# Menyalip

Terdapat jalan satu arah dari Bandara Budapest ke Hotel Forrás. Jalan tersebut memiliki panjang $L$ kilometer.

Selama acara IOI 2023, $N+1$ bus transfer melalui jalan tersebut. Bus-bus dinomori dari $0$ sampai $N$. Bus $i$ ($0 \le i < N$) dijadwalkan untuk meninggalkan bandara pada detik ke-$T[i]$ dari acara, dan dapat menempuh $1$ kilometer dalam $W[i]$ detik. Bus $N$ adalah bus cadangan dan dapat menempuh $1$ kilometer dalam $X$ detik. Waktu keberangkatan $Y$, saat bus cadangan akan meninggalkan bandara, belum ditentukan.

Menyalip di jalan tidak diperbolehkan, namun bus-bus dapat saling menyalip di **pangkalan sortir**. Terdapat $M$ ($M > 1$) pangkalan sortir, bernomor $0$ sampai $M-1$, pada posisi yang berbeda-beda di jalan. Pangkalan sortir $j$ ($0 \le j < M$) terletak di $S[j]$ kilometer dari bandara. Pangkalan sortir diurutkan berdasarkan jarak dari yang paling dekat dari bandara, sehingga, $S[j] < S[j+1]$ untuk setiap $0 \le j \le M-2$. Pangkalan sortir pertama adalah bandara dan pangkalan sortir terakhir adalah hotel, sehingga, $S[0] = 0$ dan $S[M-1] = L$.

Setiap bus berjalan pada kecepatan maksimum kecuali saat menemui bus yang lebih lambat di depan mereka, sehingga mereka berkumpul dan terpaksa berjalan bersama dengan kecepatan bus yang lebih lambat, hingga mereka tiba di pangkalan sortir berikutnya. Di sana, bus yang lebih cepat akan menyalip bus yang lebih lambat.

Secara formal, untuk setiap $i$ dan $j$ dengan $0 \le i \le N$ dan $0 \le j < M$, waktu $t_{i,j}$ (dalam detik) saat bus $i$ **tiba di** pangkalan sortir $j$ didefinisikan sebagai berikut. Misalkan $t_{i,0} = T[i]$ untuk setiap $0 \le i < N$, dan misalkan $t_{N,0} = Y$. Untuk setiap $j$ dengan $0 < j < M$:

- Definisikan **perkiraan waktu tiba** (dalam detik) bus $i$ di pangkalan sortir $j$, dirujuk oleh $e_{i,j}$, sebagai waktu bus $i$ akan tiba di pangkalan sortir $j$ jika berjalan dengan kecepatan maksimal sejak tiba di pangkalan sortir $j-1$. Sehingga, misalkan
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ untuk setiap $0 \le i < N$, dan
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- Bus $i$ tiba di pangkalan sortir $j$ pada *maksimum* dari perkiraan waktu tiba bus $i$ dan perkiraan waktu tiba bus lainnya yang tiba di pangkalan $j-1$ sebelum bus $i$. Secara formal, misalkan $t_{i,j}$ sebagai maksimum dari $e_{i,j}$ dan setiap $e_{k,j}$ dengan $0 \le k \le N$ dan $t_{k,j-1} < t_{i,j-1}$.

Penyelenggara IOI ingin menjadwalkan bus cadangan (bus $N$). Tugas Anda adalah untuk menjawab $Q$ pertanyaan dari penyelenggara, dengan format sebagai berikut: diberikan waktu $Y$ (dalam detik) saat bus cadangan meninggalkan bandara, kapan bus tersebut akan tiba di hotel?

## Detail Implementasi

Tugas Anda adalah untuk mengimplementasikan prosedur berikut.

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

- $L$: panjang jalan.
- $N$: banyaknya bus yang bukan cadangan.
- $T$: sebuah *array* sepanjang $N$ mendeskripsikan waktu-waktu bus yang bukan cadangan dijadwalkan untuk meninggalkan bandara.
- $W$: sebuah *array* sepanjang $N$ mendeskripsikan kecepatan maksimum dari bus-bus yang bukan bukan cadangan.
- $X$: waktu yang diperlukan bus cadangan untuk menempuh 1 kilometer.
- $M$: banyaknya pangkalan sortir.
- $S$: sebuah *array* sepanjang $M$ mendeskripsikan jarak dari pangkalan-pangkalan sortir dari bandara.
- Prosedur ini dipanggil tepat sekali untuk setiap kasus uji, sebelum semua panggilan `arrival_time`.

