

## International Olympiad in Informatics 2013

July 2013 6-13

Brisbane, Australia

Day 2 tasks



## روبات‌ها

Persian — ۱.۰

برادر کوچک ماریتا اسباب‌بازی‌هایش را کف اتاق رها کرده است! خوشبختانه ماریتا روبات‌های ویژه‌ای برای جمع‌آوری اسباب‌بازی‌ها طراحی کرده است. او به کمک شما برای تعیین آن که کدام روبات کدام اسباب‌بازی را جمع کند نیاز دارد.

تعداد  $T$  اسباب‌بازی داریم. اسباب‌بازی  $i$  ام دارای وزن  $W[i]$  و اندازه‌ی  $S[i]$  است که هر دو اعداد صحیح هستند. روبات‌ها به یکی از دو دسته‌ی زیر تعلق دارند: (۱) دسته‌ی ضعیف (۲) دسته‌ی کوچک.

▪ تعداد  $A$  روبات ضعیف داریم. هر روبات ضعیف دارای محدودیت وزنی  $X[i]$  می‌باشد، به این معنی که فقط اسباب‌بازی با وزن کمتر از  $X[i]$  را می‌تواند حمل کند. برای روبات‌های ضعیف اندازه‌ی اسباب‌بازی‌ها اهمیتی ندارد.

▪ تعداد  $B$  روبات کوچک داریم. هر روبات کوچک دارای محدودیت اندازه‌ی  $Y[i]$  است، به این معنی که فقط اسباب‌بازی‌های با اندازه‌ی کمتر از  $Y[i]$  را می‌تواند حمل کند. وزن اسباب‌بازی‌ها برای روبات‌های کوچک اهمیتی ندارد.

هر روبات ماریتا به یک دقیقه زمان برای جمع‌آوری هر اسباب‌بازی نیاز دارد. روبات‌های متفاوت می‌توانند اسباب‌بازی‌های متفاوت را به طور هم‌زمان جمع‌آوری کنند.

شما باید تعیین کنید که آیا روبات‌های ماریتا می‌توانند همه‌ی اسباب‌بازی‌ها را جمع کنند، و اگر این گونه است کم‌ترین زمان لازم را محاسبه کنید.

### مثال‌ها

به عنوان مثال اول، فرض کنید که تعداد  $A = 3$  روبات ضعیف با محدودیت وزنی  $X = [6, 2, 9]$ ، و تعداد  $B = 2$  روبات کوچک با محدودیت اندازه‌ی  $Y = [4, 7]$  داریم. در ضمن فرض کنید تعداد اسباب‌بازی‌ها  $T = 10$  است.

شماره‌ی اسباب‌بازی	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
وزن	۴	۸	۲	۷	۱	۵	۳	۸	۷	۱۰
اندازه	۶	۵	۳	۹	۸	۱	۳	۷	۶	۵

کم‌ترین زمانی که می‌توان همه‌ی اسباب‌بازی‌ها را جمع کرد ۳ دقیقه است.

روبات ضعیف ۰	روبات ضعیف ۱	روبات ضعیف ۲	روبات کوچک ۰	روبات کوچک ۱	
اسباب‌بازی ۰	اسباب‌بازی ۴	اسباب‌بازی ۱	اسباب‌بازی ۶	اسباب‌بازی ۲	دقیقه‌ی اول
اسباب‌بازی ۵		اسباب‌بازی ۳		اسباب‌بازی ۸	دقیقه‌ی دوم
		اسباب‌بازی ۷		اسباب‌بازی ۹	دقیقه‌ی سوم

به عنوان مثال دوم، فرض کنید تعداد  $A = 2$  روبات ضعیف با محدودیت وزنی  $X = [2, 5]$ ، و تعداد  $B = 1$  روبات کوچک با محدودیت اندازه‌ای  $Y = [2]$  داریم. در ضمن فرض کنید تعداد اسباب‌بازی‌ها  $T = 3$  است.

