



Smaltimento sostanze tossiche (smaltimento)

Per costruire il processore del suo supercomputer, Luca ha acquistato 1 barile di Acido Olimpico (formula chimica OI_2), una sostanza molto tossica, e che ora deve smaltire al minor costo possibile e senza fare danni all'ambiente. Ha a disposizione vari processi industriali che smaltiscono una sostanza, ma che possono a loro volta produrre nuove sostanze tossiche da smaltire.

Ci sono N sostanze tossiche, numerate da 0 a $N - 1$. Ci sono M processi industriali, numerati da 0 a $M - 1$. Il processo i smaltisce 1 barile della sostanza A_i , costa 1 euro, e produce K_i barili di scarto, numerati da 0 a $K_i - 1$, di varie sostanze. Più precisamente, il barile di scarto j del processo i è un barile di sostanza $B_{i,j}$ (per $0 \leq j < K_i$).

Per alcuni processi i non ci sono barili di scarto, ovvero $K_i = 0$, e perciò possono essere usati per smaltire un barile della sostanza A_i , al costo di un euro, senza generare ulteriori sostanze da smaltire.

Inizialmente c'è esattamente un barile di sostanza 0 (Acido Olimpico), e nessun altro barile. Si può scegliere liberamente quali processi effettuare, quante volte, e in che ordine. Calcola il costo minimo necessario per smaltire tutti i barili.

Implementazione

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

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

Dovrai implementare la seguente funzione:

```
C++ | long long smaltisci(int N, int M, vector<int> A, vector<vector<int>> B);
```

- Gli interi N e M rappresentano il numero di sostanze e di processi industriali.
- L'array A , indicizzato da 0 a $M - 1$, contiene in posizione i la sostanza A_i smaltita dal processo i .
- L'array bidimensionale B , indicizzato da 0 a $M - 1$, contiene le sostanze dei barili di scarto di ogni processo. Più precisamente, per $0 \leq i < M$, $B[i]$ è un array di dimensione K_i che contiene gli interi $B_{i,0}, \dots, B_{i,K_i-1}$.
- La funzione dovrà restituire il costo minimo necessario per smaltire tutti i barili. È garantito che esista una strategia che permette di smaltire tutti i barili con costo al più 10^{15} euro.

Il grader chiamerà la funzione `smaltisci` 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 testare le vostre soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama le funzioni 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 .
- Riga $2 + i$ ($0 \leq i < M$): gli interi A_i e K_i , seguiti dai K_i interi $B_{i,0}, \dots, B_{i,K_i-1}$.

Il file di output è composto da un'unica riga, contenente il valore restituito dalla funzione `smaltisci`.

Assunzioni

- $1 \leq N \leq 100\,000$.
- $1 \leq M \leq 200\,000$.
- $0 \leq K_i \leq 100\,000$ per ogni $i = 0, \dots, M - 1$.
- Tutti gli M processi hanno al più 10^7 barili di scarto in totale (cioè $K_0 + K_1 + \dots + K_{M-1} \leq 10^7$).
- Ogni processo produce al massimo un barile di scarto per ogni sostanza.
- È sempre possibile smaltire tutti i barili con costo al più 10^{15} euro.

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 [6 punti]**: $N \leq 10$ e i numeri A_0, A_1, \dots, A_{M-1} sono distinti tra loro.
- **Subtask 3 [5 punti]**: $N \leq 10$.
- **Subtask 4 [10 punti]**: I numeri A_0, A_1, \dots, A_{M-1} sono distinti tra loro.
- **Subtask 5 [12 punti]**: $K_i \leq 1$ per ogni $0 \leq i < M$.
- **Subtask 6 [18 punti]**: Non esiste una sequenza di processi che permette di ottenere un barile della sostanza i a partire da un barile della sostanza i , per ogni $0 \leq i < M$.
- **Subtask 7 [30 punti]**: $N \leq 10\,000$.
- **Subtask 8 [19 punti]**: Nessuna limitazione aggiuntiva.

