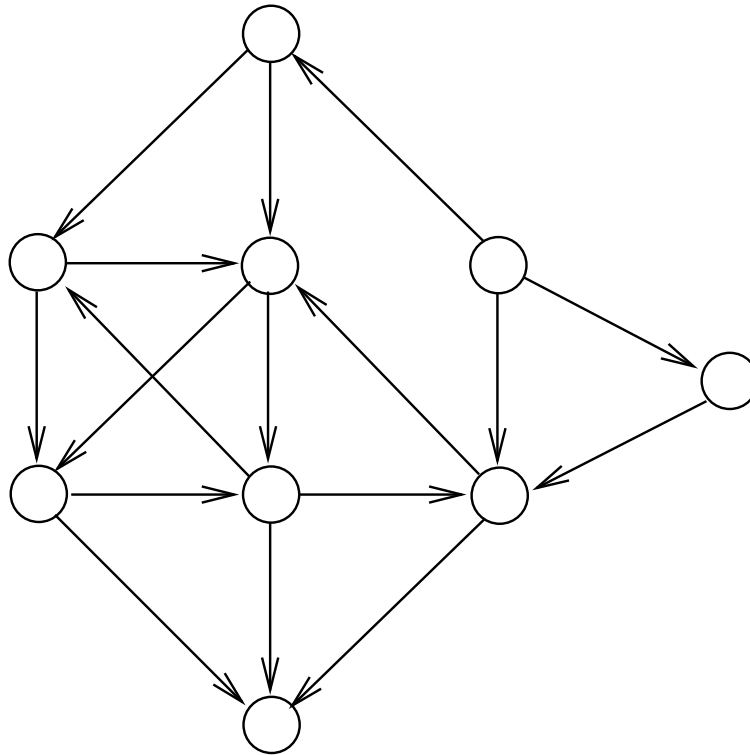


11. Mélységi keresés



1. ábra. A példa irányított gráf

```

Procedure MelyKeres(Const G: Graf.Tipus;
                    Var Apa: Fuggveny1.Tipus;
                    Var D  : Fuggveny2.Tipus;
                    Var F  : Fuggveny3.Tipus );
{
  Bemenet: G=(V,E) gráf
  Kimenet: Apa, D, F függvények,
  D(p): a p pont elérési ideje
  F(p): a p pont elhagyási ideje
  Apa(p): (Apa(p),p) feszítőfa él
}
Const
  Null=???  (:Graf.PontTip);
Type
  Paletta=(Feher, Szurke, Fekete);
Var
  Szin: Fuggveny4;{Szin: V->Paletta}
  Ido:Longint;      {az elérési-elhagyási idő számításához}
  P:Graf.PontTip;

  Procedure MelyBejar(P : Graf.Ponttip);
  {Bejárja a gráfban a P pontból elérhető, még érintetlen pontokat}
  {P még érintetlen, azaz Szin(P)=Feher}
  Var
    Q : Graf.Ponttip;
  Begin{MelyBejar}
    Szin[P]:=Szurke;           {P az új aktív pont}
    Inc(Ido);
    D[P]:=Ido;                 {P elérési időjének beállítása}
    For Q In Ki(G,P) Do       {minden P->Q élre}
      If Szin[Q]=Feher Then Begin {Q még érintetlen}
        Apa[Q]:=P;             {a (P,Q) él felvétele a feszítőfába}
        MelyBejar(Q);          {a Q-ból elérhető pontok bejárása}
      End
    Szin[P]:=Fekete;           {P bevégzett lett}
    Inc(Ido);
    F[P]:=Ido;                 {P elhagyási időjének beállítása}
  End{MelyBejar};

Begin{MelyKeres}
  Letesit(Apa);
  Letesit(D);
  Letesit(F);
  Letesit(Szin);
  For P In V Do Do Begin
    Apa[P]:=Null;
    Szin[P]:=Feher;
  End;
  Ido:= 0;
  For P In V Do
    If Szin[P]= Feher Then
      MelyBejar(P);

  Megszuntet(Szin);
End{MelyKeres};

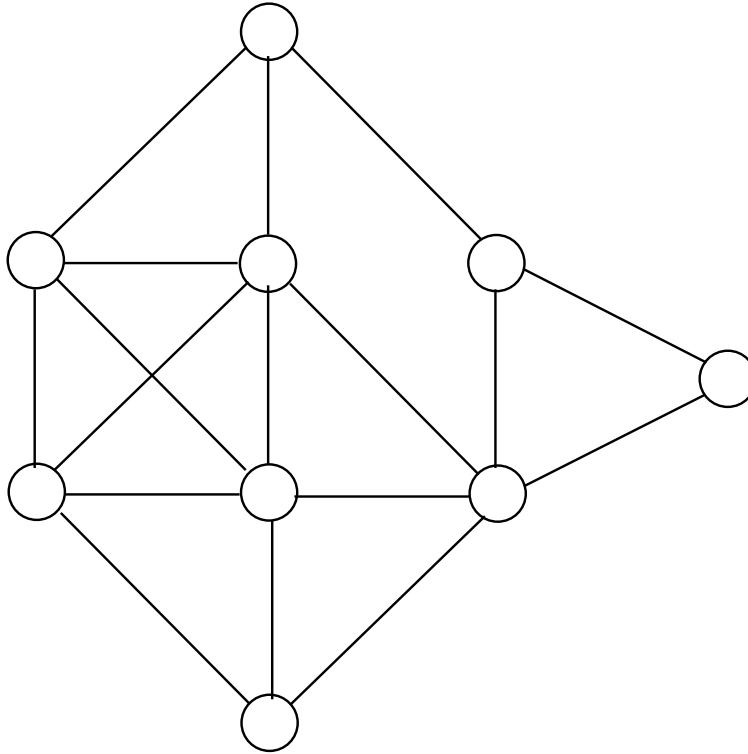
```

