TIMUS 1004. Sightseeing Trip Solution

write by Gestalti Lur

2012-07-15

题目大意

计算一个无向图上包含最少三个点的最小环并输出其路径，如果没有则输出”No solution”。

算法分析

可以用 floyd 算法计算出其最小环，每次找到环的时候记录一下路径（是否有更好的方法？）即可。

参考代码

```
{
TIMUS 1004
Hint: use floyd to find minimum circle
write by gestapolur
2012-07-08
ACCEPTED
}
program timus1004;
const
  MAXN    = 100;
  INF   = 1 shl 28;
var
  n , m , cnt : longint;
  ans          : longint;
  f , w , pre : array[ 1..MAXN , 1..MAXN ] of longint;
  sv           : array[ 1..MAXN ] of longint;

function init() : boolean;
var
  i , sx , sy , sw : longint;
begin
  read( n );
  if n = -1 then
  begin
    readln;
    exit( false );
  end;
  readln( m );

  for sx := 1 to n do
    for sy := 1 to n do
    begin
     w[ sx , sy ] := INF;
     f[ sx , sy ] := INF;
    end;

  for i := 1 to m do begin
    readln( sx , sy , sw );
    if f[ sx , sy ] > sw then begin
```

```
      f[ sx , sy ] := sw;
      pre[ sx , sy ] := sx;
     end;
     if f[ sy , sx ] > sw then begin
      f[ sy , sx ] := sw;
      pre[ sy , sx ] := sy;
     end;
     w[ sx , sy ] := f[ sx , sy ];
     w[ sy , sx ] := f[ sy , sx ];
   end;

   exit( true );
end; { init }

procedure floyd();
var
   i , j , k , cur : longint;
begin
   ans := INF;
   cnt := 0;
   for k := 1 to n do begin
     for i := 1 to k - 1 do
      for j := 1 to k - 1 do
      begin
        if ( i <> j ) and ( w[ k , j ] <> INF ) and ( w[ i , k ] <> INF ) and ( ans > f[ j
, i ] + w[ i , k ] + w[ k , j ] ) then
        begin
           ans := f[ j , i  ] + w[ i , k ] + w[ k , j ];
           //writeln( ans , ' ' , f[ i , j ] , ' ' , f[ j , i ] , ' i-j ' , i , ' ' , j , ' ' , k , ' ' , w[ i , k ]
, ' ' , w[ k , j ] );
           cnt := 0;
           cur := i;
           repeat
            inc( cnt );
            sv[ cnt ] := cur;
            cur := pre[ j , cur ];
           until ( cur = i ) or ( cur = j );

           inc( cnt );
           sv[ cnt ] := j;
           inc( cnt );
           sv[ cnt ] := k;
        end;
      end;

     for i := 1 to k do
      for j := 1 to k do
      begin
```

```
       if ( i <> j ) and ( f[ i , k ] <> INF ) and ( f[ k , j ] <> INF ) and ( f[ i , j ] > f[
i , k ] + f[ k , j ] ) then
       begin
         f[ i , j ] := f[ i , k ] + f[ k , j ];
         pre[ i , j ] := pre[ k , j ];
       end;
       if ( f[ j , i ] <> INF ) and ( f[ k , j ] <> INF ) and ( f[ k , i ] > f[ k , j ] + f[ j ,
i ] ) then
       begin
         f[ k , i ] := f[ k , j ] + f[ j , i ];
         pre[ k , i ] := pre[ j , i ];
       end;
       if ( f[ i , j ] <> INF ) and ( f[ j , k ] <> INF ) and ( f[ i , k ] > f[ i , j ] + f[ j ,
k ] ) then
       begin
         f[ i , k ] := f[ i , j ] + f[ j , k ];
         pre[ i , k ] := pre[ j , k ];
       end;
     end;
  end;
end; { floyd }

procedure out();
var i : longint ;
begin
  if cnt = 0 then begin
    writeln('No solution.');
    exit();
  end;
  for i := 1 to cnt - 1 do
    write( sv[ i ] , ' ' );
  writeln( sv[ cnt ] );
end; { out }

begin
  while not eof do begin
    if init = false then
     break;
    floyd;
    out;
  end;
end.
```