Aplenkimas

Budapešto oro uostą ir Forrás viešbutį jungia vienos krypties ir vienos juostos kelias. Kelio ilgis L kilometrų.

IOI 2023 metu $N + 1$ dalyvius vežančių autobusų važiuoja šiuo keliu. Autobusai sunumeruoti nuo 0 iki N. Autobusas i ($0 \leq i < N$) pagal tvarkaraštį turi išvykti iš oro uosto $T[i]$-ąją renginio sekundę ir gali nuvažiuoti 1 kilometrą $W[i]$ sekundžių. Autobusas N yra atsarginis ir jis 1 kilometrą gali nuvažiuoti per X sekundžių. Taip pat nėra fiksuotas laikas Y, kada šis autobusas išvyks iš oro uosto.

Kelyje lenkti draudžiama, tačiau autobusams leidžiama vieniems kitus aplenkti **rikiavimo stotelėse**. Kelyje yra M ($M > 1$) rikiavimo stotelių, sunumeruotų nuo 0 iki $M - 1$. Visos rikiavimo stotelės yra skirtingose pozicijose. Rikiavimo stotelė j ($0 \leq j < M$) yra $S[j]$ kilometrų atstumu nuo oro uosto. Rikiavimo stotelės surikiuotos atstumų nuo oro uosto didėjimo tvarka, t. y. $S[j] < S[j + 1]$ visiems $0 \leq j \leq M - 2$. Pirmoji rikiavimo stotelė yra oro uoste, o paskutinė – prie viešbučio, t. y. $S[0] = 0$ ir $S[M - 1] = L$.

Kiekvienas autobusas keliauja didžiausiu galimu savo greičiu, nebent pasiveja priekyje jo kelyje esantį autobusą; tokiu atveju autobusai papuola į kamštį ir turi keliauti lėtesniojo greičiu iki kol pasiekia kitą rikiavimo stotelę. Rikiavimo stotelėje greitesni autobusai aplenkia lėtesnius.

Kiekvienam i ir j ($0 \leq i \leq N$ ir $0 \leq j < M$), laikas $t_{i,j}$ (sekundėmis), kai autobusas i **atkeliauja** į rikiavimo stotelę j, yra formaliai apibrėžtas žemiau. Pažymėkime $t_{i,0} = T[i]$ kiekvienam $0 \leq i < N$, ir pažymėkime $t_{N,0} = Y$. Kiekvienam j, kuriam $0 < j < M$:

- Apibrėžkime autobuso i **numatomą atvykimo laiką** (sekundėmis) į rikiavimo stotelę j, žymimą $e_{i,j}$, kaip laiką, kada autobusas i atkeliautų į rikiavimo stotelę j, jeigu keliautų didžiausiu galimu autobuso greičiu nuo atvykimo laiko į rikiavimo stotelę $j - 1$. T. y.:
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j - 1])$ kiekvienam $0 \leq i < N$, ir
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j - 1])$.
- Autobuso i akeliavimo į rikiavimo stotelę j laikas yra *didžiausias* iš numatomų atvykimo laikų: autobuso i bei kiekvieno kito autobuso, kuris atvyko į stotelę $j - 1$ anksčiau, nei autobusas i. Kitaip tariant, $t_{i,j}$ yra didžiausias iš $e_{i,j}$ ir visų $e_{k,j}$ kiekvienam $0 \leq k \leq N$, kur $t_{k,j-1} < t_{i,j-1}$.

IOI organizatoriai nori suplanuoti atsarginio autobuso (autobuso N) išvykimo laiką. Atsakykite į Q organizatorių tokios formos klausimų: jei atsarginio autobuso išvykimo iš oro uosto laikas yra Y (sekundėmis), kokiu laiku jis pasieks į viešbutį?

Realizacija

Parašykite šias funkcijas.

