



International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

robots

English — 1.0

მარიტას პატარა ძმა თამაშის შემდეგ სათამაშოებს ყოველთვის იატაკზე მიმოხედავს ტოვებს. საბედნიეროდ, მარიტამ მოახერხა და სათამაშოების ასალაგებლად სპეციალური რობოტები გამოიგონა. ოღონდ, მას თქვენი დახმარება სჭირდება იმის დასადგენად, რომელ რობოტებს რომელი სათამაშოების ალაგება შეუძლიათ.

იატაკზე T რაოდენობის სათამაშოა, რომელთაგან თითოეულს მთელი რიცხვით გამოსახული $W[i]$ წონა და ასევე მთელი რიცხვით გამოსახული $S[i]$ ზომა გააჩნია. რობოტები სულ ორი სახისაა: სუსტი და პატარა.

მარიტას აქვს A რაოდენობის სუსტი რობოტი და B რაოდენობის პატარა რობოტი. თითოეულ სუსტ რობოტს აქვს წონის შეზღუდვა $X[i]$ და შეუძლია მხოლოდ ნებისმიერი ისეთი სათამაშოს აღება, რომლის წონაც მკაცრად ნაკლებია $X[i]$ -ზე. ამ შემთხვევაში სათამაშოს ზომას მნიშვნელობა არა აქვს. თითოეულ პატარა რობოტს აქვს ზომის შეზღუდვა $Y[i]$ და შეუძლია მხოლოდ ნებისმიერი ისეთი სათამაშოს აღება, რომლის ზომაც მკაცრად ნაკლებია $Y[i]$ -ზე. ამ შემთხვევაში სათამაშოს წონას მნიშვნელობა არა აქვს. გარდა ამისა, არცერთ რობოტს არ შეუძლია ერთ ჯერზე ერთზე მეტი სათამაშოს აღება.

მარიტას ყოველ რობოტს თითოეული სათამაშოს ასაღებად და საჭირო ადგილზე დასადგებად (ასალაგებლად) ერთი წუთი სჭირდება. სხვადასხვა რობოტებს შეუძლიათ სხვადასხვა სათამაშოები პარალელურად (დროის ერთი და იგივე პერიოდში) აალაგონ.

თქვენი ამოცანაა დაადგინოთ, შეუძლიათ თუ არა მარიტას რობოტებს ყველა სათამაშოს ალაგება, და თუ შეუძლიათ, მაშინ რა მინიმალური დრო დასჭირდებათ ამის გასაკეთებლად.

მაგალითები

მაგალითი 1.

დავუშვათ, რომ მარიტას აქვს $A = 3$ ცალი სუსტი რობოტი, რომელთათვისაც წონების შეზღუდვებია $X = [6, 2, 9]$ შესაბამისად, და $B = 2$ ცალი პატარა რობოტი, რომელთათვისაც ზომების შეზღუდვებია $Y = [4, 7]$ შესაბამისად. გარდა ამისა, ოთახში მიმოხეულია $T = 10$ ცალი სათამაშო, რომელთა წონები და ზომებია:

სათამაშოს ნომერი	0	1	2	3	4	5	6	7	8	9
წონა	4	8	2	7	1	5	3	8	7	10
ზომა	6	5	3	9	8	1	3	7	6	5

ამ შემთხვევაში, მინიმალური დრო ყველა ამ სათამაშოს ასალაგებლად არის სამი წუთი:

	სუსტი რობოტი 0	სუსტი რობოტი 1	სუსტი რობოტი 2	პატარა რობოტი 0	პატარა რობოტი 1
პირველი წუთი	სათამაშო 0	სათამაშო 4	სათამაშო 1	სათამაშო 6	სათამაშო 2
მეორე წუთი	სათამაშო 5		სათამაშო 3		სათამაშო 8
მესამე წუთი			სათამაშო 7		სათამაშო 9

მაგალითი 2.

დავუშვათ, რომ მართლაც აქვს $A = 2$ ცალი სუსტი რობოტი, რომელთათვისაც წონების შეზღუდვებია $X = [2, 5]$ შესაბამისად, და $B = 1$ ცალი პატარა რობოტი, რომლისთვისაც ზომის შეზღუდვაა $Y = [2]$. გარდა ამისა, ოთახში მიმოზნეულია $T = 3$ ცალი სათამაშო, რომელთა წონები და ზომებია:

სათამაშოს ნომერი	0	1	2
წონა	3	5	2
ზომა	1	3	2

არცერთ რობოტს არ შეუძლია სათამაშოს აღება, რომლის წონაა 5 და ზომა 3. ამიტომ, ამ რობოტებით შეუძლებელია მოცემული სათამაშოების ალაგება.