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

- $Y$: waktu saat bus cadangan (bus $N$) meninggalkan bandara.
- Prosedur ini harus mengembalikan waktu kapan bus cadangan akan tiba di hotel.
- Prosedur ini dipanggil tepat $Q$ kali.

## Contoh

Perhatikan serangkaian panggilan berikut:

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

Hiraukan bus $4$ (yang belum dijadwalkan), tabel berikut ini menunjukkan perkiraan waktu tiba dan waktu tiba sesungguhnya dari bus-bus yang bukan cadangan di setiap pangkalan sortir:

| $i$ | | $t_{i,0}$ | | $e_{i,1}$ | $t_{i,1}$ | | $e_{i,2}$ | $t_{i,2}$ | | $e_{i,3}$ | $t_{i,3}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $0$ | | $20$ | | $25$ | $30$ | | $40$ | $40$ | | $55$ | $55$ |
| $1$ | | $10$ | | $30$ | $30$ | | $70$ | $70$ | | $130$ | $130$ |
| $2$ | | $40$ | | $60$ | $60$ | | $100$ | $100$ | | $160$ | $180$ |
| $3$ | | $0$ | | $30$ | $30$ | | $90$ | $90$ | | $180$ | $180$ |

Waktu tiba pada pangkalan $0$ adalah waktu bus-bus dijadwalkan untuk meninggalkan bandara. Sehingga, $t_{i,0} = T[i]$ untuk $0 \le i \le 3$.

Perkiraan waktu tiba dan waktu tiba sesungguhnya di pangkalan sortir $1$ dihitung sebagai berikut:

- Perkiraan waktu tiba di pangkalan $1$:
  - 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 $3$: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Waktu tiba sesungguhnya di pangkalan $1$:
  - Bus $1$ dan $3$ tiba di pangkalan $0$ sebelum bus $0$, sehingga $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - Bus $3$ tiba di pangkalan $0$ sebelum bus $1$, sehingga $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Bus $0$, bus $1$ dan bus $3$ tiba di pangkalan sortir $0$ sebelum bus $2$, sehingga $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Tidak ada bus yang tiba di pangkalan $0$ sebelum bus $3$, sehingga $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Bus $4$ memerlukan $10$ detik untuk menempuh $1$ kilometer dan kini dijadwalkan untuk meninggalkan bandara pada detik ke-$0$. Pada kasus ini, tabel berikut ini menunjukkan waktu tiba dari setiap bus. Satu-satunya perubahan terkait perkiraan waktu tiba dan waktu tiba sesungguhnya dari bus-bus yang bukan cadangan ditandai dengan garis bawah.

| $i$ | | $t_{i,0}$ | | $e_{i,1}$ | $t_{i,1}$ | | $e_{i,2}$ | $t_{i,2}$ | | $e_{i,3}$ | $t_{i,3}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $0$ | | $20$ | | $25$ | $30$ | | $40$ | $40$ | | $55$ | $\underline{60}$ |
| $1$ | | $10$ | | $30$ | $30$ | | $70$ | $70$ | | $130$ | $130$ |
| $2$ | | $40$ | | $60$ | $60$ | | $100$ | $100$ | | $160$ | $180$ |
| $3$ | | $0$ | | $30$ | $30$ | | $90$ | $90$ | | $180$ | $180$ |
| $4$ | | $0$ | | $10$ | $10$ | | $30$ | $30$ | | $60$ | $60$ |

Kita lihat bahwa bus $4$ tiba di hotel pada detik ke-$60$. Maka, prosedur harus mengembalikan $60$.