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

- L: kelio ilgis.
- N: neatsarginių autobusų kiekis.
- T: N dydžio masyvas, nurodantis neatsarginių autobusų išvykimo iš oro uosto laikus.
- W: N ilgio masyvas, nurodantis didžiausią neatsarginių autobusų greitį.
- X: laikas, per kurį atsarginis autobusas nuvažiuoja 1 kilometrą.
- M: rikiavimo stotelių skaičius.
- S: M dydžio masyvas, nurodantis atstumus nuo oro uosto iki rikiavimo stotelių.
- Ši funkcija kviečiama lygiai vieną kartą kiekvienam testui prieš iškviečiant `arrival_time`.

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

- Y: laikas, kada atsarginis autobusas (autobusas N) išvažiuos iš oro uosto.
- Ši funkcija turėtų grąžinti laiką, kada atsarginis autobusas atvyks į viešbutį.
- Ši funkcija iškviečiama lygiai Q kartų.

Pavyzdys

Panagrinėkime šiuos iškvietimus:

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

Žemiau esančioje lentelėje pavaizduoti numatomi ir tikri neatsarginių autobusų atvykimo laikai į kiekvieną rikiavimo stotelę, jei ignoruojame 4-ą autobusą (kuriam dar nepriskirtas išvykimo laikas):

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

Atvykimo laikai į 0-inę stotelę yra autobusų išvykimo iš oro uosto laikai. T. y. $t_{i,0} = T[i]$ visiems $0 \le i \le 3$.

Numatyti ir tikri atvykimo į 1-ą rikiavimo stotelę laikai skaičiuojami taip:

- Numatyti atvykimo laikai į 1-ąją stotelę:
  - Autobusas 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - Autobusas 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - Autobusas 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - Autobusas 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Tikri atvykimo laikai į 1-ą stotelę:
  - Autobusai 1 ir 3 atvyksta į 0-inę stotelę anksčiau, nei autobusas 0, taigi $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - Autobusas 3 atvyksta į 0-inę stotelę anksčiau nei autobusas 1, taigi $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Autobusas 0, autobusas 1 ir autobusas 3 atvyksta į 0-inę rikiavimo stotelę anksčiau nei autobusas 2, taigi $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Joks autobusas neatvyksta į 0-inę stotelę anksčiau nei autobusas 3, taigi $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Autobusui 4 reikia 10 sekundžių nuvažiuoti 1 kilometrą, ir jis išvyksta iš oro uosto 0-inę sekundę. Lentelėje žemiau nurodyti kiekvieno autobuso atvykimo laikai. Vienintelis numatomų ir tikrų laikų pasikeitimas tarp neatsarginių autobusų yra pabrauktas.

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

Matome, kad autobusas 4 atvyksta į viešbutį 60-ą sekundę. Taigi, funkcija turėtų grąžinti 60.

```
arrival_time(50)
```

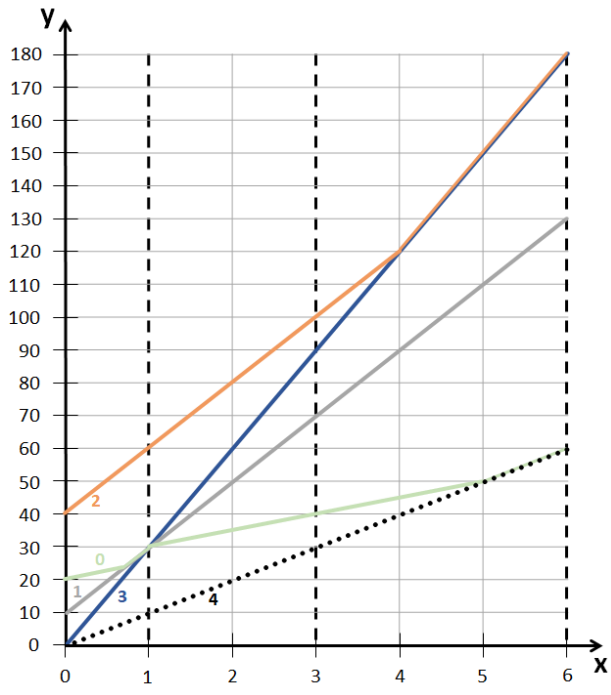
Autobusas 4 dabar išvyksta iš oro uosto 50-ą sekundę. Šiuo atveju neatsarginių autobusų atvykimo laikai, lyginant su pirmąja lentele, nepakito ir yra parodyti lentelėje:

