

Όνομα Άσκησης: Milking Cows

Πηγή: USACO 1.2

Εκφώνηση

Τρεις αγρότες σηκώνονται κάθε πρωί στις 5:00 και πηγαίνουν στο στάβλο για να αρμέξουν τρεις αγελάδες. Ο 1^{ος} αγρότης ξεκινά να αρμέγει την αγελάδα του στην ώρα 300 (σημαίνει 300 δευτερόλεπτα μετά τις 5:00) και τελειώνει την ώρα 1000. Ο 2^{ος} ξεκινά στην ώρα 700 και τελειώνει την ώρα 1200 ενώ ο 3^{ος} στην ώρα 1500 και 2100 αντίστοιχα. Ο μεγαλύτερος χρόνος συνεχούς αρμέγματος (δηλαδή υπήρχε τουλάχιστον ένας αγρότης που άρμεγε) ήταν 900 δευτερόλεπτα (300 με 1200). Αντιστοίχως ο μέγιστος χρόνος όπου καμιά αγελάδα δεν αρμεγόταν ήταν 300 (1500 πλην 1200).

Στόχος σας είναι να γράψετε ένα πρόγραμμα που θα δέχεται τους χρόνους N αγροτών ($1 \leq N \leq 5000$) που αρμέγουν N αγελάδες και θα τυπώνεται (σε δευτερόλεπτα):

- Το μέγιστο χρόνο συνεχούς αρμέγματος.
- Το μέγιστο χρόνο που καμιά αγελάδα δεν αρμεγόταν.

Δεδομένα Εισόδου

Στη γραμμή 1 υπάρχει ένας ακέραιος αριθμός που αντιστοιχεί στο N .

Στη συνέχεια υπάρχουν N ζευγάρια ακεραίων αριθμών που αντιστοιχούν στην έναρξη και τη λήξη του αρμέγματος για κάθε αγρότη. Οι αριθμοί είναι μικρότεροι του 1000000

Δεδομένα Εξόδου

Μια γραμμή που περιέχει 2 ακεραίους αριθμούς.

Παράδειγμα Εισόδου (milk2.in)

```
3
300 1000
700 1200
1500 2100
```

Παράδειγμα Εξόδου (milk2.out)

```
900 300
```

Επεξήγηση

Κάθε αγρότης αρμέγει μία αγελάδα, υπάρχουν όμως αγρότες που μπορεί να αρμέγουν τις αγελάδες τους ταυτόχρονα. Μπορεί επίσης να υπάρχει άρμεγμα των αγελάδων για σε όλο το διάστημα.

Είσοδος

2

100 200

200 300

Έξοδος

200 0

Μπορεί να υπάρχει μόνο ένας αγρότης.

Είσοδος

1

100 200

Έξοδος

100 0

Τέλος να επισημάνουμε ότι δεν λέει πουθενά ότι τα στοιχεία θα δίνονται ταξινομημένα

Hints

Ταξινομήστε. Όχι απαραίτητα παράλληλα.

Λύση

Το πρόβλημα λύνεται με ταξινόμηση. Αποθηκεύουμε τα ζευγάρια των αριθμών σε 2 μονοδιάστατους πίνακες (ή vectors) και τα ταξινομούμε. ΟΧΙ παράλληλη ταξινόμηση.

Για παράδειγμα

Είσοδος

4

100 150

20 200

300 700

600 620

Τα ταξινομημένα vectors (πίνακες) θα είναι ως εξής

Start	Finish
20	150
100	200
300	620
600	700

Ξεκινώ από τη 2^η θέση του πίνακα και ελέγχω αν ο χρόνος έναρξης στη θέση i είναι μικρότερος από τον χρόνο λήξης στη θέση $i-1$. Αν ισχύει αυτό τότε σημαίνει ότι υπάρχει οι 2 αγρότες δούλευαν ταυτόχρονα. Αν όχι υπολογίζω τους χρόνους και ελέγχω αν είναι μέγιστοι.

Κώδικας

```
#include<iostream>
```

```
#include<algorithm>
```

```
#include<vector>
```

```
#include<fstream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int N,i,checkAll=0;
```

```

long int x,y,startMax,max1,temp,max2=0;

vector< int > start,finish;


ifstream fin("milk2.in");

ofstream fout("milk2.out");

fin>>N;

//Αποθήκευση μεταβλητών στα 2 vectors

for(i=0;i<N;i++){

    fin>>x>>y;

    start.push_back(x);

    finish.push_back(y);

}

//Ταξινόμηση για το κάθε vector ξεχωριστά

sort(start.begin(), start.end());

sort(finish.begin(), finish.end());

//Αρχική ώρα για συνεχόμενο άρμεγμα

startMax=start[0];

//Αν είναι μόνο ένας τότε το μέγιστο άρμεγμα είναι η ώρα του 1ου αγρότη

max1=finish[0]-start[0];

for(i=1;i<N;i++){

    //έλεγχος για διακοπή αρμέγματος

    if(start[i]>finish[i-1]){

        temp=finish[i-1]-startMax;

        //έλεγχος αν αυτός ο χρόνος είναι μεγαλύτερος

        //από αυτόν που έχει καταχωρηθεί σαν μέγιστος

```

```

    if(temp>max1)

        max1=temp;

    //εύρεση χρόνου που δεν υπάρχει άρμεγμα και

    //έλεγχος αν είναι μέγιστος

    temp=start[i]-finish[i-1];

    if(temp>max2)

        max2=temp;


    startMax=start[i];

    //λειτουργεί σαν flag ότι υπάρχει χρόνος μη αρμέγματος

    checkAll=1;

}

}

//έλεγχος αν υπάρχει χρόνος μη αρμέγματος

if(checkAll==0)

    max1=finish[N-1]-start[0];

fout<<max1<<" "<<max2<<"\n";

return 0;

}

```