Möödasõidud

Budapesti lennujamast Forrási hotelli viib ühesuunaline ühe sõidurajaga L kilomeetri pikkune maantee.

IOI 2023 jooksul sõidavad seda maanteed mööda $N+1$ bussi, mis on nummerdatud 0 kuni N. Buss number i (kus $0 \leq i < N$) väljub lennujaamast sekundil $T[i]$ ja suudab kilomeetri läbida $W[i]$ sekundiga. Buss number N on lisabuss, mis suudab kilomeetri läbida X sekundiga, aga mille lennujaamast väljumise aeg Y pole veel otsustatud.

Ühe sõidurajaga maanteel pole möödasõitmine üldiselt võimalik. Bussid saavad üksteisest mööduda ainult spetsiaalsetes **vahejaamades**. Maanteel on erinevates punktides kokku M ($M > 1$) vahejaama, mis on nummerdatud 0 kuni $M-1$. Vahejaam number j (kus $0 \leq j < M$) asub maanteel $S[j]$ kilomeetri kaugusel lennujaamast. Vahejaamad on antud nende kauguste kasvamise järjekorras: iga $0 \leq j \leq M-2$ korral $S[j] < S[j+1]$. Esimene vahejaam on alati lennujaamas ja viimane hotellis: $S[0] = 0$ ja $S[M-1] = L$.

Iga buss sõidab oma maksimaalse kiirusega, kuni ta jõuab järele eespool sõitvale aeglasemale bussile. Seejärel sõidavad bussid kuni järgmise vahejaamani aeglasema bussi kiirusega. Vahejaamas sõidavad kiiremad bussid aeglasematest mööda.

Formaalsemalt on iga $0 \leq i \leq N$ ja $0 \leq j < M$ korral bussi i vahejaama j **saabumise aeg** määratud järgmiselt. Olgu iga $0 \leq i < N$ korral $t_{i,0} = T[i]$ ja $t_{N,0} = Y$. Siis iga $0 < j < M$ korral:

- Defineerime bussi i vahejaama j **oodatava saabumisaja** $e_{i,j}$ kui aja, mil buss i saabuks vahejaama j, kui see buss sõidaks oma maksimaalse kiirusega alates vahejaama $j-1$ saabumise ajast. Teisisõnu:
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ iga $0 \leq i < N$ korral,
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- Bussi i tegelik vahejaama j saabumise aeg on bussi i ja kõigi temast varem vahejaama $j-1$ saabunud busside oodatavate vahejaama j saabumise aegade *maksimum*. Formaalsemalt, $t_{i,j}$ on $e_{i,j}$ ja kõigi selliste $e_{k,j}$ maksimum, kus $0 \leq k \leq N$ ja $t_{k,j-1} < t_{i,j-1}$.

IOI korraldajad tahavad nüüd kasutada lisabussi (buss number N). Sinu ülesanne on vastata Q küsimusele kujul "kui lisabuss väljub lennujaamast sekundil Y, siis millisel sekundil see hotelli juurde jõuab?"

Realisatsioon

Lahendusena tuleb realiseerida järgmised funktsioonid:

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

- L: maantee pikkus.
- N: tavabusside arv.
- T: N elemendiga massiiv: tavabusside lennujaamast väljumise ajad.
- W: N elemendiga massiiv: tavabusside maksimumkiirused.
- X: lisabussi maksimumkiirus.
- M: vahejaamade arv.
- S: M elemendiga massiiv: vahejaamade kaugused lennujaamast.
- Seda funktsiooni käivitatakse igas testis täpselt üks kord ja seda tehakse enne funktsiooni `arrival_time` esimest käivitamist.

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

- Y: lisabussi (buss N) lennujaamast väljumise aeg.
- See funktsioon peab tagastama lisabussi hotelli jõudmise aja.
- Seda funktsiooni käivitatakse täpselt Q korda.

Näide

Vaatleme järgmist väljakutsete jada:

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

Bussi number 4 (mille lennujaamast väljumise aeg pole veel teada) arvestamata on teiste busside oodatavad ja tegelikud vahejaamadesse saabumise ajad:

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

Vahejaama 0 saabumise ajad on lennujaamast väljumise ajad: $t_{i,0} = T[i]$ iga $0 \le i \le 3$ korral.

Vahejaama 1 saabumise oodatavad ja tegelikud ajad arvutame järgmiselt:

- Oodatavad vahejaama 1 saabumise ajad:
  - Buss 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.

  - Buss 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30.$
  - Buss 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60.$
  - Buss 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30.$
- Tegelikud vahejaama 1 saabumise ajad:
  - Bussid 1 ja 3 saabuvad vahejaama 0 enne bussi $0,$ seega $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30.$
  - Buss 3 saabub vahejaama 0 enne bussi 1, seega $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30.$
  - Bussid $0,$ 1 ja 3 saabuvad vahejaama 0 enne bussi $2,$ seega $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60.$
  - Ükski buss ei saabu vahejaama 0 enne bussi 3, seega $t_{3,1} = \max([e_{3,1}]) = 30.$

```
arrival_time(0)
```

Buss 4 läbib kilomeetri 10 sekundiga ja väljub lennujaamast sekundil 0. Sel juhul on busside vahejaamadesse saabumise ajad sellised, nagu allolevas tabelis. Tavabusside saabumise aegades on eelmise tabeliga võrreldes ainult üks muutus, millele on joon alla tõmmatud.

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

Nagu näha, saabub lisabuss sel juhul hotelli sekundil 60. Seega peab funktsioon `arrival_time` tagastama arvu 60.