A MELYKERES algoritmus futási ideje $\Theta(V + E)$. (Feltéve, hogy

For $Q \in K_i(G, P)$

iteráció lineáris időben ($\Theta(K_i(G, P))$) megvalósítható.)

Mélységi bejárás irányítatlan gráfra.



2. ábra. Példa irányítatlan gráf

11.1. A mélységi bejárás tulajdonságai

Mélységi feszítő erdő (MFE).

$$MFE = (V, F) : F = \{(p, q) : \text{Apa}(q) = p\}$$

A MELYSKERES algoritmus által kiszámított d és f függvényekre teljesül:

$$(\forall u \in V)(d(u) < f(u))$$

$$(\forall u \neq v \in V)(d(u) \neq d(v) \wedge f(u) \neq f(v))$$

11.1. tétel. (Zárójelzési tétel) Mélységi keresést alkalmazva a $G = (V, E)$ gráfra, a következő három feltétel közül pontosan az egyik teljesül minden u és v pontra:

1. A $[d(u), f(u)]$ és $[d(v), f(v)]$ intervallumok diszjunktak és egyik sem leszármazottja a másiknak a MFE-ben.
2. A $[d(v), f(v)] \subseteq [d(u), f(u)]$ és v leszármazottja u -nek a MFE-ben.
3. A $[d(u), f(u)] \subseteq [d(v), f(v)]$ és u leszármazottja v -nek a MFE-ben.

Bizonyítás. Legyen $d(u) < d(v)$. Két eset lehet:

a.) $d(v) < f(u)$

Ekkor v elérésekor u színe szürke volt, tehát v leszármazottja MFE-ben u -nak. Továbbá, v -t előbb elhagyjuk, mielőtt visszatérnénk u -hoz, tehát $f(v) < f(u)$. Tehát: $d(v) < d(v) < f(v) < f(u)$

b.) $d(v) \geq f(u)$

Ekkor $d(u) < f(u) < d(v) < f(v)$, tehát a két intervallum diszjunkt. Ebből következik, hogy mind u , mind v elérésekor a másik nem lehetett szürke, tehát nem leszármazottjai egymásnak MFE-ben. ■

11.2. következmény. A v pont akkor és csak akkor leszámazottja az u pontnak MFE-ben, ha $d(u) < d(v) < f(v) < f(u)$

11.3. tétel. (Fehér út tétel) Minden v pont akkor és csak akkor leszámazottja az u pontnak MFE-ben, ha a $d(u)$ időben (u eléré-
sekor) van csupa fehér pontokon át haladó $u \rightsquigarrow v$ út G -ben.