Esempi di input/output

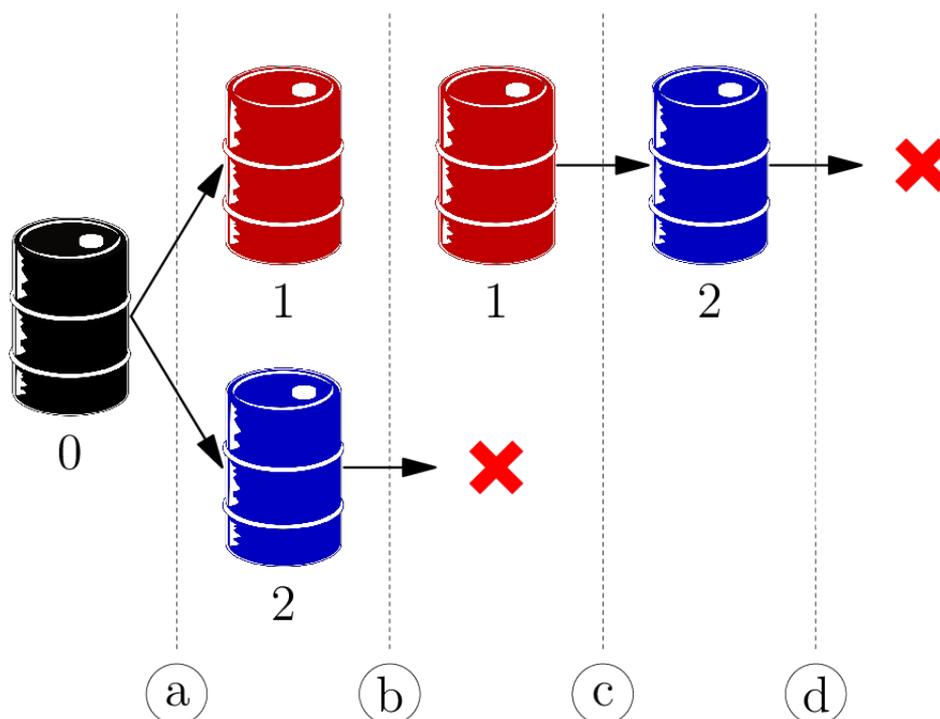
stdin	stdout
3 4 1 1 2 1 2 0 2 0 2 1 2 2 0	4
6 12 0 1 2 0 2 2 4 1 1 4 1 0 2 1 0 2 4 1 3 4 5 3 1 1 4 1 2 4 1 5 5 1 2 5 1 3 5 1 1	10

Spiegazione

Nel **primo caso di esempio** è possibile procedere nel seguente modo:

- si usa il processo 2 per smaltire un barile (l'unico inizialmente presente) di sostanza 0, producendo un barile di sostanza 1 e un barile di sostanza 2;
- si usa il processo 3 per smaltire un barile di sostanza 2, senza produrre nessuna sostanza di scarto;
- si usa il processo 0 per smaltire un barile di sostanza 1, producendo un barile di sostanza 2;
- si usa nuovamente il processo 3 per smaltire un barile di sostanza 2, e a questo punto tutte le sostanze sono state smaltite.

Il costo di questa strategia è 4 euro, ed è possibile mostrare che non ve ne sono di migliori.



Continua nella pagina successiva...

Nel **secondo caso di esempio** una strategia possibile è la seguente:

- (a) si usa il processo 0 per smaltire il primo barile (di sostanza 0) producendone uno di sostanza 2;
- (b) si usa il processo 5 per smaltire il barile di sostanza 2 producendone uno di ognuna delle sostanze 1, 3, 4 e 5;
- (c) si usa il processo 3 per smaltire il barile di sostanza 1, senza nessuno scarto;
- (d) si usa il processo 6 per smaltire il barile di sostanza 3, producendo un barile di sostanza 1;
- (e) si usa il processo 8 per smaltire il barile di sostanza 4, producendo un barile di sostanza 5;
- (f) si usa due volte il processo 11 per smaltire i due barili di sostanza 5, producendo due barili di sostanza 1;
- (g) si usa tre volte il processo 3 per smaltire i tre barili di sostanza 1, senza nessuno scarto.

Il costo complessivo di questa strategia è 10 euro.

