



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

# robots

Latvian — 1.0

Maritas mazajam brālim patīk izmētāt rotaļlietas pa dzīvojamās istabas grīdu un tās nesavākt. Par laimi, rotaļlietu savākšanai Marita ir izveidojusi speciālus robotus. Tomēr viņai ir nepieciešama palīdzība, lai noteiktu, kuras rotaļlietas kuram robotam jāsavāc.

Pavisam ir  $T$  rotaļlietas, katru no kurām raksturo tās svars (vesels skaitlis  $W[i]$ ) un izmērs (vesels skaitlis  $S[i]$ ). Ir divu veidu roboti: *vājie* un *mazie*.

- Pavisam ir  $A$  vājie roboti. Katram vājam robotam ir paceļamā svara ierobežojums  $X[i]$  un tas var pārvietot tikai tādas rotaļlietas, kuru svars ir stingri mazāks par  $X[i]$ . Rotaļlietas izmēram šajā gadījumā nav nozīmes.
- Pavisam ir  $B$  mazie roboti. Katram mazajam robotam ir pārvietojamā priekšmeta izmēra ierobežojums  $Y[i]$  un tas var pārvietot tikai tādas rotaļlietas, kuru izmērs ir stingri mazāks par  $Y[i]$ . Rotaļlietas svaram šajā gadījumā nav nozīmes.

Katram no Maritas robotiem vienas rotaļlietas savākšanai nepieciešama viena minūte. Vienā un tajā pašā laikā dažādi roboti var nodarboties ar dažādu rotaļlietu savākšanu.

Jūsu uzdevums ir uzrakstīt programmu, kas nosaka, vai Maritas roboti var savākt visas rotaļlietas un, ja var, tad kāds mazākais laiks tam nepieciešams.

## Piemēri

Pirmajā piemērā pieņemsim, ka ir  $A = 3$  vājie roboti ar pārvietojamu priekšmetu svara ierobežojumiem attiecīgi  $X = [6, 2, 9]$ ,  $B = 2$  mazie roboti ar pārvietojamu priekšmetu izmēru ierobežojumiem attiecīgi  $Y = [4, 7]$ , un  $T = 10$  rotaļlietas ar šādiem parametriem:

Rotaļlietas numurs	0	1	2	3	4	5	6	7	8	9
Svars	4	8	2	7	1	5	3	8	7	10
Izmērs	6	5	3	9	8	1	3	7	6	5

Mazākais laiks, kādā iespējams aizvēkt visas rotaļlietas, ir trīs minūtes. Tabulā katras rotaļlietas numuram norādīta minūte, kurā tā tiks savākta, kā arī kurš no robotiem to paveiks:

	Vājais[0]	Vājais[1]	Vājais[2]	Mazais[0]	Mazais[1]
1.minūte	0	4	1	6	2
2.minūte	5		3		8
3.minūte			7		9

Otrajā piemērā pieņemsim, ka ir  $A=2$  vājie roboti ar pārvietojamo priekšmetu svara ierobežojumiem attiecīgi  $X=[2, 5]$ ,  $B=1$  mazais robots ar pārvietojamo priekšmetu izmēra ierobežojumu  $Y=[2]$ , un  $T=3$  rotaļlietas ar šādiem parametriem:

Rotaļlietas numurs	0	1	2
Svars	3	5	2
Izmērs	1	3	2

Neviens robots nevar pārvietot rotaļlietu Nr.1, kuras svars ir 5 un izmērs - 3, tāpēc roboti nevar savākt visas rotaļlietas.

## Implementācija

Jums jāiesūta fails, kas implementā šādu funkciju `putaway()`:

### Jūsu funkcija: `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;
```

### Apraksts

Šai funkcijai jāaprēķina un jāizvada mazākais visu rotaļlietu savākšanai nepieciešamais laiks minūtēs, vai arī jāizvada `-1`, ja rotaļlietas savākt nav iespējams.

### Parametri

- $A$  : Vājo robotu skaits.
- $B$  : Mazo robotu skaits.
- $T$  : Rotaļlietu skaits.
- $X$  : Veselu skaitļu masīvs garumā  $A$ , kas satur visu vājo robotu pārvietojamo priekšmetu svara ierobežojumus.

- $Y$ : Veselu skaitļu masīvs garumā  $B$ , kas satur visu mazo robotu pārvietojamo priekšmetu izmēru ierobežojumus.
- $W$ : Veselu skaitļu masīvs garumā  $T$ , kas satur visu rotaļlietu svarus.
- $S$ : Veselu skaitļu masīvs garumā  $T$ , kas satur visu rotaļlietu izmērus.
- *Rezultāts*: Mazākais visu rotaļlietu savākšanai nepieciešamais minūšu skaits vai  $-1$ , ja visas rotaļlietas savākt nav iespējams.

## Piemēra sesija

Pirmo piemēru apraksta šāda sesija:

Parametrs	Vērtība
<b>A</b>	3
<b>B</b>	2
<b>T</b>	10
<b>X</b>	[6, 2, 9]
<b>Y</b>	[4, 7]
<b>W</b>	[4, 8, 2, 7, 1, 5, 3, 8, 7, 10]
<b>S</b>	[6, 5, 3, 9, 8, 1, 3, 7, 6, 5]
<b>Rezultāts</b>	3

Otro piemēru apraksta šāda sesija:

Parameters	Vērtība
<b>A</b>	2
<b>B</b>	1
<b>T</b>	3
<b>X</b>	[2, 5]
<b>Y</b>	[2]
<b>W</b>	[3, 5, 2]
<b>S</b>	[1, 3, 2]
<b>Rezultāts</b>	-1

## Ierobežojumi

- Laika ierobežojums: 3 sekundes
- Atmiņas ierobežojums: 64 MiB

- $1 \leq T \leq 1,000,000$
- $0 \leq A, B \leq 50,000$  un  $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2,000,000,000$

## Apakšuzdevumi

Apakšuzdevums	Punkti	Papildus ievaddatu ierobežojumi
1	14	$T = 2$ un $A + B = 2$ (tieši divas rotaļlietas un divi roboti)
2	14	$B = 0$ (ir tikai vājie roboti)
3	25	$T \leq 50$ un $A + B \leq 50$
4	37	$T \leq 10,000$ un $A + B \leq 1,000$
5	10	<i>Nav papildus ierobežojumu</i>

## Eksperimentēšana

Jūsu datorā esošais piemēru vērtētājs ielasīs ievaddatus no `robots.in` faila, šādā formātā:

- 1. rinda: `A B T`
- 2. rinda: `X[0] ... X[A-1]`
- 3. rinda: `Y[0] ... Y[B-1]`
- nākamās `T` rindas: `W[i] S[i]`

Augstāk esošais piemērs jāapraksta šādā veidā:

```
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
```

Ja `A = 0` vai `B = 0`, tad atbilstošai rindai (2. rindai vai 3. rindai) jābūt tukšai.

---

## Piezīmes par valodām

C/C++ Jums jāiekļauj `#include "robots.h"`.

Pascal Jums jādefinē `unit Robots`. Visi masīvi tiek numurēti sākot no `0` (nevis `1`).

Iepazīstieties ar risinājumu piemēru šabloniem uz jūsu datora.