

XẾP HÀNG

Trong giờ ăn trưa tại Học viện Công nghệ Bưu chính Viễn thông, có n sinh viên đang xếp hàng để lấy đồ.

Cảm thấy chán vì phải đứng đợi một mình, vì vậy mỗi sinh viên viết ra mã sinh viên của mình đứng ngay trước và ngay sau của mình. Nếu không có ai đứng trước hoặc không có ai đứng sau thì viết ra 0.

Đột nhiên, xe chở nước sôi đi qua, tất cả sinh viên phải tránh. Khi họ trở lại, họ không nhớ vị trí của mình mà chỉ nhớ mã sinh viên của người đứng trước và người đứng sau.

Hãy giúp các sinh viên PTIT tìm lại vị trí của mình!!!!

Input

Dòng đầu tiên gồm số tự nhiên n ($2 \leq n \leq 2 \cdot 10^5$) – số lượng sinh viên.

N dòng tiếp theo, dòng thứ i gồm cặp số tự nhiên a_i, b_i ($0 \leq a_i, b_i \leq 10^6$), với a_i là mã sinh viên của người đứng trước, b_i là mã sinh viên của người đứng sau 1 sinh viên nào đó. Nếu không có ai đứng trước hoặc không có ai đứng sau nhập 0.

Mã sinh viên của mỗi sinh viên là khác nhau.

Output

Trên 1 dòng, in ra n số x_1, x_2, \dots, x_n , danh sách của các sinh viên theo thứ tự ban đầu.

Example

Input:

4

92 31

0 7

31 0

7 141

Output:

92 7 31 141

Minh họa test đề bài.

#####

Nhận Xét:

– Đề bài cho ta biết “Nếu không có ai đứng trước nhập 0” vậy có nghĩa ta sẽ xác định được kết quả ở vị trí thứ 2. tạm gọi số này là “số đầu tiên” đi.

Hướng dẫn:

– Từ nhận xét trên ta có thể tính được các vị trí 2, 4, 6, 8, n

– Từ số đầu tiên, ta thực hiện chèn nhị phân tìm số trong dãy A[i] (lưu ý là phải sort A, và B theo A trước), gọi k là vị trí tìm thấy. rồi ghi kq vào vị trí 4, rồi tiếp tục lấy B[k] làm mốc xong rồi chèn rồi ghi vào vị trí 6, tương tự....8,...10,...n.. dừng lại khi k=0.

– Tương tự từ gợi ý trên bạn có thể dễ dàng tìm kết quả cho các vị trí 1, 3, 5,

* Thật ra lúc làm bài này mình quên việc có thể dùng mảng đánh dấu. nên các bạn có thể dùng mảng đánh dấu. mà ko cần SORT và chèn nhị phân. code tham khảo mình viết theo ý tưởng chèn nhị phân. các bạn có thể code lại bằng đánh dấu sẽ dễ hơn...

####

Gợi ý của bạn Trình Gia Lạc:

“CÓ 2 THẲNG SUẤT HIỆN 1 LẦN TRONG ĐÓ THẲNG NÀO Ở BÊN TRÁI THÌ LÀ ĐẦU TIÊN, THẲNG THỨ 2 LÀ THẲNG ĐỨNG R TRONG CẶP (0, R), TỪ 2 THẲNG NÀY SUY LUẬN RA TẤT CẢ THẲNG CÒN LẠI: 1->3, 2 -> 4 ,...”

Lời giải từ BTC

Phân dãy sinh viên thành 2 dãy, một dãy là các sinh viên đứng ở vị trí chẵn, một dãy là các sinh viên đứng ở vị trí lẻ, sau đó gộp dãy.

Có thể xây dựng dãy các sinh viên ở vị trí lẻ bằng cách “nhảy” từ người có a = 0.

Dãy các sinh viên ở vị trí chẵn có thể xây dựng bằng phương pháp tương tự.

Code Tham khảo

```
const  fi="";

      nmax=2*100000;

type  data=longint;

var

      f:text;

      A,B,kq:array[0..nmax+1] of data;
```

n,sodau,so1,so2:data;

so:array[0..1000000] of data;

procedure sort(l,r: longint);

var

i,j,x,y: longint;

begin

i:=l;

j:=r;

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

repeat

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

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

if not(i>j) then

begin

y:=a[i]; a[i]:=a[j]; a[j]:=y;

y:=b[i]; b[i]:=b[j]; b[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 docfile;
```

```
var i,j,dem:data;
```

```
begin
```

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

```
    readln(f,n);
```

```
    fillchar(so,sizeof(so),0);
```

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

```
        begin
```

```
            readln(f,a[i],b[i]);
```

```
            so[a[i]]:=so[a[i]] xor 1;
```

```
            so[b[i]]:=so[b[i]] xor 1;
```

```
        end;
```

```
    for i:=0 to 1000000 do
```

```
        if so[i]<>0 then
```

```
            begin
```

```
                so1:=i;
```

```
                for j:=i+1 to 1000000 do
```

```
                    if so[j]<>0 then
```

```
                        begin
```

```
                            so2:=j;
```

```
                            break;
```

```
                        end;
```

```
                    break;
```

```

        end;

    close(f);

end;

function tknpA(x:data):data;

var

    dau,cuoi,giua:data;

begin

    dau:=1; cuoi:=n;

    while dau<=cuoi do

        begin

            giua:=(dau+cuoi) div 2;

            if A[giua]=X then

                exit(giua)

            else

                if x>a[giua] then

                    dau:=giua+1

                else

                    cuoi:=giua-1;

                end;

            end;

        exit(0);

    end;

end;

procedure xuli;

var    i,j,x,dem,vt:data;

```

```

begin
    sort(1,n);
    if tknpA(so1)<>0 then
        sodau:=so1
    else
        sodau:=so2;
    ////////////
    x:=B[1];
    kq[2]:=x;
    dem:=2;
    repeat
        vt:=tknpA(x);
        if vt<=1 then break;
        inc(dem,2);
        kq[dem]:=B[vt];
        x:=B[vt];
    until false;
    x:=sodau;
    kq[1]:=x;
    dem:=1;
    repeat
        vt:=tknpA(x);
        if vt<=1 then break;
        inc(dem,2);
        kq[dem]:=B[vt];

```

```
        x:=B[vt];  
until false;  
for i:=1 to n do  
    write(kq[i], ' ');  
end;  
  
begin  
    docfile;  
    xuli;  
end.
```