

Algoritmizálás

Horváth Gyula

Szegedi Tudományegyetem
Természettudományi és Informatikai Kar
horvath@inf.u-szeged.hu

7. Gráfalgoritmusok II.

7.1. Feladat: Útcaseprő

A város utcáinak takarítását úgy szervezték meg, hogy minden éjszaka egy utcaseprő géppel végig kell menni minden utcán, mindkét irányban pontosan egyszer.

Készítsünk programot (SEPRO.PAS, SEPRO.C,), amely megad egy útvonalat, amelyen az utcaseprő gép haladva minden utcában mindkét irányban pontosan egyszer halad végig és visszaér a kiindulási helyére!

Bemenet

A SEPRO.BE szöveges állomány első sorában a kereszteződések N száma ($1 < N \leq 10000$), és az utcák M ($1 \leq M \leq 30000$) száma van. Az útkereszteződések az $1, \dots, N$ számokkal azonosítjuk. A következő M sor mindegyike két különböző egész számot tartalmaz, két útkereszteződés sorszámát, u és v , ami azt jelenti, hogy a u és v útkereszteződés között utca van. ($1 \leq u, v \leq N$). Bármely két kereszteződés között legfeljebb egy utca van. A város úthálózata olyan, hogy bármely kereszteződésből bármely másikba el lehet jutni.

Kimenet

A SEPRO.KI szöveges állomány egyetlen sorába olyan útvonalat kell írni, amely az 1. kereszteződésnél indul és minden utcában mindkét irányban pontosan egyszer halad végig, és visszaér az 1. kereszteződésbe! Több megoldás esetén bármelyik megoldható.

Példa bemenet és kimenet

```
sepro.be
5 7
1 2
1 3
2 3
2 5
3 5
2 4
4 5
sepro.ki
1 3 5 4 2 5 2 3 2 1 2 4 5 3 1
```

Megoldás

Az úthálózat irányítatlan gráf. Vegyünk egy 1. gyökerű mélységi feszítőfát. Ekkor a gráf minden éle vagy faél, vagy visszaél lesz. Járjuk be a feszítőfát a faélek mentén, továbbá, ha a mélységi bejárás során kiderül, hogy $p \rightarrow q$ visszaél, akkor menjünk el q -ba, majd vissza p -be.

```
1 program Utcasepro ;
2 const
3   MaxN=10000;
4   MaxM=100000;
```

```

5  type
6    Lanc=^ Cella;
7    Cella=record pont:integer; csat:Lanc end;
8    Paletta=(Feher, Szurke, Fekete);
9  var
10   n,m:longint;
11   G: array[1..MaxN] of Lanc;
12   Szin : array[1..MaxN] of Paletta;
13   Apa : array[1..MaxN] of integer;
14   KiF:Text;
15  var
16   i:longint;

17  procedure Beolvas;
18  var
19   BeF:Text;
20   x,y:integer;
21   p:lanc;
22  begin
23   assign(BeF,'sepro.be'); reset(BeF);
24   readln(BeF,n,m);
25   for i:=1 to n do G[i]:=nil;
26   for i:=1 to m do begin
27     read(BeF,x, y);
28     new(p);
29     p^.pont:=y;
30     p^.csat:=G[x];
31     G[x]:=p;
32     new(p);
33     p^.pont:=x;
34     p^.csat:=G[y];
35     G[y]:=p;
36   end;
37   close(BeF);
38  end;

39  procedure MelyBejar(p:integer);
40  var lp:Lanc;
41   q:integer;
42  begin
43   Szin[p]:=Szurke;
44   lp:=G[p];
45   while lp<>nil do begin
46     q:=lp^.pont;
47     if (Szin[q]=Feher) then begin
48       Apa[q]:=p;
49       write(KiF, q, ' ');
50       MelyBejar(q);
51       write(KiF, p, ' ');
52     end else if (Szin[q]=Szurke)and(Apa[p]<>q) then begin
53       write(KiF, q, ' ',p, ' ');
54     end;
55     lp:=lp^.csat;
56   end;
57   Szin[p]:=Fekete;
58  end;

59  begin {prog}

```

```

60  Beolvas;
61  for i:=1 to n do begin
62      Apa[ i ]:=0;
63      Szin[ i ]:=Feher;
64  end;
65  Apa[1]:= -1;
66  assign (KiF, 'sepro.ki'); rewrite (KiF);
67  write (KiF,1, ' ');
68  MelyBejar (1);
69  writelN (KiF);
70  close (KiF);
71  end.

```

7.2. Euler-út, Euler-köt

7.1. definíció. A $G = (V, E)$ (irányított vagy irányítatlan) gráfban Euler-séta olyan séta, amely a gráf minden élét pontosan egyszer tartalmazza.

7.2. definíció. A $G = (V, E)$ (irányított vagy irányítatlan) gráfban Euler-kör olyan séta, amely a gráf minden élét pontosan egyszer tartalmazza, beleértve a séta utolsó pontjából az első pontba vezető élet is.

7.3. tétel. A $G = (V, E)$ irányított gráfban akkor és csak akkor van Euler-kör, ha G erősen összefüggő, és

$$(\forall v \in V)(KiFok(v) = BeFok(v)) \quad (1)$$

7.4. tétel. A $G = (V, E)$ irányított gráfban akkor és csak akkor van Euler-séta, ha van benne Euler-kör, vagy pontosan egy olyan $a \in V$ és $b \in V$ pont van, hogy

$$KiFok(a) = BeFok(a) + 1 \quad (2)$$

$$KiFok(b) + 1 = BeFok(b) \quad (3)$$

$$(\forall v \in V)(v \neq a \wedge v \neq b \Rightarrow KiFok(v) = BeFok(v)) \quad (4)$$

és G bármely pontja elérhető a -ból.

