

Scale di Hogwarts (hogwarts)

A Hogwarts, la più prestigiosa scuola di magia del mondo, si sa che *alle scale piace cambiare!* Dopo un lungo viaggio in treno e la cerimonia di smistamento, sei pronto per la tua prima lezione: Pozioni. Il castello ha N sale, numerate da 0 a $N - 1$, tra loro collegate da M scale. Il tuo dormitorio si trova nella sala 0 e la lezione di Pozioni si tiene nell'aula allestita nella stanza $N - 1$, nei sotterranei del castello. Fortunatamente, prima di intraprendere il percorso attraverso le sale, conosci l'orario in cui compare e quello in cui scompare ogni scala. Per percorrere una scala impieghi esattamente 1 minuto, ma puoi sostare nelle sale per tutto il tempo che ritieni necessario.

Il professore di Pozioni è molto severo e non tollera ritardi, dunque devi assolutamente arrivare a lezione nel minor tempo possibile. Trova un modo per raggiungere l'aula nel minimo tempo possibile, se un modo per raggiungerla esiste!

Implementazione

Dovrai sottoporre un unico file, con estensione `.c`, `.cpp` o `.pas`.

📎 Tra gli allegati a questo task troverai un template `hogwarts.c`, `hogwarts.cpp`, `hogwarts.pas` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

C/C++	<code>int raggiungi(int N, int M, int A[], int B[], int inizio[], int fine[]);</code>
Pascal	<code>function raggiungi(N,M: longint; A,B,inizio,fine: array of longint): longint;</code>

- L'intero N rappresenta il numero di sale del castello.
- L'intero M rappresenta il numero di scale.
- Gli array A , B , $inizio$ e $fine$ sono indicizzati da 0 a $M - 1$ e contengono le informazioni sulla comparsa e sparizione delle scale: l' i -esima scala collega tra loro le sale $A[i]$ e $B[i]$, compare al tempo $inizio[i]$ e scompare al tempo $fine[i]$.
- La funzione deve restituire il minimo tempo necessario per andare dalla sala 0 alla sala $N - 1$; se non è possibile raggiungere la sala $N - 1$, deve restituire il valore -1 .

Il grader chiamerà la funzione `raggiungi` e ne stamperà il valore restituito sul file di output.

Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per verificare 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 $M + 1$ righe, contenenti:

- Riga 1: gli interi N e M .
- Righe 2, \dots , $M + 1$: l' i -esima di queste righe contiene, nell'ordine, i valori $A[i]$, $B[i]$, $inizio[i]$ e $fine[i]$ per $i = 0, \dots, M - 1$.

Il file di output è composto da un'unica riga, contenente:

- Riga 1: il valore restituito dalla funzione `raggiungi`.

Assunzioni

- $2 \leq N \leq 500\,000$.
- $1 \leq M \leq 1\,000\,000$.
- $0 \leq A[i], B[i] \leq N - 1$ per ogni $i = 0, \dots, M - 1$.
- Non ci sono scale che collegano una sala a se stessa ($A[i] \neq B[i]$).
- Non ci sono due o più scale che collegano le stesse due sale.
- Ogni scala è percorribile in una qualunque delle due direzioni.
- $0 \leq \text{inizio}[i] < \text{fine}[i] \leq 2\,000\,000$.

Assegnazione del punteggio

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

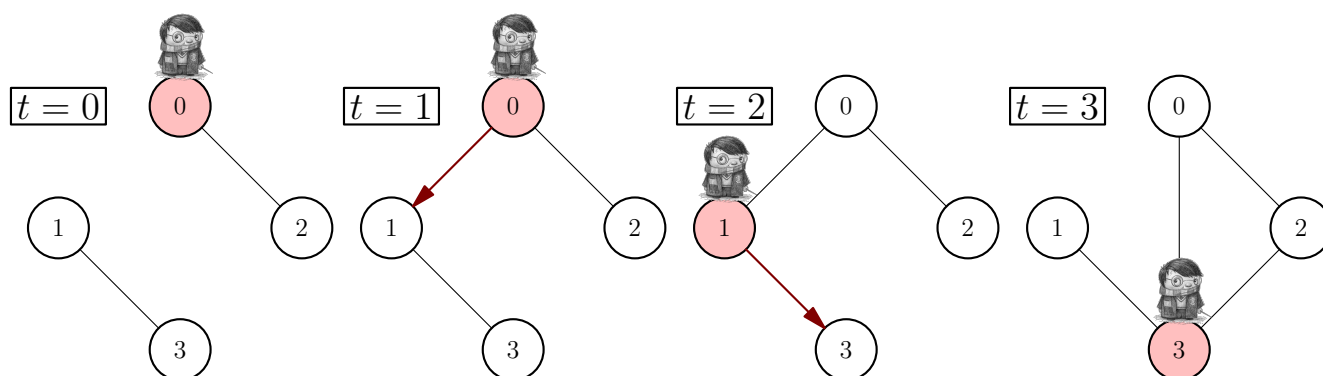
- **Subtask 1 [0 punti]**: Casi d'esempio.
- **Subtask 2 [10 punti]**: $N \leq 10, M \leq 15$ e $\text{fine}[i] \leq 20$ per ogni i .
- **Subtask 3 [21 punti]**: Tutte le scale sono fisse, cioè il tempo di inizio è 0 per tutte e il tempo di fine è uguale per tutte.
- **Subtask 4 [18 punti]**: Le scale scompaiono soltanto, cioè il tempo di inizio è 0 per tutte.
- **Subtask 5 [22 punti]**: $N \leq 1000, M \leq 2000, \text{fine}[i] \leq 5000$ per ogni i .
- **Subtask 6 [29 punti]**: Nessuna limitazione specifica.

Esempi di input/output

4 5 0 2 0 5 0 1 1 3 0 3 3 6 3 2 3 8 3 1 0 10	3
3 2 0 1 3 5 1 2 2 4	-1

Spiegazione

Nel **primo caso di esempio** il modo più veloce per andare dalla sala 0 alla 3 è aspettare 1 minuto, poi prendere la scala che collega 0 e 1 (ci metti 1 minuto) e poi prendere immediatamente la scala che collega 1 e 3 (anche qui ci metti 1 minuto), impiegando in totale 3 minuti per arrivare a lezione.



Nel **secondo caso di esempio** non è possibile andare dalla sala 0 alla sala 2! Infatti dovresti necessariamente passare per la sala 1 perché non c'è mai una scala che collega direttamente 0 e 2. Al tempo 3 compare una scala per andare da 0 a 1, dunque puoi trovarti nella stanza 1 al più presto dopo 4 minuti, e in quell'istante scompare la scala che collega le sale 1 e 2, impedendoti di raggiungere la destinazione.