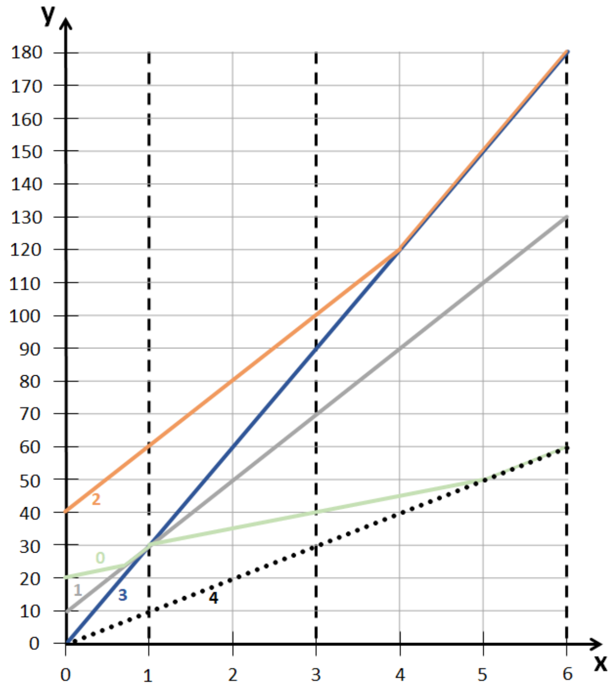
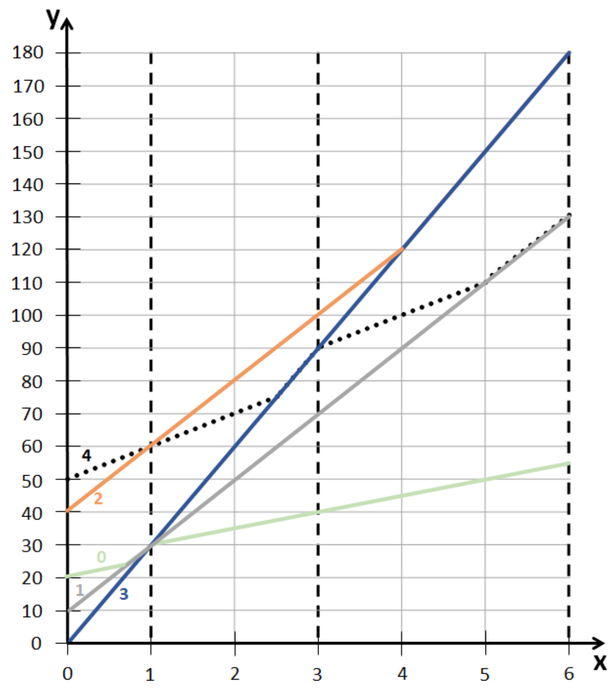
```
arrival_time(50)
```

Bus $4$ kini dijadwalkan untuk meninggalkan bandara pada detik ke-$50$. Pada kasus ini, tidak ada perubahan waktu tiba dari bus-bus yang bukan cadangan dibandingkan dengan tabel awal. Waktu tiba ditunjukkan pada tabel berikut ini.

| $i$ | | $t_{i,0}$ | | $e_{i,1}$ | $t_{i,1}$ | | $e_{i,2}$ | $t_{i,2}$ | | $e_{i,3}$ | $t_{i,3}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $0$ | | $20$ | | $25$ | $30$ | | $40$ | $40$ | | $55$ | $55$ |
| $1$ | | $10$ | | $30$ | $30$ | | $70$ | $70$ | | $130$ | $130$ |
| $2$ | | $40$ | | $60$ | $60$ | | $100$ | $100$ | | $160$ | $180$ |
| $3$ | | $0$ | | $30$ | $30$ | | $90$ | $90$ | | $180$ | $180$ |
| $4$ | | $50$ | | $60$ | $60$ | | $80$ | $90$ | | $120$ | $130$ |

Bus $4$ menyalip bus $2$ yang lebih lambat pada pangkalan sortir $1$ saat mereka bersama-sama tiba. Selanjutnya, bus $4$ berkumpul dengan bus $3$ di antara pangkalan $1$ dan pangkalan $2$, sehingga bus $4$ tiba di pangkalan $2$ pada detik ke-$90$, bukan pada detik ke-$80$. Setelah meninggalkan pangkalan $2$, bus $4$ berkumpul dengan bus $1$ hingga mereka tiba di hotel. Bus $4$ tiba di hotel pada detik ke-$130$. Maka, prosedur harus mengembalikan $130$.

Kita dapat membuat *plot* waktu yang diperlukan untuk setiap bus mencapai setiap jarak dari bandara. Sumbu x dari *plot* menyatakan jarak dari bandara (dalam kilometer) dan sumbu y dari *plot* menyatakan waktu (dalam detik). Garis patah-patah vertikal menandai posisi-posisi dari pangkalan sortir. Garis tegas berbeda (disertai dengan index bus) menyatakan empat bus yang bukan cadangan. Garis hitam titik-titik menyatakan bus cadangan.

| `arrival_time(0)` | `arrival_time(50)` |
|---|---|
|  |  |



## Batasan

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

## Subsoal

1. (9 poin) $N = 1, Q \le 1\,000$
2. (10 poin) $M = 2, Q \le 1\,000$
3. (20 poin) $N, M, Q \le 100$
4. (26 poin) $Q \le 5\,000$
5. (35 poin) Tidak ada batasan tambahan.

## Contoh *Grader*

Contoh *grader* membaca masukan dengan format berikut:

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

Contoh *grader* mencetak keluaran Anda dengan format berikut:

- baris $1+k$ ($0 \leq k < Q$): nilai kembali dari `arrival_time` untuk pertanyaan $k$