Bizonyítás. $\Rightarrow$: Tfh. $u \rightsquigarrow v$ a MFE-ben. Legyen w ezen út egy tetszőleges pontja. Az 11.2. következmény szerint $d(u) < d(w)$, tehát w fehér a $d(u)$ időben.

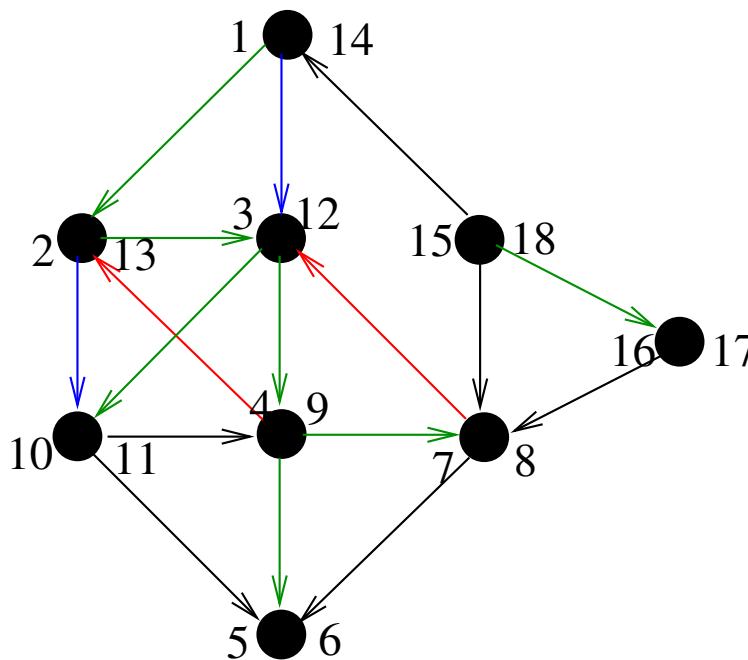
$\Leftarrow$: Tfh. van olyan $u \rightarrow v_1 \rightarrow \dots \rightarrow v_k = v$ út, hogy minden v_i fehér a $d(u)$ időben, de v nem lesz leszámazottja u -nak MFE-ben. Feltessük, hogy $\forall i < k$ -ra v_i leszámazottja lesz u -nak MFE-ben. Mivel v_{k-1} leszámazottja u -nak, ezért 11.2. következmény miatt $f(v_{k-1}) < f(u)$. De $d(u) < d(v) < f(v_{k-1})$, így $d(u) < d(v) < f(v_{k-1}) < f(u)$, tehát az 11.2. következmény miatt v leszámazottja u -nak MFE-ben. ■

Élek osztályozása

1. **Faél:** $(u, v) \in V$ faél, ha bekerül a MFE élei közé, azaz $\text{Apa}(v) = u$.
2. **Visszaél:** $(u, v) \in V$ visszaél, ha u leszámazottja v -nek a MFE-ben.
3. **Előreél:** $(u, v) \in V$ visszaél, ha v leszámazottja u -nek a MFE-ben és nem faél.
4. **Keresztél:** Minden más esetben $(u, v) \in V$ keresztél.

