



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

robots

Greek — 1.0

Ο μικρός αδελφός της Μαρίας έχει αφήσει τα παιχνίδια του σκορπισμένα στο πάτωμα του καθιστικού του.

Ευτυχώς η Μαρία έχει κατασκευάσει κάποια ειδικά ρομπότ για να καθαρίσουν τα παιχνίδια. Χρειάζεται όμως την βοήθειά σου, για να καθορίσει πιο ρομπότ θα μαζέψει ποια παιχνίδια.

Υπάρχουν T παιχνίδια, που το κάθε ένα έχει βάρος που συμβολίζεται από τον ακέραιο $W[i]$ και μέγεθος που συμβολίζεται από τον ακέραιο $S[i]$. Τα ρομπότ είναι δύο ειδών: *αδύναμα* και *μικρά*.

- Υπάρχουν A αδύναμα ρομπότ. Το κάθε αδύναμο ρομπότ μπορεί να σηκώσει βάρος μέχρι το όριο $X[i]$, και μπορεί να μεταφέρει οποιοδήποτε παιχνίδι βάρους σίγουρα μικρότερο από $X[i]$. Το μέγεθος του παιχνιδιού δεν παίζει κανένα ρόλο.
- Υπάρχουν B μικρά ρομπότ. Κάθε μικρό ρομπότ μπορεί να σηκώσει μέγεθος μέχρι το όριο $Y[i]$, και μπορεί να μεταφέρει οποιοδήποτε παιχνίδι μεγεθους σίγουρα μικρότερο από $Y[i]$. Το βάρος του παιχνιδιού δεν παίζει κανένα ρόλο.

Κάθε ένα από τα ρομπότ χρειάζεται ένα λεπτό για να απομακρύνει ένα παιχνίδι. Διαφορετικά ρομπότ μπορούν να απομακρύνουν διαφορετικά παιχνίδια ταυτόχρονα.

Το πρόβλημα σας είναι να προσδιορίσετε κατά πόσο τα ρομπότ της Μαρίας μπορούν να απομακρύνουν όλα τα παιχνίδια και αν ναι, ποιος είναι ο συντομότερος χρόνος που μπορούν να το κάνουν αυτό.

Παραδείγματα

Σαν πρώτο παράδειγμα, υποθέστε ότι υπάρχουν $A = 3$ αδύναμα ρομπότ που μπορούν να μεταφέρουν βάρος κάτω από τα όρια $X = [6, 2, 9]$, $B = 2$ μικρά ρομπότ με όρια μεγέθους κάτω από $Y = [4, 7]$, και $T = 10$ παιχνίδια όπως φαίνεται παρακάτω:

Αριθμός παιχνιδιού	0	1	2	3	4	5	6	7	8	9
Βάρος	4	8	2	7	1	5	3	8	7	10
Μέγεθος	6	5	3	9	8	1	3	7	6	5

Ο συντομότερος χρόνος για να απομακρυνθούν όλα τα παιχνίδια είναι 3 λεπτά:

	Αδύναμο ρομπότ 0	Αδύναμο ρομπότ 1	Αδύναμο ρομπότ 2	Μικρό ρομπότ 0	Μικρό ρομπότ 1
1ο Λεπτό	Toy 0	Toy 4	Toy 1	Toy 6	Toy 2
2ο Λεπτό	Toy 5		Toy 3		Toy 8
3ο Λεπτό			Toy 7		Toy 9

Σαν δεύτερο παράδειγμα, υποθέστε ότι υπάρχουν $A = 2$ αδύναμα ρομπότ που μπορούν να μεταφέρουν βάρος κάτω από τα όρια $X = [2, 5]$, $B = 1$ μικρά ρομπότ που μπορούν να μεταφέρουν μεγεθος κάτω από τα όρια $Y = [2]$, και $T = 3$ παιχνίδια όπως φαίνεται παρακάτω:

Αριθμός Παιχνιδιού	0	1	2
Βάρος	3	5	2
Μέγεθος	1	3	2

Κανένα από τα ρομπότ δεν μπορεί να μεταφέρει παιχνίδι με βάρος 5 και μέγεθος 3, και έτσι είναι αδύνατο για τα ρομπότ να απομακρύνουν όλα τα παιχνίδια.