იმპლემენტაცია

თქვენ უნდა გააგზავნოთ ფაილი, რომელშიც უნდა მოახდინოთ ფუნქცია `putaway()`-ის იმპლემენტაცია შემდეგნაირად:

თქვენი ფუნქცია: **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;
```

აღწერა

ამ ფუნქციამ უნდა გამოთვალოს წუთების ის მინიმალური რაოდენობა, რომელიც დასჭირდებათ რობოტებს ყველა სათამაშოს ასალაგებლად, ან დააბრუნოს -1 , თუ ეს შეუძლებელია.

პარამეტრები

- **A**: სუსტი რობოტების რაოდენობა.
- **B**: პატარა რობოტების რაოდენობა.
- **T**: სათამაშოების რაოდენობა.
- **X**: მასივი სიგრძით **A**, რომელიც შეიცავს თითოეული სუსტი რობოტის წონათა შეზღუდვებს გამოსახულს მთელი რიცხვებით.
- **Y**: მასივი სიგრძით **B**, რომელიც შეიცავს თითოეული პატარა რობოტის ზომათა შეზღუდვებს გამოსახულს მთელი რიცხვებით.
- **W**: მასივი სიგრძით **T**, რომელიც შეიცავს თითოეული სათამაშოს მთელი რიცხვებით გამოსახულ წონებს.
- **S**: მასივი სიგრძით **T** რომელიც შეიცავს თითოეული სათამაშოს მთელი რიცხვებით გამოსახულ ზომებს.
- *Returns*: წუთების ის მინიმალური რაოდენობა, რომელიც დასჭირდებათ რობოტებს ყველა სათამაშოს ასალაგებლად, ან -1 , თუ ეს შეუძლებელია.

სეანსის ნიმუში

შემდეგი სეანსი აღწერს ზემოთ მოტანილ პირველ მაგალითს:

პარამეტრი	მნიშვნელობა
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

შემდეგი სეანსი აღწერს ზემოთ მოტანილ მეორე მაგალითს:

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

შეზღუდვები

- დროის ლიმიტი: 3 წამი
- მეხსიერების ლიმიტი: 64 MiB
- $1 \leq T \leq 1,000,000$
- $0 \leq A, B \leq 50,000$ და $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2,000,000,000$

ქვეამოცანები

ქვეამოცანა	ქულები	შესატანი მონაცემების დამატებითი შეზღუდვა
1	14	$T = 2$ და $A + B = 2$ (ზუსტად ორი სათამაშო და ორი რობოტი)
2	14	$B = 0$ (ყველა რობოტი სუსტია)
3	25	$T \leq 50$ და $A + B \leq 50$
4	37	$T \leq 10,000$ და $A + B \leq 1,000$
5	10	(None)

ექსპერიმენტირება

სანიმუშო გრადერმა თქვენი კომპიუტერიდან უნდა წაიკითხოს შესატანი მონაცემები ფაილიდან `robots.in` შემდეგი ფორმატით:

- სტრიქონი 1: `A B T`
- სტრიქონი 2: `X[0] ... X[A-1]`
- სტრიქონი 3: `Y[0] ... Y[B-1]`
- შემდეგი `T` რაოდენობის სტრიქონი: `W[i] S[i]`

მაგალითად, ზემოთ მოტანილი პირველი მაგალითი წარმოდგენილი უნდა იყოს შემდეგი ფორმატით:

```
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
```

თუ `A = 0` ან `B = 0`, მაშინ შესაბამისი სტრიქონი (სტრიქონი 2 ან სტრიქონი 3) ცარიელი იქნება.

შენიშვნები პროგრამული ენებისათვის

C/C++ თქვენ უნდა გამოიყენოთ `#include "robots.h"`.

Pascal You must define the `unit Robots`. All arrays are numbered beginning at `0` (not `1`).

მაგალითებისათვის ამოხსნათა შაბლონები იხილეთ თქვენს კომპიუტერზე.