Állítás

1. $(u, v) \in V$ akkor és csak akkor faél, ha az $u \rightarrow v$ él vizsgálatakor v színe fehér.
2. $(u, v) \in V$ akkor és csak akkor visszaél ha az $u \rightarrow v$ él vizsgálatakor v színe szurke.
3. $(u, v) \in V$ akkor és csak akkor előreél ha az $u \rightarrow v$ él vizsgálatakor v színe fekete és $d(u) < d(v)$.
4. $(u, v) \in V$ akkor és csak akkor keresztél ha az $u \rightarrow v$ él vizsgálatakor v színe fekete és $d(u) > d(v)$.



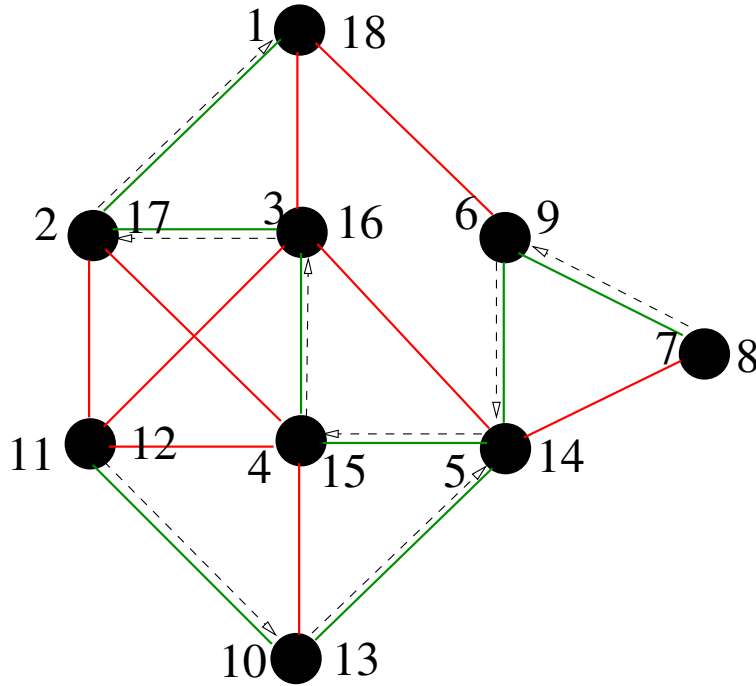
3. ábra. Az élek osztályozása. zöld:faél, piros:visszaél, kék:előreél, fekete:keresztél.

11.4. tétel. Ha G irányítatlan gráf, akkor bármely mélységi keresésre minden éle vagy fa-él, vagy vissza-él.

Bizonyítás. Tfh. $d(u) < d(v)$. Mivel $v \in Ki(G, u)$, ezért valamikor végrehajtódik az $u \rightarrow v$ él vizsgálata, és ekkor $Szin(u) = Szurke$. Milyen lehet v színe?

1. $Szin(v) = Feher$: Ekkor (u, v) faél lesz.
2. $Szin(v) = Szurke$: Ekkor (u, v) visszaél lesz.
3. $Szin(v) = Fekete$: Nem lehet, mert ez azt jelentené, hogy $f(v) < d(u)$.

■



4. ábra. Az élek osztályozása irányítatlan gráf mélységi bejárására. zöld:faél, piros:visszaél.

Topologikus rendezés

Def. Egy $G = (V, E)$ irányított gráf topologikus rendezésén a V pontjainak egy olyan $\langle v_1, v_2, \dots, v_n \rangle$ ($n = |V|$) felsorolását értjük, amelyre teljesül:

Minden $(u, v) \in E$ éltre, u előbb áll a felsorolásban, mint v .

11.5. lemma. A $G = (V, E)$ irányított gráfnak akkor és csak akkor van topologikus rendezése, ha G körmentes.

11.6. lemma. A $G = (V, E)$ irányított gráfban akkor és csak akkor van kör, ha van visszaéle.

Bizonyítás. $\Rightarrow$: Tfh. $u \rightarrow v$ visszaél. Ekkor $v \rightsquigarrow u \rightarrow v$, tehát van kör G -ben.