Υλοποίηση

Πρέπει να υποβάλλετε ένα αρχείο που να υλοποιεί την συνάρτηση `putaway()` ως εξής:

Η συνάρτηση: `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;
```

Περιγραφή

Αυτή η συνάρτηση θα πρέπει να υπολογίζει τον ελάχιστο αριθμό λεπτών που απαιτούνται για τα ρομπότ να απομακρύνουν όλα τα παιχνίδια, ή να επιστρέψει `-1` αν αυτό δεν είναι δυνατό.

Παράμετροι

- A : Ο αριθμός των αδύναμων ρομπότ.
- B : Ο αριθμός των μικρών ρομπότ.
- T : Ο αριθμός των παιχνιδιών.
- X : Πίνακας μήκους A που περιέχει ακεραίους οι οποίοι προσδιορίζουν το όριο του βάρους για κάθε ένα από τα αδύναμα ρομπότ.
- Y : Πίνακας μήκους B που περιέχει ακεραίους οι οποίοι προσδιορίζουν το όριο του μεγεθους για κάθε ένα από τα μικρά ρομπότ.
- W : Πίνακας μήκους T που περιέχει ακεραίους που δίνουν το βάρος του κάθε παιχνιδιού.

- S : Πίνακας μήκους T που περιέχει ακέραιους που δίνουν το μέγεθος κάθε παιχνιδιού.
- *Επιστρέφει*: Τον μικρότερο αριθμό σε λεπτά που απαιτούνται για να απομακρυνθούν όλα τα παιχνίδια, ή -1 εάν αυτό δεν είναι εφικτό.

Παράδειγμα

Το παρακάτω περιγράφει το πρώτο από τα παραπάνω παραδείγματα:

Παράμετρος	Τιμή
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]
Επιστρέφει	3

Το παρακάτω περιγράφει το δεύτερο από τα παραπάνω παραδείγματα:

Παράμετροι	Τιμές
A	2
B	1
T	3
X	[2, 5]
Y	[2]
W	[3, 5, 2]
S	[1, 3, 2]
Επιστρέφει	-1

Περιορισμοί

- Οριο Χρόνου: 3 δευτερόλεπτα
- Οριο Μνήμης: 64 MiB (Megabyte)
- $1 \leq T \leq 1,000,000$
- $0 \leq A, B \leq 50,000$ and $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2,000,000,000$

Υποπροβλήματα

Υποπροβλήματα	Βαθμοί	Επιπλέον Περιορισμοί κατά την εισαγωγή
1	14	$T = 2$ and $A + B = 2$ (ακριβώς 2 παιχνίδια και 2 ρομπότ)
2	14	$B = 0$ (όλα τα ρομπότ είναι αδύναμα)
3	25	$T \leq 50$ και $A + B \leq 50$
4	37	$T \leq 10,000$ και $A + B \leq 1,000$
5	10	(Κανένα)

Πειραματισμός

Το πρόγραμμα αξιολόγησης (grader) στον υπολογιστή σας θα πάρει εισαγωγή από το αρχείο `robots.in`, Το οποίο πρέπει να έχει την ακόλουθη μορφή:

- γραμμή 1: `A B T`
- γραμμή 2: `X[0] ... X[A-1]`
- γραμμή 3: `Y[0] ... Y[B-1]`
- επόμενες `T` γραμμές: `W[i] S[i]`

Έτσι, το πρώτο παράδειγμα θα πρέπει να παράξει την παρακάτω έξοδο (output) :

```
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
```

Εάν `A = 0` ή `B = 0` τότε η αντίστοιχη γραμμή (γραμμή 2 ή γραμμή 3) πρέπει να είναι κενή.

Οδηγίες για τη γλώσσα

- C/C++** Στον κώδικα χρησιμοποιείστε την οδηγία `#include "robots.h"`.
- Pascal** Στον κώδικα θα πρέπει να οριστεί `unit Robots`. Η αρίθμηση των στοιχείων των πινάκων ξεκινά από `0` (και όχι `1`).

See the solution templates on your machine for examples.