



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

robots

magyar — 1.0

Marita speciális robotokat használ a szanaszét heverő játékok összegyűjtésére.

T játék van, ismerjük mindegyik $W[i]$ súlyát és $S[i]$ méretét. Kétféle robot van: könnyű és kicsi.

- A darab könnyű robot van. Az i . csak $X[i]$ -nél kisebb súlyú játékot tud mozgatni a helyére. A játék mérete nem számít.
- B darab kicsi robot van. Az i . csak $Y[i]$ -nél kisebb méretű játékot tud mozgatni a helyére. A játék súlya nem számít.

Mindegyik robot 1 perc alatt rak helyre 1 tárgyat. Egyszerre akárhány robot dolgozhat.

Feladatod megadni, hogy minimum mennyi idő alatt tudja az összes játékot helyre rakni vagy közölni, hogy nem megoldható a feladat.

Példák

Az első példában $A = 3$ könnyű robot van $X = [6, 2, 9]$ súlykorláttal, $B = 2$ kicsi robot van $Y = [4, 7]$ méretkorláttal, és $T = 10$ játék van:

Játék sorszám	0	1	2	3	4	5	6	7	8	9
Súly	4	8	2	7	1	5	3	8	7	10
Méret	6	5	3	9	8	1	3	7	6	5

Legkevesebb 3 perc alatt tudnak a robotok helyrerakni minden játékot:

	Könnnyű robot 0	Könnnyű robot 1	Könnnyű robot 2	Kicsi robot 0	Kicsi robot 1
Első perc	0. játék	4. játék	1. játék	6. játék	2. játék
Második perc	5. játék		3. játék		8. játék
Harmadik perc			7. játék		9. játék

A második példában $A = 2$ könnyű robot van $X = [2, 5]$ súlykorlátokkal, $B = 1$ kicsi robot $Y = [2]$ méretkorláttal, és $T = 3$ játék:

Játék sorszám	0	1	2
Súly	3	5	2
Méret	1	3	2

Egyetlen robot sem tudja elvinni az 5 súlyú és 3 méretű játékot, tehát a problémának nincs megoldása.

Megvalósítás

A `putaway()` függvényt kell beküldened:

A függvényed: `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;
```

Leírás

A függvény számítsa ki a legkisebb időt, ami alatt a robotok az összes játékot helyre rakják. `-1` legyen az értéke, ha nincs megoldás.

Paraméterek

- A : a könnyű robotok száma.
- B : a kicsi robotok száma.
- T : a játékok száma.
- X : A elemű tömb, a könnyű robotok súlykorlátját tartalmazza.
- Y : B elemű tömb, a kicsi robotok méretkorlátját tartalmazza.
- W : T elemű tömb, a játékok súlyát tartalmazza.
- S : T elemű tömb, a játékok méretét tartalmazza.
- *Visszatérési érték*: a legkisebb idő, ami alatt a robotok az összes játékot helyre rakják. `-1` legyen az értéke, ha nincs megoldás.

Példák

Az első példa:

Paraméter	Érték
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]
Visszatérési érték	3

Második példa:

Paraméter	Érték
A	2
B	1
T	3
X	[2, 5]
Y	[2]
W	[3, 5, 2]
S	[1, 3, 2]
Visszatérési érték	-1

Korlátok

- Időlimit: 3 másodperc
- Memória limit: 64 MiB
- $1 \leq T \leq 1,000,000$
- $0 \leq A, B \leq 50,000$ és $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2,000,000,000$

Részfeladatok

Részfeladat	Pontszámok	További korlátok
1	14	$T = 2$ és $A + B = 2$ (pontosan 2 játék és 2 robot)
2	14	$B = 0$ (minden robot könnyű)
3	25	$T \leq 50$ és $A + B \leq 50$
4	37	$T \leq 10,000$ és $A + B \leq 1,000$
5	10	(Nincs)

Gyakorlás

A mintaértékelő a `robots.in` fájlból olvassa a bemenetet, amelynek formája:

- 1. sor: `A B T`
- 2. sor: `X[0] ... X[A-1]`
- 3. sor: `Y[0] ... Y[B-1]`
- a következő T sor: `W[i] S[i]`

Az első példa fájlja így néz ki:

```
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
```

Ha $A = 0$ vagy $B = 0$, akkor a megfelelő sor (2. vagy 3.) üres.

Nyelvi előírások

C/C++ Importáld a `#include "robots.h"`-t.

Pascal Definiáld a `unit Robots`-t. A tömbök indexelése `0`-tól kezdődik (nem `1`-től).

Lásd a mintaprogramokat a gépeden!