



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

robots

Italian — 1.0

Il fratello minore di Marita ha abbandonato i propri giocattoli sul pavimento del soggiorno! Per fortuna, Marita ha sviluppato dei robot specializzati nel raccogliere i giocattoli: ora ha solo bisogno del tuo aiuto per determinare quali robot dovranno mettere via quali giocattoli.

Ci sono T giocattoli, ciascuno con un peso intero $W[i]$ e una dimensione intera $S[i]$. I robot sono di due tipi: *weak* e *small*.

- Ci sono A robot weak. Ciascun robot weak ha un limite di peso $X[i]$, e può trasportare un qualunque giocattolo di peso strettamente minore di $X[i]$. La dimensione del giocattolo non ha importanza.
- Ci sono B robot small. Ciascun robot small ha un limite di dimensione $Y[i]$, e può trasportare un qualunque giocattolo di dimensione strettamente minore di $Y[i]$. Il peso del giocattolo non ha importanza.

Ciascun robot di Marita impiega un minuto per mettere via un giocattolo, ma robot diversi possono mettere via giocattoli diversi nello stesso momento.

Il tuo obiettivo è determinare se i robot di Marita possono riporre tutti i giocattoli, e in tal caso determinare il tempo minimo richiesto per farlo.

Esempi

Nel primo esempio, ci sono $A = 3$ robot weak con limiti di peso $X = [6, 2, 9]$ e $B = 2$ robot small con limiti di dimensione $Y = [4, 7]$ per $T = 10$ giocattoli caratterizzati come segue:

Numero del giocattolo	0	1	2	3	4	5	6	7	8	9
Peso	4	8	2	7	1	5	3	8	7	10
Dimensione	6	5	3	9	8	1	3	7	6	5

Il tempo minimo richiesto per mettere via tutti i giocattoli è di tre minuti:

	Robot weak 0	Robot weak 1	Robot weak 2	Robot small 0	Robot small 1
Min. 1	Giocattolo 0	Giocattolo 4	Giocattolo 1	Giocattolo 6	Giocattolo 2
Min. 2	Giocattolo 5		Giocattolo 3		Giocattolo 8
Min. 3			Giocattolo 7		Giocattolo 9

Nel secondo esempio, ci sono $A = 2$ robot weak con limiti di peso $X = [2, 5]$ e $B = 1$ robot small con limiti di dimensione $Y = [2]$ per $T = 3$ giocattoli caratterizzati come segue:

Numero del giocattolo	0	1	2
Peso	3	5	2
Dimensione	1	3	2

Nessun robot può raccogliere il giocattolo di peso 5 e dimensione 3, di conseguenza è impossibile che i robot riescano a mettere via tutti i giocattoli.

Implementazione

Devi sottoporre un file che implementi la funzione `putaway()` come segue:

Funzione: `putaway()`

C/C++

```
int putaway(int A, int B, int T,  
            int X[], int Y[], int W[], int S[]);
```

Pascal

```
function putaway(A, B, T : LongInt;  
                var X, Y, W, S : array of LongInt) : LongInt;
```

Descrizione

Questa funzione deve calcolare il tempo minimo in minuti necessario affinché i robot mettano via tutti i giocattoli, o restituire `-1` se non è possibile mettere via tutti i giocattoli.

Parametri

- A : Il numero di robot weak.
- B : Il numero di robot small.
- T : Il numero di giocattoli.
- X : Un array di interi di lunghezza A che specifica il limite di peso per ciascun robot weak.
- Y : Un array di interi di lunghezza B che specifica il limite di dimensione per ciascun robot small.
- W : Un array di interi di lunghezza T contenente il peso di ogni giocattolo.
- S : Un array di interi di lunghezza T contenente la dimensione di ogni giocattolo.
- *Restituisce*: Il numero di minuti più piccolo necessario per mettere via tutti i giocattoli, o `-1` se questa operazione non è possibile.

Sessioni di esempio

La seguente sessione descrive il primo esempio riportato sopra.

Parametro	Valore
A	3
B	2
T	10
X	[6, 2, 9]
Y	[4, 7]
W	[4, 8, 2, 7, 1, 5, 3, 8, 7, 10]
S	[6, 5, 3, 9, 8, 1, 3, 7, 6, 5]
Restituisce	3

La seguente sessione descrive il secondo esempio riportato sopra:

Parametro	Valore
A	2
B	1
T	3
X	[2, 5]
Y	[2]
W	[3, 5, 2]
S	[1, 3, 2]
Restituisce	-1

Limiti

- Tempo limite: 3 secondi
- Limite di memoria: 64 MiB
- $1 \leq T \leq 1\,000\,000$
- $0 \leq A, B \leq 50\,000$ e $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2\,000\,000\,000$

Subtask

Subtask	Punteggio	Limiti aggiuntivi
1	14	$T = 2$ e $A + B = 2$ (due giocattoli e due robot)
2	14	$B = 0$ (tutti i robot sono weak)
3	25	$T \leq 50$ e $A + B \leq 50$
4	37	$T \leq 10\,000$ e $A + B \leq 1\,000$
5	10	(Nessun limite aggiuntivo)

Testing

Il grader di esempio sul computer legge l'input dal file `robots.in`, che deve essere nel seguente formato:

- linea 1: `A B T`
- linea 2: `X[0] ... X[A-1]`
- linea 3: `Y[0] ... Y[B-1]`
- le successive T linee: `W[i] S[i]`

Il primo esempio può essere fornito nel seguente formato:

```
3 2 10
6 2 9
4 7
4 6
8 5
2 3
7 9
1 8
5 1
3 3
8 7
7 6
10 5
```

Se $A = 0$ oppure $B = 0$ allora le linee corrispondenti (linea 2 o linea 3) sono vuote.

Note relative al linguaggio

C/C++ Devi inserire `#include "robots.h"`.

Pascal Devi definire `unit Robots`. Tutti gli array sono numerati a partire da `0` (e non `1`).

Vedi i template di soluzione nel tuo computer per alcuni esempi.