$\Leftarrow$: Tfh. $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_{k-1} \rightarrow v_k = v_1$ kör, továbbá a kör pontjai közül v_1 -et érjük el először a mélységi bejárás során. Tehát $d(v_1)$ időben van csupa fehér ponton át haladó $v_1 \rightsquigarrow v_k$ út. A fehér út tétel miatt v_k leszármazottja lesz a v_1 -nek a MFE-ben, tehát $f(v_k) \rightarrow v_1$ visszaél lesz. ■

11.7. tétel. Ha a G irányított gráf körmentes, akkor pontjainak minden olyan $\langle v_1, \dots, v_n \rangle$ felsorolása, amelyre

$$f(v_i) > f(v_{i+1}); i = 1, \dots, n-1$$

G egy topologikus rendezése lesz bármely mélységi bejárással kiszámított f elhagyási függvényre.

Bizonyítás. Legyen $u \rightarrow v \in E$ tetszőleges él.

a.) $d(u) < d(v)$, azaz u -t előbb érem el, mint v -t. Ekkor a fehér út tétel miatt v leszármazottja lesz u -nak a MFE-ben, tehát $d(u) < d(v) < f(v) < f(u)$.

b.) $d(u) > d(v)$, azaz v -t előbb érem el, mint u -t. Mivel nincs $v \rightsquigarrow u$ út G -ben, ezért v -t befejezem, mielőtt u -t elérném, tehát $f(v) < d(u) < f(u)$. ■

```

{Globális programelemek a Topologikus rendezéshez;}
Const
    MaxN=???; {a gráf pontjainak max. száma}
Type
    PontTip=1..MaxN;
    Vektor=Array[PontTip] Of 0..MaxN;

Procedure TopRend(Const G: Graf;
                  Var S: Vektor);

Var
    Szin: Array[PontTip] of (Feher,Szurke,Fekete);
    VanKor:Boolean;
    p:1..MaxN;
    SorInd:Longint;

    Procedure MelyBejar(P : PontTip);
    Var
        Q:Graf.Ponttip;
    Begin{MelyBejar}
        Szin[P]:=Szurke;
        For Q In Ki(G,P) Do Begin
            If Szin[Q]=Feher Then
                MelyBejar(Q);
            Else If Szin[Q]=Szurke Then Begin{kör!}
                VanKor=True;
                Exit
            End;
            If VanKor Then Exit;
        End{for Q}
        Szin[P]:=Fekete;
        Dec(SorInd);
        S[SorInd]:=P;
    End {MelyBejar};

Begin{Toprend}
    For P:=1 To N Do
        Szin[P]:=Feher;
    VanKor:=False;
    SorInd:=N+1;
    For P:=1 To N Do Begin
        If Szin[P]=Feher Then
            MelyBejar(P);
        If VanKor Then Begin
            S[1]:=0;
            Break
        End;
    End{for P};

End {TopRend};

```

A TOPREND algoritmus futási ideje $\Theta(V + E)$. (Feltéve, hogy

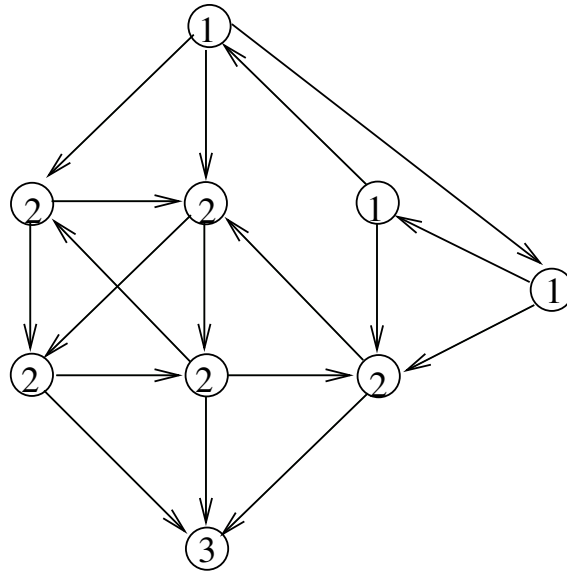
For Q In Ki(G,P)

iteráció lineáris időben ($\Theta(Ki(G,P))$) megvalósítható.)

