

## Episodio II: un lungo viaggio (bus)

☄️ è finalmente atterrato con la sua delegazione al cosmodromo di Baikonur.<sup>1</sup> Durante il viaggio spaziale diverse edizioni delle OII si sono susseguite, e ora è il momento di andare a Biella per l'edizione 2022!

Purtroppo, viaggiare sulla Terra non è cosa semplice per un alieno: l'astronave non è in grado di farlo, camminare sulla Terra è faticoso, e persino prendere un mezzo pubblico può sollevare troppe domande e far perdere giorni tra noiosi controlli burocratici, interrogatori, analisi scientifiche e vivisezioni.

Per questo motivo, ☄️ ha deciso che si limiterà alle sole linee di bus, prendendosi tutto il tempo necessario (per fortuna è in anticipo di mesi!) ma facendo il minor numero di cambi, per limitare il numero di controlli a cui sarà sottoposto.

Ci sono  $N$  fermate di bus sulla Terra, numerate da 0 a  $N - 1$ . Il cosmodromo di Baikonur si trova alla fermata 0, mentre l'ITIS Quintino Sella di Biella si trova alla fermata  $N - 1$ . Le fermate sono collegate da  $L$  diverse linee di bus, dove l' $i$ -esima linea ( $0 \leq i < L$ ) effettua le  $K_i$  fermate  $F_{i,0}, F_{i,1}, \dots, F_{i,K_i-1}$ , in quest'ordine. In altre parole, tale bus parte dalla fermata  $F_{i,0}$ , effettua  $K_i - 2$  fermate intermedie, e infine termina nella fermata  $F_{i,K_i-1}$ . Nota che non è detto che ogni linea di bus sia appaiata da una linea che fa le stesse fermate in ordine inverso.

☄️ vorrebbe quindi prendere un bus nella fermata 0, effettuare il minor numero possibile (anche 0) di cambi, e, per finire, scendere nella fermata  $N - 1$ . Un cambio consiste nello scendere dal bus su cui si sta viaggiando a una certa fermata  $j$  e salire su un altro bus (eventualmente anche della stessa linea) alla medesima fermata. Aiuta ☄️ a pianificare gli spostamenti nel modo migliore, o determina che è impossibile raggiungere la fermata  $N - 1$  nel modo descritto!



Figura 1: Il bus di linea 26 per Biella centrale.

## Implementazione

Dovrai sottoporre un unico file, con estensione `.cpp`.

 Tra gli allegati a questo task troverai un template `bus.cpp` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

```
C++ | int pianifica(int N, int L, vector<vector<int>> F);
```

- L'intero  $N$  rappresenta il numero di fermate.
- L'intero  $L$  rappresenta il numero di linee di bus.
- Il vettore di vettori  $F$ , indicizzato da 0 a  $L - 1$ , contiene le informazioni sulle fermate degli bus. Più precisamente, per  $0 \leq i < L$ ,  $F[i]$  è un vettore di lunghezza  $K_i$  che contiene gli interi  $F_{i,0}, F_{i,1}, \dots, F_{i,K_i-1}$ , in quest'ordine.
- La funzione deve restituire il numero minimo di cambi necessari affinché ☄️ possa raggiungere la fermata  $N - 1$  dalla fermata 0, oppure  $-1$  se questo è impossibile.

<sup>1</sup>Il più grande spaziorporto sulla terra attualmente operativo, situato in Kazakistan.

## Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama la funzione che dovete implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da  $L + 1$  righe, contenenti:

- Riga 1: gli interi  $N$  e  $L$ , separati da uno spazio.
- Riga  $2 + i$  ( $0 \leq i < L$ ): l'intero  $K_i$ , seguito dai  $K_i$  interi  $F_{i,0}, \dots, F_{i,K_i-1}$ .

Il file di output è composto da una sola riga, contenente il valore  $S$  restituito dalla funzione `pianifica`.

## Assunzioni

- $2 \leq N \leq 100\,000$ .
- $1 \leq L \leq 100\,000$ .
- $K_i \geq 2$  per ogni  $0 \leq i < L$ .
- $K_0 + K_1 + \dots + K_{L-1} \leq 300\,000$ .
- $0 \leq F_{i,j} < N$  per ogni  $0 \leq i < L$  e  $0 \leq j < K_i$ .
- $F_{i,j} \neq F_{i,j+1}$  per ogni  $0 \leq i < L$  e  $0 \leq j < K_i - 1$ .

## Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test che lo compongono.

- **Subtask 1 [ 0 punti]**: Casi d'esempio.
- **Subtask 2 [17 punti]**:  $K_i = 2$  per ogni  $0 \leq i < L$ .
- **Subtask 3 [20 punti]**:  $N \leq 2000$ . Inoltre  $K_0 + K_1 + \dots + K_{L-1} \leq 4000$ .
- **Subtask 4 [19 punti]**:  $F_{i,j} < F_{i,j+1}$  per ogni  $0 \leq i < L$  e  $0 \leq j < K_i - 1$ .
- **Subtask 5 [12 punti]**:  $K_i$  è dispari e  $F_{i,j} = F_{i,K_i-j-1}$  per ogni  $0 \leq i < L$  e  $0 \leq j < K_i - 1$  (ogni bus ripercorre il proprio tragitto nella direzione opposta).
- **Subtask 6 [14 punti]**: Ogni stazione compare al più 50 volte nell'input.
- **Subtask 7 [18 punti]**: Nessuna limitazione aggiuntiva.

## Esempi di input/output

stdin	stdout
8 3 7 2 3 6 1 4 3 5 2 6 7 2 0 1	3
4 2 4 2 3 1 0 2 0 1	-1

stdin	stdout
5 3 5 0 3 1 2 4 2 1 4 2 0 1	0

## Spiegazione

Nel **primo caso di esempio**, la soluzione ottimale consiste in:

- prendere la **linea 2** in corrispondenza della fermata 0;
- cambiare in corrispondenza della fermata 1 e prendere la **linea 0**;
- scendere alla fermata 3 e salire nuovamente sulla **linea 0**;
- cambiare in corrispondenza della fermata 6 e prendere la **linea 1**;
- scendere infine alla fermata 7.

La soluzione ottimale si compone quindi di 3 cambi.

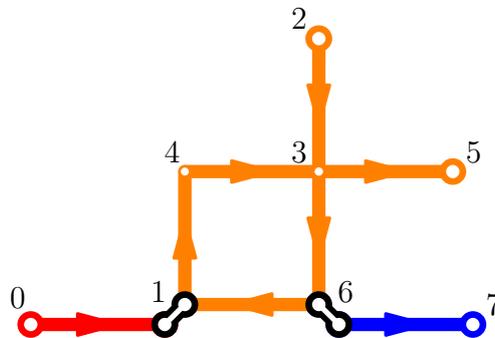


Figura 2: Primo caso di esempio

Nel **secondo caso di esempio**, non è possibile raggiungere la fermata 3 partendo dalla fermata 0.

Nel **terzo caso di esempio**, la soluzione ottimale consiste nel prendere la **linea 0** dalla fermata 0 alla fermata 4, senza effettuare nessun cambio. Si noti che è anche possibile raggiungere la fermata 4 prendendo la **linea 2** e poi cambiando alla fermata 1 con la **linea 1**, tuttavia questa soluzione richiede un numero maggiore di cambi.

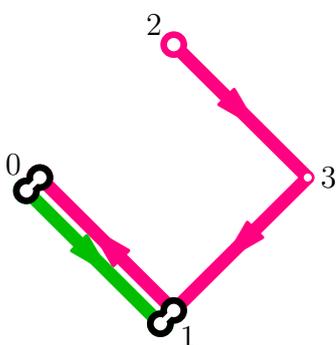


Figura 3: Secondo caso di esempio

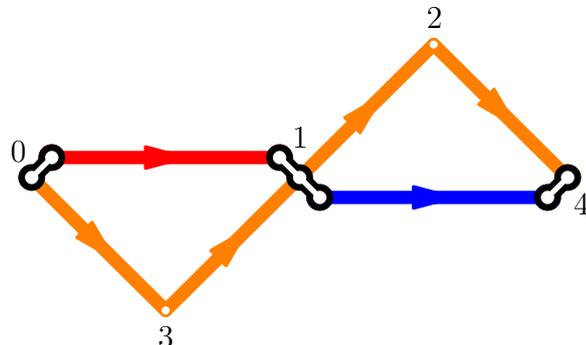


Figura 4: Terzo caso di esempio