

# BỔ TRÍ PHÒNG HỌP

Có  $n$  cuộc họp đánh số từ 1 đến  $n$  đăng ký làm việc tại một phòng hội thảo. Cuộc họp  $i$  cần được bắt đầu ngay sau thời điểm  $s_i$  và kết thúc tại thời điểm  $f_i$ . Hỏi có thể bố trí phòng hội thảo phục vụ được nhiều nhất bao nhiêu cuộc họp, sao cho khoảng thời gian làm việc của hai cuộc họp bất kỳ là không giao nhau.

## Input

Dòng đầu tiên chứa số nguyên dương  $n$  ( $n \leq 10000$ )

Dòng thứ  $i$  trong số  $n$  dòng tiếp theo chứa hai số nguyên dương  $s_i, f_i$  ( $s_i < f_i \leq 32000$ ) ( $1 \leq i \leq n$ ).

## Output

Dòng đầu tiên ghi số  $K$  là số các cuộc họp được chấp nhận phục vụ

## Gợi ý:

– Các bạn dùng sắp xếp theo chiều tăng dần của  $B[i]$  ( $1 \leq i \leq N$ ) rồi dùng thuật toán tham lam tìm dãy cuộc họp từ  $1..i$  thỏa mãn  $B[i] > A[i] \geq B[1..i-1]$ .

## Code Tham Khao Pascal:

```
const  fi="";

      nmax=10000;

type

      data=longint;

      dulieu=record

          u,v:data;

      end;

var

      f:text;

      A:array[1..nmax] of dulieu;

      n:data;

procedure docfile;

var    i:data;
```

```

begin
    assign(f,fi); reset(f);

    readln(f,n);

    for i:=1 to n do
        readln(f,a[i].u,a[i].v);

    close(f);
end;

procedure sort(l,r: longint);

var
    i,j,x: longint;
    y:dulieu;

begin
    i:=l;
    j:=r;

    x:=a[(l+r) div 2].v;

    repeat
        while a[i].v<x do
            inc(i);

        while x<a[j].v do
            dec(j);

        if not(i>j) then
            begin
                y:=a[i];
                a[i]:=a[j];
                a[j]:=y;
            end;
    until i=j;
end;

```

```

        inc(i);

        j:=j-1;

    end;

until i>j;

if l<j then

    sort(l,j);

    if i<r then

        sort(i,r);

    end;

procedure xuli;

var    i,j:data;

        res,cuoi,k:data;

begin

    res:=0; cuoi:=0;

    i:=1;

    while i<=n do

        begin

            if (a[i].u>=cuoi) and (a[i].v>cuoi) then

                begin

                    inc(res);

                    cuoi:=a[i].v;

                end;

            inc(i);

        end;

    writeln(res);

```

```
end;
```

```
begin
```

```
    docfile;
```

```
    sort(1,n);
```

```
    xuli;
```

```
end.
```

## CODE 2

```
Const  Fi="";
```

```
    Fo="";
```

```
Type  Data  =    record
```

```
    Site, Start, Finish: Longint;
```

```
End;
```

```
Var   F:      Text;
```

```
    A:      Array[1..10000] of Data;
```

```
    N, I, J, Sum, Last:  Longint;
```

```
Procedure  Init;
```

```
    Begin
```

```
        Assign(f,fi);
```

```
        Reset(f);
```

```
        Readln(f,n);
```

```
        For i:=1 to n do
```

```
            Readln(f,a[i].Start,a[i].Finish);
```

close(f);

Sum:=0;

Last:=0;

i:=1;

End;

Procedure sort(l,r: longint);

var

i,j,x: longint;

y:data;

begin

i:=l;

j:=r;

x:=a[(l+r) div 2].Finish;

repeat

while a[i].Finish<x do

inc(i);

while x<a[j].Finish do

dec(j);

if not(i>j) then

begin

y:=a[i];

a[i]:=a[j];

a[j]:=y;

inc(i);

j:=j-1;

```

        end;
    until i>j;
    if l<j then
        sort(l,j);
    if i<r then
        sort(i,r);
    end;
Procedure    Process;

Begin
    Sort(1,n);
    While i<=n do
        Begin
            If (a[i].Start>=Last) and (a[i].Finish>last) then
                Begin
                    Inc(Sum);
                    Last:=a[i].Finish;
                    Inc(k);
                End;
            Inc(i);
        End;
    End;
End;

Procedure    Print;

Begin
    Assign(f,fo);
    Rewrite(f);

```

```
    Writeln(f,Sum);
```

```
    Close(f);
```

```
End;
```

```
Begin
```

```
    Init;
```

```
    Process;
```

```
    Print;
```

```
End.
```