

Fiesta loca

Enlace original: <https://judge.org/problems/P27158.en>

La vida de Masao es dura. Vive en Alcorcón, está de ERTE, y ya no puede invitar casi amigos a casa. Pero hoy un montón de amigos le han dicho que vendrán. Y quiere saber si estará cumpliendo las medidas de seguridad.

Dados los tiempos de llegada y partida de cada uno de sus amigos, ¿cuántos amigos coincidirán a la vez como máximo? ¿Y durante cuanto rato?

Input Format

El problema consta de varios casos. Cada caso empieza con el número de amigos n . En cada una de las siguientes n líneas hay dos números l, p . Los tiempos de llegada y partida de cada amigo. Un caso con $n = 0$ indica el final de la entrada.

Constraints

$$1 \leq n \leq 10^5$$

$$0 \leq l < p \leq 10^9$$

Output format

Para cada caso escribid el número máximo de amigos y durante cuanto tiempo habrá este número de amigos.

Solución

Tendremos dos tipos de evento que ordenaremos por tiempo en el que suceden. El primero tipo será la llegada de un amigo y el segundo la partida. Iteraremos por los momentos en los que suceden estos eventos. Es fácil de ver que los máximos se dan en estos momentos. Llevaremos un contador de cuantos amigos hay en cada momento y en el caso que estemos en un intervalo en el que estén el máximo de amigos hasta el momento otro contador que indica desde cuando tenemos este número máximo. Observar que hay que tener cuidado con el caso en el que en un mismo momento llegan y se van varios amigos.

Código

C++

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main(){
8      int n;
9      while(cin >> n and n != 0){
10         vector<pair<int, int> > v;
11         for(int i = 0; i < n; ++i){
12             int a,b;
13             cin >> a >> b;
14             // 0 indica que llega y 1 que se va
15             v.push_back(pair<int,int>({a,0}));
16             v.push_back(pair<int,int>({b,1}));
17         }
18         sort(v.begin(), v.end()); // ordenamos los eventos por tiempo y por tipo
19         int maxAmigos = 0, maxTiempo = 0;
20         int contadorAmigos = 0, ultimoTiempo = v[0].first;
21         int it = 0;
22         while(it < 2*n){ // hay 2n eventos
23             int momento = v[it].first;
24             if(contadorAmigos > maxAmigos){
25                 maxAmigos = contadorAmigos;
26                 maxTiempo = momento - ultimoTiempo;
27             }
28             else if(contadorAmigos == maxAmigos){
29                 maxTiempo = max(maxTiempo, momento-ultimoTiempo);
30             }
31             int amigosAntiguo = contadorAmigos;
32             //actualizamos todos los eventos en el momento de golpe
33             while(it < 2*n and v[it].first == momento){
34                 if(v[it].second == 0) contadorAmigos++;
35                 else contadorAmigos--;
36                 ++it;
37             }
38             //si antes y despues hay los mismos amigos tenemos un intervalo mas
39             //↪ largo
40             if(amigosAntiguo != contadorAmigos)
41                 ultimoTiempo = momento;
42         }
43         cout << maxAmigos << " " << maxTiempo << endl;
44     }
}

```

Python

```
1 line = input().strip().split()
2 n = int(line[0])
3
4 while n != 0:
5
6     v = []
7     for i in range(n):
8         v.append([int(line[2*i+1]), 0])
9         v.append([int(line[2*i+2]), 1])
10    v.sort()
11    maxAmigos = 0
12    maxTiempo = 0
13    contadorAmigos = 0
14    ultimoTiempo = v[0][0]
15    it = 0
16    while it < 2*n: # hay 2n eventos
17        momento = v[it][0]
18        if contadorAmigos > maxAmigos:
19            maxAmigos = contadorAmigos
20            maxTiempo = momento - ultimoTiempo
21
22        elif contadorAmigos == maxAmigos:
23            maxTiempo = max(maxTiempo, momento-ultimoTiempo)
24
25        amigosAntiguo = contadorAmigos
26        # actualizamos todos los eventos en el momento de golpe
27        while it < 2*n and v[it][0] == momento:
28            if v[it][1] == 0:
29                contadorAmigos = contadorAmigos + 1
30            else:
31                contadorAmigos = contadorAmigos - 1
32            it = it+1
33
34        # si antes y despues hay los mismos amigos tenemos un intervalo mas largo
35        if amigosAntiguo != contadorAmigos:
36            ultimoTiempo = momento
37
38    print(str(maxAmigos) + " " + str(maxTiempo))
39    line = input().strip().split()
40    n = int(line[0])
```