



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

robots

Español-Colombia

— 1.0

¡El hermanito de Marita dejó todos sus juguetes botados en la sala! Afortunadamente, Marita construyó robots especiales para recoger los juguetes. Ella necesita su ayuda para determinar cuáles robots deben recoger cuáles juguetes.

Hay T juguetes, cada uno con un peso $W[i]$ y un volumen $S[i]$. Hay dos tipos de robots: *débiles y pequeños*.

- Hay A robots débiles. Cada robot débil tiene un límite de carga $X[i]$; puede cargar cualquier juguete que pese estrictamente menos que $X[i]$. El tamaño del juguete no importa.
- Hay B robots pequeños. Cada robot pequeño tiene un límite de volumen $Y[i]$; puede cargar cualquier juguete que tenga estrictamente volumen menor que $Y[i]$. El peso del juguete no importa.

Cada uno de los robots de Marita se demora un minuto en guardar cada juguete. Dos robots diferentes pueden guardar dos juguetes diferentes al mismo tiempo.

Su tarea es determinar si los robots de Marita pueden guardar todos los juguetes y en caso de ser posible, determinar el tiempo más corto en que pueden hacerlo.

Ejemplos

Primero, suponga que hay $A = 3$ robots débiles con límites de carga $X = [6, 2, 9]$, $B = 2$ robots pequeños con límites de volumen $Y = [4, 7]$ y $T = 10$ juguetes:

Número de juguetes	0	1	2	3	4	5	6	7	8	9
Peso	4	8	2	7	1	5	3	8	7	10
Volumen	6	5	3	9	8	1	3	7	6	5

El tiempo mínimo en el que se pueden guardar los juguetes es de tres minutos:

	Robot débil 0	Robot débil 1	Robot débil 2	Robot pequeño 0	Robot pequeño 1
Primer minuto	Juguete 0	Juguete 4	Juguete 1	Juguete 6	Juguete 2
Segundo minuto	Juguete 5		Juguete 3		Juguete 8
Tercer Minuto			Juguete 7		Juguete 9

Segundo, suponga que hay $A = 2$ robots débiles con límites de carga $X = [2, 5]$, $B = 1$ robot pequeño con límite de volumen $Y = [2]$ y $T = 3$ juguetes:

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

Ningún robot puede recoger el juguete de peso 5 y volumen 3, entonces es imposible que los robots puedan recoger los juguetes.

Implementación

Usted debe enviar un programa que implemente la función `putaway()` de la siguiente forma:

Su 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 que se necesitan para guardar todos los juguetes, o debe retornar `-1` si no es posible.

Parámetros

- `A`: El número de robots débiles.
- `B`: El número de robots pequeños.
- `T`: El número de juguetes.
- `X`: Un arreglo de `A` enteros que representa el límite de carga de cada robot débil.

- **Y**: Un arreglo de **B** enteros que representa el límite de volumen de cada robot pequeño.
- **W**: Un arreglo de **T** enteros que representa el peso de cada juguete.
- **S**: Un arreglo de **T** enteros que representa el volumen de cada juguete.
- *Returns*: El mínimo número de minutos requeridos para guardar todos los juguetes o **-1** si esto no es posible.

Sesión de Ejemplo

La siguiente sesión muestra el primer ejemplo explicado anteriormente:

Parámetro	Valor
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]
Returns	3

La siguiente sesión muestra el segundo ejemplo explicado anteriormente:

Parámetro	Valor
A	2
B	1
T	3
X	[2, 5]
Y	[2]
W	[3, 5, 2]
S	[1, 3, 2]
Returns	-1

Restricciones

- Límite de tiempo: 3 segundos.
- Límite de memoria: 64 MB.

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

Subtareas

Subtarea	Puntos	Restricciones
1	14	$T = 2$ y $A + B = 2$ (Hay 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 calificador en su máquina leerá de el archivo `robots.in`, que va en 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]`

Así, el primer ejemplo iría en 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 (la línea 2 ó 3) debe estar vacía.

Notas de Lenguaje

C/C++ Debe incluir la siguiente línea: `#include "robots.h"`.

Pascal Debe definir `unit Robots`. Todos los arreglos comienzan en `0` (no `1`).

Para más información vea los esqueletos de solución en su máquina.