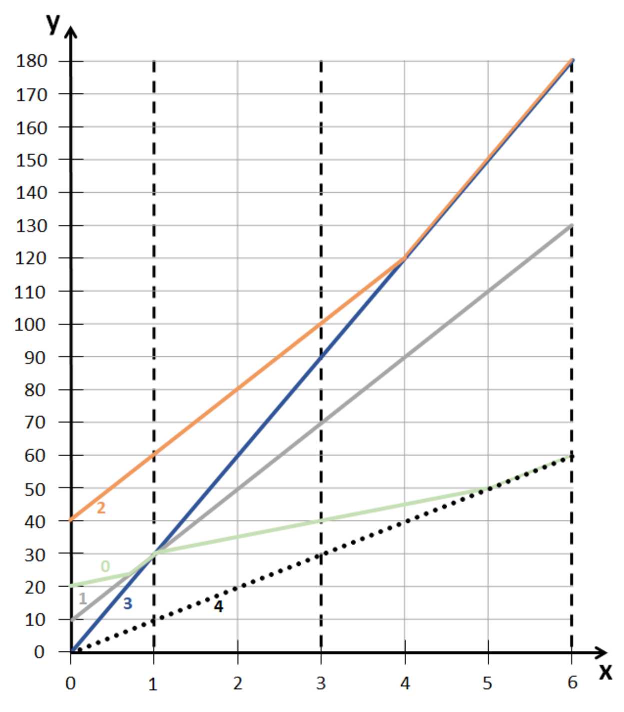
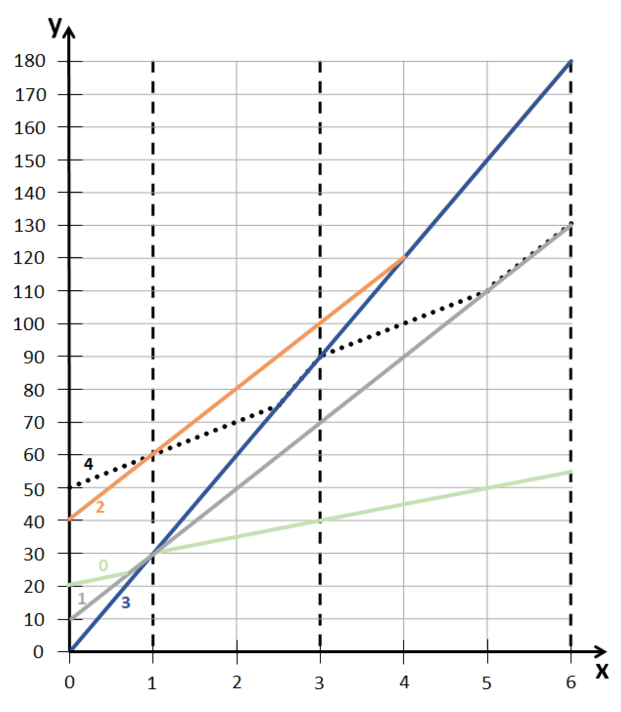
```
arrival_time(50)
```

Buss 4 ja väljub lennujaamast sekundil 50. Sel juhul pole tavabusside vahejaamadesse saabumise aegades esialgse tabeliga võrreldes mingeid muutusi. Kõigi busside vahejaamadesse saabumise ajad on näidatud allolevas tabelis.

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

Buss 4 möödub aeglasemast bussist 2 vahejaamas 1, kuhu nad saabuvad üheaegselt. Edasi jääb buss 4 vahejaamade 1 ja 2 vahel bussi 3 taha kinni ja saabub vahejaama 2 sekundi 80 asemel sekundil 90. Jaamast 2 väljumise järel jääb buss 4 hotelli saabumiseni kinni bussi 1 taha. Buss 4 saabub hotelli sekundil 130. Seega peab funktsioon `arrival_time` tagastama 130.

Allolevad joonised kujutavad busside liikumist graafiliselt. X-teljel on busside kaugused lennujaamast (kilomeetrites) ja Y-teljel aeg (sekundites). Vertikaalsed katkendjooned näitavad vahejaamade asukohti. Värvilised pidevjooned (millest igaühe juures on ka bussi number) kujutavad tavabusside liikumist. Punktiirjoon kujutab lisabussi liikumist.

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

Piirangud

- $1 \le L \le 10^9$.
- $1 \le N \le 1\,000$.

- $0 \leq T[i] \leq 10^{18}$ iga $0 \leq i < N$ korral.
- $1 \leq W[i] \leq 10^9$ iga $0 \leq i < N$ korral.
- $1 \leq X \leq 10^9$.
- $2 \leq M \leq 1\,000$.
- $0 = S[0] < S[1] < \cdots < S[M-1] = L$.
- $1 \leq Q \leq 10^6$.
- $0 \leq Y \leq 10^{18}$.

Alamülesanded

1. (9 punkti) $N = 1$, $Q \leq 1\,000$.
2. (10 punkti) $M = 2$, $Q \leq 1\,000$.
3. (20 punkti) $N, M, Q \leq 100$.
4. (26 punkti) $Q \leq 5\,000$.
5. (35 punkti) Lisapiirangud puuduvad.

Hindamisprogramm

Arhiivis olev hindamisprogramm loeb sisendi järgmises vormingus:

- rida 1: $L\ N\ X\ M\ Q$
- rida 2: $T[0]\ T[1]\ \ldots\ T[N-1]$
- rida 3: $W[0]\ W[1]\ \ldots\ W[N-1]$
- rida 4: $S[0]\ S[1]\ \ldots\ S[M-1]$
- rida $5+k$ (kus $0 \leq k < Q$): Y väärtus küsimuses number k

Hindamisprogramm väljastab vastuse järgmises vormingus:

- rida $1+k$ (kus $0 \leq k < Q$): funktsiooni `arrival_time` küsimusele number k tagastatud vastus