7.5. tétel. A $G = (V, E)$ irányítatlan gráfban akkor és csak akkor van Euler-kör, ha összefüggő, és

$$(\forall v \in V)(Fok(v) \text{ páros}) \quad (5)$$

7.6. tétel. A $G = (V, E)$ irányított gráfban akkor és csak akkor van Euler-séta, ha van benne Euler-kör, vagy összefüggő és pontosan egy olyan $a \in V$ és $b \in V, a \neq b$ pont van, hogy

$$Fok(a) \text{ páratlan} \quad (6)$$

$$Fok(b) \text{ páratlan} \quad (7)$$

$$(\forall v \in V)(v \neq a \wedge v \neq b \Rightarrow Fok(v) \text{ páros}) \quad (8)$$

```

1  Program EulerUt; {Euler-út keresés irányított gráfra}
2  Const
3      MaxP=10000;      {a pontok max. száma}
4      MaxM=100000;    {az élek max. száma}
5      Null=0;
6  Type
7      PontTip=0..MaxP;
8      Cella=Record
9          Pont: PontTip;
10         csat: 0..MaxM;
11     End;
12     Graf=Array [1..MaxP] Of 0..MaxM; {a gráfábrázolás típusa }
13     Uttip=array [0..MaxM] of 0..MaxP;

```

```

14 Var
15   N,                {a pontok száma }
16   E: Longint;      {az élek száma}
17   G: Graf;
18   El: Array [1..MaxP] of Cella; {az első aktív él: p->q=El[G[p]].pont}
19   Szabad: longint; {az első szabad cella}
20   Ut: UtTip;       {a séta pontjai }
21   KiFok, BeFok: Array [1..MaxP] of longint;

22 Procedure Beolvas; Var BeF: Text;
23   u, v: PontTip;
24   i: Longint;
25   Guv: longint;
26 Begin
27   Assign(BeF, 'eulerkor.be'); Reset(BeF);
28   ReadLn(BeF, N, E);
29   For u:=1 To N Do Begin      {Inicializálás}
30     G[u]:= Null;
31     KiFok[u]:=0;
32     BeFok[u]:=0;
33   End;
34   For i:=1 To E Do Begin    {az input beolvasása}
35     ReadLn(BeF, u, v);
36     Guv:=szabad; inc(szabad);
37     El[Guv].pont:=v;
38     El[Guv].csat:=G[u];
39     G[u]:=Guv;
40     Inc(KiFok[u]);
41     Inc(BeFok[v]);
42   End{for i};
43
44   Close(BeF);
45 End{Beolvas};

46 Procedure KiIr; Var
47   KiF: Text;
48   i: longint;
49 Begin
50   Assign(KiF, 'eulerkor.ki'); Rewrite(KiF);
51   for i:=1 to E do
52     write(KiF, Ut[i], ' ');
53   Writeln(kiF);
54   Close(KiF);
55 End{KiIr};

56 Procedure VeremBe(p: word);
57 begin end;
58 Procedure Torol; begin end;
59 function Teteje: word;
60 begin end;
61 function NemUres: boolean;
62 begin end;
63 Procedure Letesit(v: longint);
64 begin end;

65 procedure EulerUt(
66   var G: graf;           //a gráf
67   n: word; m: longint; //a pontok és élek száma

```

```

68     a: PontTip;           //van a-ból induló Euler-út
69     var Ut: UtTip);      //az út
70 var p: PontTip;
71     k: word;
72 begin
73     Letesit(m); {verem létesítés}
74     VeremBe(a);
75     k:=m;
76     while NemÜres do begin {amíg a verem nem üres}
77         p:=Teteje;
78         if G[p]=Null then begin {nincs már p-ből benemjárt él}
79             Torol;           {p-t töröljük a veremből}
80             Ut[k]:=p;
81             dec(k);
82         end else begin      {tovább a p->q élen}
83             VeremBe(El[G[p]].pont);
84             G[p]:=El[G[p]].csat;
85         end;
86     end {while};
87 end {EulerUt};

88 begin {Program}
89     Beolvas;
90     EulerUt(G,n,E,Ut);
91     KiIr;
92 end.

```

Euler-kör keresés irányítatlan gráfra

```

1 Program EulerKor; {Euler-út(kör) keresés irányítatlan gráfra}
2 Const
3     MaxP=10000;      {a pontok max. száma}
4     MaxM=100000;    {az élek max. száma}
5     Null=0;
6 Type
7     PontTip=0..MaxP;
8     Cella=Record
9         Pont: PontTip;
10        masik: longint;
11        aktiv: boolean;
12        csat: 0..MaxM;
13    End;
14 Graf=Array [1..MaxP] Of 0..MaxM; {a gráfábrázolás típusa }
15 Utip=array [0..MaxM] of 0..MaxP;
16 Var
17     N,E: Longint;      {a pontok száma, az élek száma }
18     G: Graf;
19     El: Array [1..MaxP] of Cella; {az első aktív él: p->q=El[G[p]].pont}
20     Szabad: longint;   {az első szabad cella}
21     Ut: UtTip;        {a séta pontjai }
22     Fok: Array [1..MaxP] of longint;
23     a: PontTip;

24 Procedure Beolvas; Var BeF: Text;
25     u,v: PontTip;
26     i, Guv,Gvu: longint;
27 Begin
28     Assign(BeF, 'eulerkor.be'); Reset(BeF);
29     ReadLn(BeF, N, E);

```