

Bài toán LCA (Least Common Ancestor) :

u vào : 1 cây v i n nh.

Ch t v n : v i 2 nút u, v b t k c a cây T, ch t v n LCA(u,v) cho bi t cha chung g n nh t c a 2 nh u,v trong cây T, t c là cho bi t nh xa g c nh t là cha c a c u, v.

Cách làm:

- T u vào ta có th xây d ng c m ng cha[1...n] v i cha[i] cho ta bi t nút cha c a nút i. Sau ó ta có th xây d ng m ng b[1...n][0...logN] v i b[i,j] cho ta bi t nút t iên th 2^j c a nút i. Xây d ng m ng b m t $N \log N$ s d ng ph ng pháp QH nh sau:

```
procedure creat;  
begin  
    b[0,0]:=0;  
    for i:=1 to n do b[i,0]:=cha[i];  
    for j:=1 to log do  
        for i:=0 to n do  
            b[i,j]:=b[b[i,j-1],j-1];  
end;
```

- G i d(i) là kho ng cách t i g c c a nút i. Ta tính d[i] b ng DFS nh sau:

```
function dfs(u:longint):longint;  
var v:longint;  
begin  
    free[u]:=false;  
    if u=0 then exit(0);  
    if not free[cha[u]] then  
        begin  
            d[u]:=d[cha[u]]+1;  
            exit(d[u]);  
        end  
    else  
        d[u]:=dfs(cha[u])+1;  
        exit(d[u]);  
end;  
procedure tinh;  
begin  
    for i:=1 to n do free[i]:=true;  
    free[0]:=false;  
    d[0]:=0;  
    for i:=1 to n do if free[i] then d[i]:=dfs(i);  
end;
```

- xác định LCA(u,v) ta thể hiện các bước sau :
 + Giả sử $d(u) > d(v)$, ta thay u bằng mẹ nút t tiên c a u n khi $d(u)=d(v)$.

Gi t là chênh l ch kho ng cách so v i g c c a u và v. Bình th ng a u v t tiên sao cho $d[u]=d[v]$ ta ch c n th c hi n $u=cha[u]$ t l n, nh ng nh th quá m t th i gian. nhanh h n ta l i d ng m ng b ã xây d ng trên. Ta c n a u v t tiên th t c a u, nh v y ta c n phân tích t thành t ng các l y th a c a 2 s d ng m ng b.

ví d c n a u v t tiên th 7 c a nó, ta có $7=2^2+2^1+2^0$

\Rightarrow t tiên th 7 c a u = $b[b[b[u,2],1],0]$

(có c $7=2^2+2^1+2^0$ ta s d ng hàm `getbit` d i).

+ Khi $d(u)=d(v)$ ta thay u và v b ng 2 nút t tiên t ng ng sao cho v n th a m ã $d(u)=d(v)$ n khi $u=v$. Khi ó ta có c k t qu c n tìm.

Trong quá trình thay m t nút b ng nút t tiên c a nó, ta s s d ng m ng b có th nh y m t l n c nhi u b c. Khi ó ph c t p c a thu t toán s là $\langle N \log N, \log N \rangle$.

C th :

```
function getbit(k,x:longint):byte;
begin
  getbit:=(x shr (k-1)) and 1;
end;
```

```
function lca(u,v:longint):longint;
var t,bit,k:longint;
begin
  if d[u]>=d[v] then
    begin
      if d[u]>d[v] then
        begin
          t:=d[u]-d[v];
          for bit:=log downto 1 do
            if getbit(bit,t)=1
              then u:=b[u,bit-1];
          end;
          for k:=log downto 0 do
            if b[u,k]<>b[v,k] then
              begin
                u:=b[u,k];
                v:=b[v,k];
              end;
          end;
        end;
      else
        t:=d[v];
      end;
    end;
  for k:=t downto 0 do
    if b[u,k]<>b[v,k] then
      begin
        u:=b[u,k];
        v:=b[v,k];
      end;
  end;
  return u;
```

```

        end else lca:=b[u,k];
        if u=v then lca:=u;
        end else result:=lca(v,u);
    end;

```

Áp dụng vào bài FSELECT(<http://vn.spoj.pl/problems/FSELECT/>)

Sau khi tham dự IOI và OLPSV, Nguyễn chuyển đến một ngôi nhà mới. Khu nhà mới của Nguyễn có N ngôi nhà nằm hàng xóm ($N \leq 200000$). Vì điều kiện mà Nguyễn thích các bạn trẻ là N . Giữa các ngôi nhà có một đường thẳng thành cây. Khoảng cách giữa hai căn nhà khác nhau là 1 đơn vị. Có K cuộc chạy ($K \leq N/2$) của Nguyễn đã ra làm quen với các bạn mới. Tính toán chi phí mà các bạn, Nguyễn muốn biết xem khoảng cách xa nhất của 2 ngôi nhà trong một cuộc chạy là bao nhiêu? Bạn hãy giúp Nguyễn giải quyết vấn đề này.