11.2. Erősen összefüggő komponensek

Def. $u \sim v$ ha $u \rightsquigarrow v$ és $v \rightsquigarrow u$

Def. Egy u pontot tartalmazó erősen összefüggő komponens: $C(u) = \{v \in V : u \sim v\}$



5. ábra. Erősen összefüggő komponensek

Def. A $G = (V, E)$ gráf transzponáltja: $G^T = (V, E^T)$; ahol $E^T = \{(p, q) : (q, p) \in E\}$

Elvi algoritmus

1. Számítsuk ki a MELYKERES algoritmussal az $f(u)$ elhagyási értékeket.
2. A G^T transzponált gráfra alkalmazzuk a MELYKERES eljárást úgy, hogy a pontokra a MELYBEJÁR eljárást f szerint csökkenő sorrendben hívjuk.
3. A 2. pontban az egy mélységi feszítőfába kerülő pontok alkotnak egy erősen összefüggő komponenst.

```

Procedure GEOK(Const G: Graf.Tipus;
               Var H: Halmazok.Tipus);
Var
  V      : Verem.Tipus;
  Bejart: Halmaz.Tipus;
  P      : Graf.Ponttip;
  PNev  : Halmazok.NevTip;

```

```

Procedure MelyBejar(P : Graf.Ponttip);
Var
  Q : Graf.Ponttip;
Begin{MelyBejar}
  Bovit(Bejart, P);
  For Q In Ki(G, P) Do
    If Not Eleme(Bejart, Q) Then
      MelyBejar(Q);
  Verembe(V, P);
End{MelyBejar};

```

```

Procedure MelyBejarT(P, Pnev: Graf.Ponttip);
Var
  Q : Graf.Ponttip;
Begin{MelyBejarT}

```

```

    Bovit(Bejart,P);
    For Q In Be(G,P) Do Begin
        If Not Eleme(Bejart,Q) Then Begin
            Halmazok.Bovit(H,Pnev,Q);
            MelyBejarT(Q,Pnev);
        End;
    End{MelyBejarT};

Begin {GEOK}
    Halmazok.Letesit(H);
    Halmaz.Letesit(Bejart);
    Verem.Letesit(V);
    For P In V Do
        If Not Eleme(Bejart,P) Then
            MelyBejar(P);

    Uresit(Bejart);
    While Not Urese(V) Do Begin
        { A transzponált gráf mélységi bejárása csökkenő F érték,
          azaz a V veremben levő sorrend szerint }
        VeremBol(V,P);
        If Not Eleme(Bejart,P) Then Begin
            Bovit(H, P, P);      { új komponens }
            MelyBejarT(P,P);
        End;
    End;

    Halmaz.Megszuntet(Bejart);
    Verem.Megszuntet(V);
End{GEOK};

```

11.8. lemma. Legyen C és C' a $G = (V, E)$ irányított gráf két különböző erősen összefüggő komponense. Továbbá, $u, v \in C$ és $u', v' \in C'$. Ha létezik $u \rightsquigarrow u'$ út, akkor nem létezhet $v' \rightsquigarrow v$ út.

Terjesszük ki a d és f függvényeket V részhalmazaira, $U \subseteq V$:

$$d(U) = \min_{u \in U} \{d(u)\}$$

$$f(U) = \max_{u \in U} \{f(u)\}$$

11.9. lemma. Legyen C és C' a $G = (V, E)$ irányított gráf két különböző erősen összefüggő komponense. Ha létezik olyan $u \rightarrow v$ él, hogy $u \in C$ és $v \in C'$, akkor $f(C) > f(C')$.

