Q \leq 1\,000$
3. (20 punktów) $N, M, Q \leq 100$
4. (26 punktów) $Q \leq 5\,000$
5. (35 punktów) Brak dodatkowych ograniczeń.

Przykładowy program oceniający

Przykładowy program oceniający wczytuje wejście w następującym formacie:

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

Przykładowy program oceniający wypisuje odpowiedzi w następującym formacie:

- wiersz $1+k$ ($0 \leq k < Q$): wartość zwrócona przez `arrival_time` dla zapytania k