



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

robots

Romanian — 1.0

Frăţiorul Maritei şi-a lăsat jucăriile peste tot pe podeaua din sufragerie. Din fericire, Marita a construit roboţi speciali pentru a strânge jucăriile. Ea are nevoie de ajutorul vostru pentru a determina care roboţi ar trebui să culeagă jucăriile.

Există T jucării, fiecare cu o greutate - număr întreg $W[i]$ şi o dimensiune - număr întreg $S[i]$. Roboţii sunt de două tipuri: *weak* şi *small*.

- Există A roboţi weak. Fiecare robot weak are o greutate limită $X[i]$, şi poate să culegă orice jucărie cu greutatea strict mai mică decât $X[i]$. Dimensiunea jucăriei nu contează.
- Există B roboţi small. Fiecare robot small are o dimensiune limită $Y[i]$, şi poate să culegă jucării cu dimensiunea strict mai mică decât $Y[i]$. Greutatea jucăriei nu contează.

Oricare robot al Maritei poate să culeagă exact o jucărie pe minut. Roboţi diferiţi pot căra jucării diferite în acelaşi timp.

Sarcina voastră este să determinaţi dacă roboţii Maritei pot culege toate jucăriile şi în caz afirmativ care este timpul minim în care acestea pot fi culese.

Exemple

Ca un prim exemplu, presupunem că există $A = 3$ roboţi weak cu greutatea limită $X = [6, 2, 9]$, $B = 2$ roboţi small cu dimensiunea limită $Y = [4, 7]$, şi $T = 10$ jucării după cum urmează:

Număr de jucării	0	1	2	3	4	5	6	7	8	9
Greutăţi	4	8	2	7	1	5	3	8	7	10
Dimensiuni	6	5	3	9	8	1	3	7	6	5

Cel mai scurt timp pentru a culege toate jucăriile este de trei minute:

	Robotul Weak 0	Robotul Weak 1	Robotul Weak 2	Robotul Small 0	Robotul Small 1
Primul minut	Jucăria 0	Jucăria 4	Jucăria 1	Jucăria 6	Jucăria 2
Al doilea minut	Jucăria 5		Jucăria 3		Jucăria 8
Al treilea minut			Jucăria 7		Jucăria 9

Ca un al doilea exemplu, presupunem că există $A = 2$ roboți weak cu greutatea limită $X = [2, 5]$, $B = 1$ robot small cu dimensiunea limită $Y = [2]$, și $T = 3$ jucării după cum urmează:

Număr jucărie	0	1	2
Greutăți	3	5	2
Dimensiuni	1	3	2

Niciun robot nu poate culege jucăria de greutate 5 și dimensiune 3 deci este imposibil ca roboții să culegă toate jucăriile.

Implementare

Trebuie să submițați un fișier care implementează funcția `putaway()` după cum urmează:

Funcția voastră: `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;
```

Descriere

Această funcție trebuie să returneze cel mai mic număr de minute necesar pentru ca roboții să culegă toate jucăriile, sau să returneze `-1` dacă acest lucru nu este posibil.

Parametri

- A : Numărul de roboți weak.
- B : Numărul de roboți small.
- T : Numărul de jucării.
- X : Un vector de întregi de lungime A conținând greutatea limită pentru fiecare robot weak.
- Y : Un vector de întregi de lungime B conținând dimensiunile limită pentru fiecare robot small.
- W : Un vector de întregi de lungime T conținând greutatea fiecărei jucării.

- S : Un vector de întregi de lungime T conținând dimensiunea fiecărei jucării.
- *Returnează*: Cel mai mic număr de minute în care se pot culege toate jucăriile sau -1 dacă jucăriile nu pot fi culese.

Exemple

Aici aveți descris primul exemplu:

Parametri	Valori
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]
Valoare returnată	3

Aici aveți descris al doilea exemplu

Parametri	Valori
A	2
B	1
T	3
X	[2, 5]
Y	[2]
W	[3, 5, 2]
S	[1, 3, 2]
Valoare returnată	-1

Restricții

- Limită de timp: 3 secunde
- Limită de memorie limit: 64 MiB
- $1 \leq T \leq 1,000,000$
- $0 \leq A, B \leq 50,000$ și $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2,000,000,000$

Subtask-uri

Subtask	Puncte	Restricții suplimentare
1	14	$T = 2$ și $A + B = 2$ (Exact două jucării și doi roboți)
2	14	$B = 0$ (toți roboții sunt weak)
3	25	$T \leq 50$ și $A + B \leq 50$
4	37	$T \leq 10,000$ și $A + B \leq 1,000$
5	10	(Niciuna)

Testare

Graderul de pe calculatorul vostru va citi datele de intrare din fișierul `robots.in`, care trebuie să aibă următorul format :

- linia 1: `A B T`
- linia 2: `X[0] ... X[A-1]`
- linia 3: `Y[0] ... Y[B-1]`
- următoarele `T` linii: `W[i] S[i]`

Astfel, primul exemplu prezentat ar trebui să primească datele de intrare în următorul format

```
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
```

Dacă `A = 0` sau `B = 0` linia corespunzătoare (linia 2 sau linia 3) trebuie să fie goală.

Note de limbaj

C/C++ Trebuie să faceți `#include "robots.h"`.

Pascal Trebuie să definiți `unit Robots`. Toți vectorii sunt indexați începând de la `0` (nu de la `1`).

Vedeți template-urile de soluții de pe calculatorul vostru pentru exemple.