



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

robots

Russian — 1.0

Маленький брат Мариты разбросал игрушки по всей комнате! Как нельзя кстати, Марита разработала специальных роботов для уборки комнат. Она просит вас помочь определить, какие игрушки должен убирать каждый из роботов.

Всего разбросано T игрушек, каждая игрушка имеет целочисленный вес $W[i]$ и целочисленный размер $S[i]$. Роботы бывают двух типов: слабые и маленькие.

- Всего есть A слабых роботов. Каждый слабый робот имеет некоторое ограничение по весу $X[i]$, то есть, он может перемещать игрушки веса строго меньше чем $X[i]$. Размер игрушки при этом не имеет значения.
- Всего есть B маленьких роботов. Каждый маленький робот имеет некоторое ограничение по размеру $Y[i]$, то есть, он может перемещать игрушки размера строго меньше чем $Y[i]$. Вес игрушки при этом не имеет значения.

Процесс уборки каждой игрушки для любого робота занимает одну минуту. Каждый робот может убирать только одну игрушку в минуту. Разные роботы могут одновременно убирать различные игрушки.

Ваша задача состоит в том, чтобы определить, возможно ли убрать все игрушки, и если да, вычислить минимальное время, за которое это возможно сделать.

Примеры

Предположим, что у Мариты есть $A = 3$ слабых робота с ограничениями по весу $X = [6, 2, 9]$, $B = 2$ маленьких робота с ограничениями по размеру $Y = [4, 7]$ и $T = 10$ игрушек с параметрами, описанными ниже:

Номер игрушки	0	1	2	3	4	5	6	7	8	9
Вес	4	8	2	7	1	5	3	8	7	10
Размер	6	5	3	9	8	1	3	7	6	5

Минимальное время, за которое можно убрать все игрушки, — три минуты. Один из оптимальных способов распределения работы между роботами указан в таблице ниже:

	Слабый робот 0	Слабый робот 1	Слабый робот 2	Маленький робот 0	Маленький робот 1
Первая минута	Игрушка 0	Игрушка 4	Игрушка 1	Игрушка 6	Игрушка 2
Вторая минута	Игрушка 5		Игрушка 3		Игрушка 8
Третья минута			Игрушка 7		Игрушка 9

Рассмотрим другой пример. Предположим, что у Мариты есть $A = 2$ слабых робота с ограничениями по весу $X = [2, 5]$, $B = 1$ маленький робот с ограничениями по размеру $Y = [2]$, а также $T = 3$ игрушки с параметрами, описанными ниже:

Номер игрушки	0	1	2
Вес	3	5	2
Размер	1	3	2

Ни один из роботов не может убрать игрушку весом 5 и размером 3, поэтому не существует способа убрать все игрушки.

Детали реализации

Ваше решение должно реализовывать функцию `putaway()`, которая должна быть описана следующим образом:

Ваша функция: `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;
```

Описание

Эта функция должна вычислять минимально возможное время в минутах, которое требуется роботам, чтобы убрать все игрушки, или возвращать `-1`, если не существует ни одного способа убрать все игрушки.

Параметры

- `A` : количество слабых роботов.
- `B` : количество маленьких роботов.
- `T` : количество игрушек.
- `X` : массив длины `A`, который содержит целые числа, каждое из которых описывает ограничение по весу для соответствующего слабого робота.
- `Y` : массив длины `B`, который содержит целые числа, каждое из которых описывает ограничение по размеру для соответствующего маленького робота.
- `W` : массив длины `T`, который содержит целые числа, каждое из которых описывает вес соответствующей игрушки.
- `S` : массив длины `T`, который содержит целые числа, каждое из которых описывает размер соответствующей игрушки.
- *Возвращаемое значение*: минимально возможное время, необходимое роботам для того, чтобы убрать все игрушки, или `-1`, если убрать игрушки невозможно.

Примеры

Следующий набор данных описывает первый пример, приведенный выше в условии задачи:

Параметр	Значение
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]
Возвращаемое значение	3

Следующий набор данных описывает второй пример, приведенный выше в условии задачи:

Параметр	Значение
A	2
B	1
T	3
X	[2, 5]
Y	[2]
W	[3, 5, 2]
S	[1, 3, 2]
Возвращаемое значение	-1

Ограничения

- Ограничение по времени: 3 секунды.
- Ограничение по памяти: 64 МиБ.
- $1 \leq T \leq 1\,000\,000$
- $0 \leq A, B \leq 50\,000$ и $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2\,000\,000\,000$

Подзадачи

Подзадача	Баллы	Дополнительные ограничения
1	14	$T = 2$ и $A + B = 2$ (в точности два робота и в точности две игрушки)
2	14	$B = 0$ (все роботы — слабые)
3	25	$T \leq 50$ и $A + B \leq 50$
4	37	$T \leq 10\,000$ и $A + B \leq 1\,000$
5	10	(Нет)

Взаимодействие с проверяющим модулем

Проверяющий модуль на вашем компьютере будет читать входные данные из файла `robots.in`, который должен иметь следующий формат:

- строка 1: `A B T`
- строка 2: `X[0] ... X[A-1]`
- строка 3: `Y[0] ... Y[B-1]`
- каждая из следующих `T` строк содержит описание одной игрушки, каждое описание состоит из двух чисел `W[i] S[i]`

В частности, первый пример из описанных выше должен быть задан в таком формате:

```
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
```

Если `A = 0` или `B = 0`, то соответствующая строка (вторая или третья) должна быть пустой.

Особенности конкретных языков программирования

- C/C++** Вы должны подключить заголовочный файл с помощью `#include "robots.h"`.
- Pascal** Вы должны написать модуль с заголовком `unit Robots`. Все массивы нумеруются, начиная с `0` (а не с `1`).

Для примера посмотрите шаблоны решений на вашем компьютере.