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

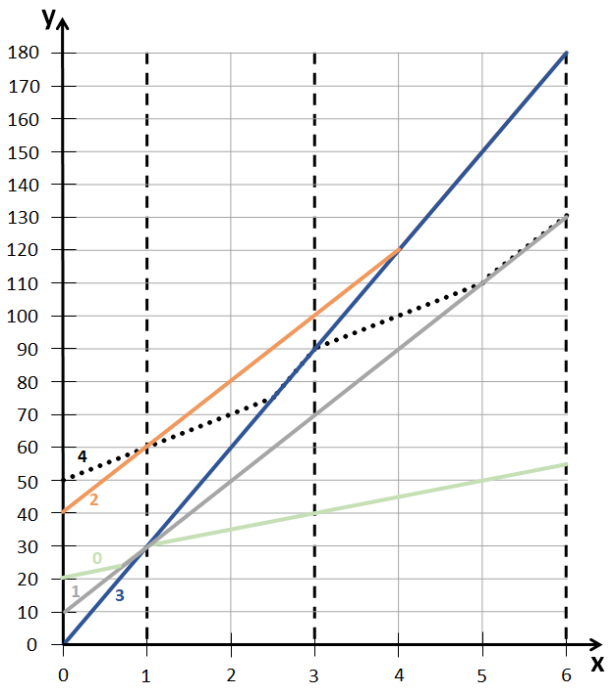
Autobusas 4 aplenkia lėtesnį autobusą 2 1-oje rikiavimo stotelėje, nes jie atvyksta tuo pačiu metu. Toliau autobusas 4 patenka į kamštį už autobuso 3 tarp 1-os ir 2-os stotelių, taigi autobusas 4 atvyksta į 2-ą stotelę 90-ą sekundę vietoj 80-os. Palikęs 2-ą stotelę 4-as autobusas patenka į kamštį už 1-o autobuso iki atvykimo į viešbutį. Autobusas 4 atvyksta į viešbutį 130-ą sekundę. Taigi funkcija turėtų grąžinti 130.

Galima nubraižyti grafiką parodyti, kiek toli nuo oro uosto autobusas bus nuvažiavęs praėjus kiekvienam laiko kiekiui. Grafiko X ašis nurodo atstumą nuo oro uosto (kilometrais), o Y ašis nurodo laiką (sekundėmis). Vertikalios punktyrinės linijos žymi rūšiavimo stotelių pozicijas. Skirtingos ištisinės linijos (su šalia užrašytais autobusų indeksais) nurodo keturių neatsarginių autobusų maršrutus. Taškuota juoda linija nurodo atsarginio autobuso maršrutą.

`arrival_time(0)`



`arrival_time(50)`



Ribojimai

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

Dalinės užduotys

1. (9 taškai) $N = 1, Q \le 1\,000$
2. (10 taškų) $M = 2, Q \le 1\,000$
3. (20 taškų) $N, M, Q \le 100$
4. (26 taškai) $Q \le 5\,000$
5. (35 taškai) Papildomų reikalavimų nėra.

Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa skaito duomenis šiuo formatu:

- 1-a eilutė: $L\ N\ X\ M\ Q$
- 2-a eilutė: $T[0]\ T[1]\ \ldots\ T[N-1]$
- 3-a eilutė: $W[0]\ W[1]\ \ldots\ W[N-1]$
- 4-a eilutė: $S[0]\ S[1]\ \ldots\ S[M-1]$
- $(5+k)$-a eilutė ($0 \le k < Q$): Y vertė k-ajai užklausai

Pavyzdinė vertinimo programa atsakymą išveda šiuo formatu:

- $(1+k)$-a eilutė ($0 \le k < Q$): funkcijos `arrival_time` grąžinta vertė k-ajai užklausai.