



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

# robots

Spanish — 1.0

El pequeño hermano de Marita ha dejado juguetes regados por toda la sala de estar. Afortunadamente Marita construyó robots que recogen juguetes. Ella necesita tu ayuda para determinar qué robots deben recoger qué juguetes.

En total hay  $T$  juguetes, cada juguete pesa una cantidad entera  $W[i]$  y mide una cantidad entera  $S[i]$ . Los robots son de dos tipos: *débiles* y *pequeños*.

- En total hay  $A$  robots débiles. Cada robot tiene un peso límite  $X[i]$ , es decir, puede cargar cualquier juguete cuyo peso sea estrictamente menor que  $X[i]$ . El tamaño del juguete no importa.
- En total hay  $B$  robots pequeños. Cada robot pequeño tiene un tamaño límite  $Y[i]$ , es decir, puede cargar cualquier juguete cuyo tamaño sea estrictamente menor que  $Y[i]$ . El peso del juguete no importa.

Cada robot de Marita tarda exactamente un minuto en recoger un juguete. Diferentes robots pueden recoger distintos juguetes al mismo tiempo.

Tu tarea es determinar si los robots de Marita pueden recoger todos los juguetes y en caso de que puedan, la mínima cantidad de tiempo para hacerlo.

## Ejemplos

Como primer ejemplo, supongamos que hay  $A = 3$  robots débiles con pesos límite  $X = [6, 2, 9]$ ,  $B = 2$  robots pequeños con tamaños límite  $Y = [4, 7]$ , y  $T = 10$  juguetes de la siguiente manera.

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 mínimo tiempo para recoger 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, supongamos que hay  $A = 2$  robots "débiles" con peso límite  $X = [2, 5]$ ,  $B = 1$  robots "pequeños" con peso límite  $Y = [2]$  y  $T = 3$  juguetes de la siguiente manera.

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

Ningún robot puede recoger el juguete de peso 5 y tamaño 3, por lo tanto es imposible para los robots recoger todos los juguetes.

## Implementación

Debes subir un archivo con la función `putaway()` implementada de la siguiente manera:

### 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 la mínima cantidad de minutos para que los robots puedan recoger todos los juguetes, en caso de que no sea posible, debe regresar -1.

### Parámetros

- $A$  : El número de robots "débiles".
- $B$  : El número de robots "pequeños".
- $T$  : El número de juguetes.
- $X$  : Un arreglo de enteros de longitud  $A$  que contiene el peso límite que puede cargar cada robot.
- $Y$  : Un arreglo de enteros de longitud  $B$  que contiene el tamaño límite que puede cargar cada robot.
- $W$  : Un arreglo de enteros de longitud  $T$  que contiene el peso de cada juguete.
- $S$  : Un arreglo de enteros de longitud  $T$  que contiene el tamaño de cada juguete.

- *Returns*: El mínimo número de minutos requerido para recoger todos los juguetes o  $-1$  en caso de que no sea posible.

## Sesión de ejemplo

La siguiente sesión describe el primer ejemplo mencionado anteriormente:

Parámetro	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>Regresa</b>	3

La siguiente sesión describe el segundo ejemplo mencionado anteriormente:

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>Regresa</b>	-1

## Restricciones

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

## Sub-problemas

Subtarea	Puntos	Restricciones adicionales
1	14	$T = 2$ and $A + B = 2$ (exactamente dos juguetes y dos 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 evaluador de ejemplo de tu computadora leerá el archivo de entrada `robots.in`, el cual debe tener el siguiente formato:

- 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 primer ejemplo debe tener el siguiente formato:

```
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 la línea correspondiente (línea 2 o línea 3) debe ser 0.

## Notas de lenguaje

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

Pascal Debes definir `unit Robots`. Todos los arreglos están numerados desde `0` (no `1`).

Revisa los templates de tu computadora para más ejemplos.