شماره‌ی اسباب‌بازی	۰	۱	۲
وزن	۳	۵	۲
اندازه	۱	۳	۲

هیچ روباتی نمی‌تواند اسباب‌بازی با وزن ۵ و اندازه‌ی ۳ را بردارد. بنابراین جمع‌آوری همه‌ی اسباب‌بازی‌ها غیرممکن است.

## پیاده‌سازی

شما باید تابع `putaway()` را در یک فایل به صورت زیر پیاده‌سازی و ارسال کنید:

### تابع شما: `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;
```

### توضیحات

این تابع باید کم‌ترین زمان ممکن (بر حسب دقیقه) را که روبات‌ها نیاز دارند تا تمام اسباب‌بازی‌ها را جمع‌آوری کنند محاسبه کند، یا عدد  $-1$  را به معنی غیر ممکن بودن برگرداند.

### پارامترها

- $A$ : تعداد روبات‌های ضعیف.
- $B$ : تعداد روبات‌های کوچک.
- $T$ : تعداد اسباب‌بازی‌ها.
- $X$ : یک آرایه با اندازه‌ی  $A$  از اعداد صحیح که محدودیت وزنی هر روبات ضعیف را مشخص می‌کند.
- $Y$ : یک آرایه با اندازه‌ی  $B$  از اعداد صحیح که محدودیت اندازه‌ی هر روبات کوچک را مشخص می‌کند.

- W: یک آرایه با اندازه‌ی T از اعداد صحیح که وزن اسباب‌بازی‌ها را مشخص می‌کند.
- S: یک آرایه با اندازه‌ی T از اعداد صحیح که اندازه‌ی اسباب‌بازی‌ها را مشخص می‌کند.
- خروجی: کم‌ترین زمان موردنیاز (برحسب دقیقه) برای جمع کردن اسباب‌بازی‌ها، یا -1 اگر چنین کاری امکان‌پذیر نباشد.

## اجرای نمونه

اجرای زیر مربوط به مثال اول بالا است:

Parameter	Value
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

اجرای زیر مربوط به مثال دوم بالا است:

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

## محدودیت‌ها

- محدودیت زمان: ۳ ثانیه

▪ محدودیت حافظه: ۶۴ مگابایت

▪  $1 \leq T \leq 1,000,000$

▪  $1 \leq A + B$  و  $0 \leq A, B \leq 50,000$

▪  $1 \leq X[i], Y[i], W[i], S[i] \leq 2,000,000,000$

## زیرمسئله‌ها

محدودیت‌های ورودی اضافی	امتیاز	زیرمسئله
$T = 2$ و $A + B = 2$ (دقیقاً دو اسباب‌بازی و دو روبات)	۱۴	۱
$B = 0$ (تمام روبات‌ها ضعیف‌اند)	۱۴	۲
$T \leq 50$ و $A + B \leq 50$	۲۵	۳
$T \leq 10,000$ و $A + B \leq 1,000$	۳۷	۴
(بدون محدودیت اضافی)	۱۰	۵

## آزمایش

مصححی که روی کامپیوتر شما قرار دارد ورودی را از فایل `robots.in` می‌خواند، که این فایل باید به شکل زیر باشد:

▪ خط ۱: `A B T`

▪ خط ۲: `X[0] ... X[A-1]`

▪ خط ۳: `Y[0] ... Y[B-1]`

▪ `T` خط بعد: `W[i] S[i]`

برای نمونه، مثال اول بالا باید به شکل زیر داده شود:

```
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
```

اگر `A = 0` یا `B = 0` آن‌گاه خط متناظر (خط ۲ یا ۳) باید خالی باشد.

---

## نکات زبان

C/C++ عبارت `#include "robots.h"` را باید به برنامه‌ی خود اضافه کنید.

Pascal باید `unit Robots` را تعریف کنید. تمام آرایه‌ها از `0` (و نه `1`) شروع می‌شوند.

برای دیدن مثال‌ها به راه‌حل‌های نمونه بر روی کامپیوتر خود مراجعه کنید.