



## International Olympiad in Informatics

2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

# robots

Nederlands (België) —

1.0

---

Al het speelgoed van Marita's kleine broertje ligt op de vloer! Gelukkig heeft Marita speciale robots gebouwd om het speelgoed op te ruimen. Ze heeft jouw hulp nodig om te bepalen welke robot welk speelgoed moet oppakken.

Er zijn  $T$  speeltjes die elk een gewicht (integer)  $W[i]$  en omvang (integer)  $S[i]$  hebben.

Er zijn twee soorten robots: *zwak* en *klein*.

- Er zijn  $A$  zwakke robots. Elke zwakke robot heeft een gewichtslimiet van  $X[i]$ , en kan één speeltje tillen zolang het gewicht ervan strikt kleiner is dan  $X[i]$ . De grootte van het speeltje maakt niet uit.
- Er zijn  $B$  kleine robots. Elke kleine robot heeft een omvangslimiet van  $Y[i]$ , en kan één speeltje tillen zolang de omvang ervan strikt kleiner is dan  $Y[i]$ . Het gewicht van het speeltje maakt niet uit.

Het kost een robot 1 minuut om een speeltje op te ruimen. Verschillende robots kunnen tegelijkertijd een verschillend speeltje opruimen.

Bepaal of het mogelijk is voor de robots om alle speelgoed op te ruimen, en zo ja, hoeveel tijd ze hier minimaal voor nodig hebben.

---

## Voorbeelden

Een eerste voorbeeld:

Stel dat er  $A = 3$  zwakke robots zijn met gewichtslimieten:  $X = [6, 2, 9]$ . Stel dat er  $B = 2$  kleine robots zijn met omvangslimieten:  $Y = [4, 7]$  zijn, en dat er verder  $T = 10$  speeltjes zijn.

Speeltje nr	0	1	2	3	4	5	6	7	8	9
Gewicht	4	8	2	7	1	5	3	8	7	10
Omvang	6	5	3	9	8	1	3	7	6	5

Het kost minimaal drie minuten om alle speeltjes op te ruimen:

	Zwakke robot 0	Zwakke robot 1	Zwakke robot 2	Kleine robot 0	Kleine robot 1
Eerste minuut	Speeltje 0	Speeltje 4	Speeltje 1	Speeltje 6	Speeltje 2
Tweede minuut	Speeltje 5		Speeltje 3		Speeltje 8
Derde minuut			Speeltje 7		Speeltje 9

Een tweede voorbeeld:

Stel dat er  $A = 2$  zwakke robots zijn met gewichtslimieten:  $X = [2, 5]$ . Stel dat er  $B = 1$  kleine robot is met omvangslimiet  $Y = [2]$ , en dat er verder  $T = 3$  speeltjes zijn.

Speeltje nr	0	1	2
Gewicht	3	5	2
Omvang	1	3	2

Geen enkele robot kan het speeltje met gewicht 5 en omvang 3 oppakken. Het is dus onmogelijk voor de robots om alle speeltjes op te ruimen.

---

## Implementatie

Je moet een bestand indienen dat de functie `putaway()` als volgt implementeert:

### Jouw Functie: `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;
```

### Beschrijving

Deze functie moet het kleinste aantal minuten bepalen waarin de robots alle speeltjes kunnen opruimen. Als dit niet mogelijk is, geef dan `-1` terug als resultaat.

### Parameters

- `A` : Het aantal zwakke robots.
- `B` : Het aantal kleine robots.
- `T` : Het aantal speeltjes.
- `X` : Een array van lengte `A` die integers bevat die voor elke zwakke robot de gewichtslimiet aangeven.
- `Y` : Een array van lengte `B` die integers bevat die voor elke kleine robot de omvangslimiet aangeven.
- `W` : Een array van lengte `T` die integers bevat die het gewicht van elk speeltje aangeven.
- `S` : Een array van lengte `T` die integers bevat die de omvang van elke speeltje aangeven.
- *Resultaat*: Het kleinste aantal minuten dat nodig is om alle speeltjes op te ruimen, of `-1` als dit niet mogelijk is.

---

## Voorbeeldscenario

Het volgende scenario beschrijft het eerste bovenstaande voorbeeld:

Parameter	Waarde
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]
Resultaat	3

Het volgende scenario beschrijft het tweede bovenstaande voorbeeld:

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

---

## Beperkingen

- Tijdslimiet: 3 seconden
- Geheugenlimiet: 64 MiB
- $1 \leq T \leq 1.000.000$
- $0 \leq A, B \leq 50.000$  en  $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2.000.000.000$

## Subtaken

Subtaak	Punten	Aanvullende beperkingen
1	14	$T = 2$ and $A + B = 2$ (exact twee speeltjes en twee robots)
2	14	$B = 0$ (alle robots zijn zwak)
3	25	$T \leq 50$ en $A + B \leq 50$
4	37	$T \leq 10,000$ en $A + B \leq 1,000$
5	10	(Geen beperkingen)

## Experimenteren

De voorbeeldgrader op jouw computer leest invoer uit het bestand `robots.in`, dat in het volgende formaat is opgesteld:

- regel 1: `A B T`
- regel 2: `X[0] ... X[A-1]`
- regel 3: `Y[0] ... Y[B-1]`
- de volgende `T` regels: `W[i] S[i]`

Het eerste bovenstaande voorbeeld is dus in het volgende formaat opgesteld:

```
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
```

Als `A = 0` of `B = 0` dan moet de bijhorende regel (regel 2 of regel 3) leeg zijn.

---

## Taalspecifieke opmerkingen

C/C++ Je moet `#include "robots.h"` gebruiken.

Pascal Je moet `unit Robots` definiëren. Alle arrays zijn genummerd vanaf `0` (niet `1`).

Bekijk de modeloplossing op je computer als voorbeeld.