Input

- Dòng 1 gồm 2 số N và K .

- N dòng tiếp theo, dòng thứ i gồm 2 số x và y . Trong đó x là thứ tự của cuộc chạy mà bạn thứ i tham gia, y là nhà hàng xóm của bạn thứ i . Nếu $y = 0$ thì đó là gốc của khu dân cư (có thể hiểu là gốc của cây).

Output

Gồm K dòng, dòng thứ i thể hiện khoảng cách xa nhất tìm được giữa 2 ngôi nhà của 2 bạn thứ i trong cuộc chạy thứ i .

Example

Input :

6 2

1 3

2 1

1 0

2 1

2 1

1 5

Output :

3

CODE:

{ \$MODE OBJFPC }

uses math;

const fi= ”;

fo= ”;

nmax=200000;

type mg=record

g,cs,kc:longint;

end;

var n,k,i,j,x,y:longint;

free:array[0..nmax] of boolean;

f:text;

d,cha:array[0..nmax] of longint;

a:array[1..nmax] of mg;

sl:array[0..nmax div 2+1] of longint;

b:array[0..nmax,0..19] of longint;

log:integer;

procedure doc;

begin

assign(f,fi);

reset(f);

readln(f,n,k);

for i:=1 to k do sl[i]:=0;

for i:=1 to n do

begin

readln(f,x,y);

cha[i]:=y;

a[i].g:=x;

inc(sl[x]);

end;

close(f);

log:=trunc((ln(n)/ln(2))+1e-8)+1;

end;

function dfs(u:longint):longint;

var v:longint;

begin

free[u]:=false;

if u=0 then exit(0);

if not free[cha[u]] then

begin

d[u]:=d[cha[u]]+1;

```

        exit(d[u]);
    end
else
    d[u]:=dfs(cha[u])+1;
    exit(d[u]);
end;

```

```

procedure tinh;
begin
    for i:=1 to n do free[i]:=true;
    free[0]:=false;
    d[0]:=0;
    for i:=1 to n do if free[i] then d[i]:=dfs(i);
end;

```

```

procedure creat;
begin
    b[0,0]:=0;
    for i:=1 to n do b[i,0]:=cha[i];
    for j:=1 to log do
        for i:=0 to n do
            b[i,j]:=b[b[i,j-1],j-1];
        end;
    end;

```

```

function getbit(k,x:longint):byte;
begin
    getbit:=(x shr (k-1)) and 1;
end;

```

```

function lca(u,v:longint):longint;
var t,bit,k:longint;
begin
    if d[u]>=d[v] then
        begin
            if d[u]>d[v] then
                begin
                    t:=d[u]-d[v];
                    for bit:=log downto 1 do
                        if getbit(bit,t)=1
                            then u:=b[u,bit-1];
                    end;
                    for k:=log downto 0 do
                        if b[u,k]<>b[v,k] then
                            begin
                                u:=b[u,k];

```

```

        v:=b[v,k];
        end else lca:=b[u,k];
        if u=v then lca:=u;

        end else result:=lca(v,u);
end;

```

```

function kc(u,v:longint):longint;
var
    tt:longint;
begin
    tt:=lca(u,v);
    exit(d[u]+d[v]-2*d[tt]);
end;

```

```

procedure qs(l,r:longint);
var    d1,d2:longint;
        tam1:mg;
begin
    if l>=r then exit;
    i:=l; j:=r;
    d1:=a[(l+r)div 2].g;
    d2:=a[(l+r)div 2].kc;
    repeat
        while (a[i].g<d1) or ((a[i].g=d1)and(a[i].kc>d2)) do inc(i);
        while (a[j].g>d1) or ((a[j].g=d1)and(a[j].kc<d2)) do dec(j);
        if i<=j then
            begin
                tam1:=a[i];
                a[i]:=a[j];
                a[j]:=tam1;
                inc(i); dec(j);
            end;
        until i>j;
        qs(l,j); qs(i,r);
    end;
end;

```

```

procedure xuli;
var    t,max,cc,res,tam:longint;
begin

```

```

for i:=1 to n do
begin
a[i].cs:=i;
a[i].kc:=d[i];
end;
qs(1,n);
creat;
for i:=2 to k do sl[i]:=sl[i]+sl[i-1];
sl[0]:=0;
for t:=1 to k do
begin
max:=-1;
for i:=sl[t-1]+1 to sl[t] do
if d[a[i].cs]>max then
begin
max:=d[a[i].cs];
cc:=a[i].cs;
end;
res:=-1;
for i:=sl[t-1]+1 to sl[t] do
begin
tam:=kc(cc,a[i].cs);
if d[cc]+d[a[i].cs]>res then
begin
if res<tam then res:=tam;
end else if d[cc]+d[a[i].cs]<=res then break;
end;
writeln(f,res)
end;
end;
end;
end;

```

```

begin
doc;
tinh;
assign(f,fo);
rewrite(f);
xuli;
close(f);
end.

```