



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

robots

Slovenščina — 1.0

Maritin mlajši brat je pustil igrače ležati povsod po sobi. K sreči je Marita izdelala robote za pospravljanje igrač. Sedaj bi rada, da ji pomagaš ugotoviti, kateri roboti naj pospravijo katere igrače.

V sobi je T igrač. Teža i -te igrače znaša $W[i]$, njena velikost pa $S[i]$. Vse teže in velikosti so cela števila. Roboti so dveh vrst: *šibki* in *mali*.

- Marita ima A šibkih robotov. Nosilnost i -tega šibkega robota znaša $X[i]$. To pomeni, da lahko nese poljubno igračo, ki tehta *strogo* manj kot $X[i]$. Velikost igrače ni pomembna.
- Marita ima B malih robotov. Velikostna omejitev i -tega malega robota znaša $Y[i]$. To pomeni, da lahko nese poljubno igračo z velikostjo, *strogo* manjšo od $Y[i]$. Teža igrače ni pomembna.

Vsak Maritin robot pospravi po eno igračo v eni minuti. Različni roboti lahko hkrati pospravljajo različne igrače.

Tvoja naloga je ugotoviti, ali lahko Maritini roboti pospravijo vse igrače. Če je to mogoče, moraš ugotoviti še najkrajši čas, v katerem lahko opravijo svoje delo.

Primeri

Za prvi primer predpostavimo, da ima Marita tri ($A = 3$) šibke robote z nosilnostmi $X = [6, 2, 9]$, dva ($B = 2$) mala robota z velikostnima omejitvama $Y = [4, 7]$ in $T = 10$ igrač s sledečimi podatki:

Številka igrače	0	1	2	3	4	5	6	7	8	9
Teža igrače	4	8	2	7	1	5	3	8	7	10
Velikost igrače	6	5	3	9	8	1	3	7	6	5

Najkrajši čas, v katerem lahko roboti pospravijo vse igrače, znaša 3 minute:

	Šibki robot 0	Šibki robot 1	Šibki robot 2	Mali robot 0	Mali robot 1
Prva minuta	igrača 0	igrača 4	igrača 1	igrača 6	igrača 2
Druga minuta	igrača 5		igrača 3		igrača 8
Tretja minuta			igrača 7		igrača 9

Za drugi primer predpostavimo, da ima Marita dva ($A = 2$) šibka robota z nosilnostima $X = [2, 5]$, enega ($B = 1$) malega robota z velikostno omejitvijo $Y = [2]$ in tri ($T = 3$) igrače s sledečimi podatki:

Številka igrače	0	1	2
Teža igrače	3	5	2
Velikost igrače	1	3	2

Nobeden od robotov ne more nesti igrače s številko 1 (s težo 5 in velikostjo 3), zato roboti ne morejo pospraviti vseh igrač.

Implementacija

Oddaj datoteko, v kateri je implementirana funkcija `putaway()` po sledečih navodilih:

Tvoja funkcija: `putaway()`

Opis

Ta funkcija naj izračuna najmanjše število minut, v katerih lahko roboti pospravijo vse igrače. Če tega ne morejo storiti, naj funkcija vrne `-1`.

Parametri

- `A`: Število šibkih robotov.
- `B`: Število malih robotov.
- `T`: Število igrač.
- `X`: Polje dolžine `A`, ki vsebuje nosilnosti posameznih šibkih robotov.
- `Y`: Polje dolžine `B`, ki vsebuje velikostne omejitve posameznih malih robotov.
- `W`: Polje dolžine `T`, ki vsebuje teže posameznih igrač.
- `S`: Polje dolžine `T`, ki vsebuje velikosti posameznih igrač.
- *Vrača*: Najmanjše število minut, v katerih lahko roboti pospravijo vse igrače, oziroma `-1`, če tega ne morejo storiti.

Vzorčni klic funkcije `putaway()`

Sledeči klic funkcije `putaway` ustreza prvemu primeru:

Parameter	Vrednost
<code>A</code>	<code>3</code>
<code>B</code>	<code>2</code>
<code>T</code>	<code>10</code>
<code>X</code>	<code>[6, 2, 9]</code>
<code>Y</code>	<code>[4, 7]</code>
<code>W</code>	<code>[4, 8, 2, 7, 1, 5, 3, 8, 7, 10]</code>
<code>S</code>	<code>[6, 5, 3, 9, 8, 1, 3, 7, 6, 5]</code>
Vrne	<code>3</code>

Sledeči klic funkcije `putaway` ustreza drugemu primeru:

Parameter	Vrednost
A	2
B	1
T	3
X	[2, 5]
Y	[2]
W	[3, 5, 2]
S	[1, 3, 2]
Vrne	-1

Omejitve

- Časovna omejitev: 3 sekunde
- Prostorska omejitev: 64 MiB
- $1 \leq T \leq 1\,000\,000$
- $0 \leq A, B \leq 50\,000$ in $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2\,000\,000\,000$

Podnaloge

Podnaloga	Točke	Dodatne vhodne omejitve
1	14	$T = 2$ in $A + B = 2$ (natanko dve igrači in dva robota)
2	14	$B = 0$ (vsi roboti so šibki)
3	25	$T \leq 50$ in $A + B \leq 50$
4	37	$T \leq 10\,000$ in $A + B \leq 1000$
5	10	(Brez)

Preizkušanje

Vzorčni ocenjevalnik na tvojem računalniku bere vhod iz datoteke `robots.in`, ki mora biti sledeče oblike:

- vrstica 1: `A B T`
- vrstica 2: `X[0] ... X[A-1]`
- vrstica 3: `Y[0] ... Y[B-1]`
- naslednjih `T` vrstic: `W[i] S[i]`

Prvemu primeru ustreza datoteka sledeče oblike:

```
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
```

Če je `A = 0` ali `B = 0`, potem je pripadajoča vrstica (vrstica 2 ali vrstica 3) prazna.

Jezikovne opombe

C/C++ Potrebujš `#include "robots.h"`.

Pascal Definiraj `unit Robots`. Oštevilčenje vseh polj se prične z `0` in ne z `1`.

Za primere si oglej predloge rešitev na tvojem računalniku.