



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

# robots

Spanish — 1.0

El hermano pequeño de Marita tiene el suelo de la habitación lleno de juguetes. Por suerte, Marita ha desarrollado unos robots que le ayudarán a recogerlos, pero ahora necesita tu ayuda para determinar, por cada uno de los juguetes, qué robot lo debe recoger.

Hay  $T$  juguetes, cada uno con un peso  $W[i]$  y un tamaño  $S[i]$ . Tanto el peso como el tamaño son números enteros. Los robots pueden ser de dos tipos: *débiles* o *pequeños*.

- Hay  $A$  robots débiles. Cada uno de ellos tiene un peso límite  $X[i]$ , y puede transportar cualquier juguete que tenga un peso estrictamente menor que  $X[i]$ . El tamaño del juguete no importa.
- Hay  $B$  robots pequeños. Cada uno de ellos tiene un tamaño límite  $Y[i]$ , y puede cargar cualquier juguete de tamaño estrictamente menor que  $Y[i]$ . El peso del juguete no importa.

Cada uno de los robots de Marita tarda un minuto en guardar cada juguete en su sitio. Dos robots diferentes pueden guardar simultáneamente dos juguetes diferentes.

Tu trabajo consiste en determinar si los robots de Marita pueden guardar todos los juguetes y, en caso afirmativo, cuál sería el mínimo tiempo en el que podrían hacerlo.

## Ejemplos

Como primer ejemplo, supón que hay  $A = 3$  robots débiles con límite de peso  $X = [6, 2, 9]$ ,  $B = 2$  robots pequeños con límite de tamaño  $Y = [4, 7]$ , y  $T = 10$  juguetes con el peso y tamaño que se describe a continuación:

Número de juguete	0	1	2	3	4	5	6	7	8	9
Peso	4	8	2	7	1	5	3	8	7	10
Tamaño	6	5	3	9	8	1	3	7	6	5

El tiempo mínimo necesario para guardar todos los juguetes es de tres minutos:

	<b>Robot débil 0</b>	<b>Robot débil 1</b>	<b>Robot débil 2</b>	<b>Robot pequeño 0</b>	<b>Robot pequeño 1</b>
<b>Primer minuto</b>	Juguete 0	Juguete 4	Juguete 1	Juguete 6	Juguete 2
<b>Segundo minuto</b>	Juguete 5		Juguete 3		Juguete 8
<b>Tercer minuto</b>			Juguete 7		Juguete 9

Como segundo ejemplo, supón que  $A = 2$  robots débiles con límites de peso  $X = [2, 5]$ ,  $B = 1$  robot pequeño con límite de tamaño  $Y = [2]$ , y  $T = 3$  con el peso y tamaño que se describe a continuación:

Número de juguete	0	1	2
Peso	3	5	2
Tamaño	1	3	2

No hay ningún robot que pueda recoger un juguete de peso 5 y tamaño 3, por lo que es imposible que los robots puedan guardar todos los juguetes.

## Implementación

Se te pide que envíes un archivo que implemente la función `putaway()` como se describe a continuación:

### Tu Función: `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;
```

### Descripción

Esta función debe calcular el mínimo número de minutos requeridos para que los robots guarden todos los juguetes, o bien devolver `-1` si esto no es posible.

### Parámetros

- $A$ : Número de robots débiles.
- $B$ : Número de robots pequeños.
- $T$ : Número de juguetes.
- $X$ : Un array de tamaño  $A$  que contiene enteros que especifican el peso límite para cada robot débil.
- $Y$ : Un array de tamaño  $B$  que contiene enteros que especifican el tamaño límite de cada robot pequeño.
- $W$ : Un array de tamaño  $T$  que contiene enteros que dan el peso de cada juguete.
- $S$ : Un array de tamaño  $T$  que contiene enteros que dan el tamaño de cada juguete.

- *Returns*: El mínimo número de minutos requeridos para guardar todos los juguetes en su sitio, o  si esto no es posible.

## Secuencia de ejemplo

La secuencia siguiente describe el primero de los ejemplos:

Parámetros	Valor
<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>Returns</b>	3

La siguiente secuencia describe el segundo de los ejemplos:

Parámetro	Valor
<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>Returns</b>	-1

## Restricciones

- Tiempo límite: 3 segundos
- Límite de memoria: 64 MiB
- $1 \leq T \leq 1,000,000$
- $0 \leq A, B \leq 50,000$  y  $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2,000,000,000$

## Subtareas

Subtarea	Puntos	Restricciones adicionales de entrada
1	14	$T = 2$ y $A + B = 2$ (Exactamente 2 robots)
2	14	$B = 0$ (Todos los robots son débiles)
3	25	$T \leq 50$ y $A + B \leq 50$
4	37	$T \leq 10,000$ y $A + B \leq 1,000$
5	10	(Ninguna)

## Experimentación

El corrector de ejemplo que tienes disponible en tu ordenador lee la entrada desde el archivo `robots.in`, que debe estar en el formato siguiente:

- línea 1: `A B T`
- línea 2: `X[0] ... X[A-1]`
- línea 3: `Y[0] ... Y[B-1]`
- las siguientes `T` líneas: `W[i] S[i]`

Por ejemplo, el ejemplo anterior se daría en el formato siguiente:

```
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
```

Si `A = 0` o `B = 0` entonces las líneas correspondientes (línea 2 o línea 3) deberán estar vacías.

## Apuntes del Lenguaje

C/C++ Debes incluir la línea `#include "robots.h"`.

Pascal Tienes que definir `unit Robots`. Todos los arrays se numeran desde `0` (no `1`).

Échale un vistazo a los moldes que hay en tu ordenador para ver ejemplos.