Bizonyítás. a.) $d(C) < d(C')$. Legyen $x \in C$, amelyre $d(x)$ minimális, azaz az elsőnek elért pont C -ben. Tehát bármely $w \in C'$ pontra a $d(x)$ időben létezik csupa fehér pontokon át haladó

$x \rightsquigarrow u \rightarrow v \rightsquigarrow w$ út. Tehát a fehér út tétel miatt C és C' minden pontja leszármozottja lesz x -nek a MFE-ben. Így a 11.2. következmény miatt $f(x) = f(C) > f(C')$.

b.) $d(C) > d(C')$. Legyen $y \in C'$, amelyre $d(y)$ minimális, azaz az elsőnek elért pont C' -ben. Mint az előző esetben, a fehér út tétel miatt $f(y) = f(C')$. Mivel van $u \rightarrow v$ él, ezért az előző lemma miatt nem lehet C' egyetlen pontjából sem eljutni C -beli pontba, így y -ból sem. Tehát $f(y)$ időben C minden pontja fehér, tehát $f(w) > f(y)$ minden $w \in C$. Tehát $f(C) > f(C')$. ■

11.10. következmény. Legyen C és C' a $G = (V, E)$ irányított gráf két különböző erősen összefüggő komponense. Ha létezik olyan $u \rightarrow v \in G^T$ él, hogy $u \in C$ és $v \in C'$, akkor $f(C) < f(C')$.

Bizonyítás. $u \rightarrow v \in G^T \Rightarrow v \rightarrow u \in E$. Mivel G és G^T erősen összefüggő komponensei megegyeznek, így az előző lemma miatt $f(C) < f(C')$. ■

11.11. tétel. Az EOK algoritmus helyes.

Bizonyítás. Legyenek G E.Ö. komponensei:

$C(r_1), \dots, C(r_k)$, hogy $f(r_1) > f(r_2) > \dots > f(r_k)$ és
 $f(r_i) = f(C(r_i)), i = 1, \dots, k$

Legyen $F(r_i), (i = 1, \dots, k)$ a G^T bejárásával kapott r_i -gyökerű mélységi feszítőfa pontjainak halmaza.

Megmutatjuk, hogy $C(r_i) = F(r_i)$. A bizonyítást i -szerinti indukcióval végezzük. Tfh. az első $i - 1$ komponensre az állítás igaz.

$C(r_{i-1})$ G^T bejárása után MelyBejarT(r_i) hívás következik.

$f(r_i) = f(C(r_i))$, ezért a fehér út tétel miatt (G^T -ben) $C(r_i)$ minden pontja bekerül $F(r_i)$ -be. Továbbá, a 11.10. következmény miatt minden él, amely $C(r_i)$ -ből kivezet, csak olyan C' komponensbeli pontba vezethet, amelyre $f(C') > f(C(r_i))$, tehát ez már korábban bejárt komponens. ■

11.3. Euler-kör és Euler-út

A $\langle p_1, p_2, \dots, p_k \rangle$ sétát a $G = (V, E)$ (irányított, vagy irányítatlan) gráf Euler-útjának nevezzük, ha a séta a gráf minden élét pontosan egyszer tartalmazza. Azaz $(\forall (u, v) \in E)(\exists ! i)(u = p_i \wedge v = p_{i+1})$.

A $\langle p_1, p_2, \dots, p_k \rangle$ séta Euler-kör, ha Euler-út és $p_1 = p_k$.

11.12. tétel. A $G = (V, E)$ irányítatlan gráfban akkor és csak akkor van Euler-út, ha G összefüggő, és vagy minden pont páros fokszámú, vagy pontosan két olyan pontja van, amelyek páratlan fokúak.

11.13. tétel. A $G = (V, E)$ irányítatlan gráfban akkor és csak akkor van Euler-kör, ha G összefüggő, és minden pont páros fokszámú.

11.14. tétel. A $G = (V, E)$ irányított gráfban akkor és csak akkor van Euler-út, ha G összefüggő, és vagy minden $v \in V$ pontra $KiFok(G, v) = BeFok(G, v)$, vagy van $a, b \in V$, hogy $BeFok(G, a) + 1 = KiFok(G, a)$ és $KiFok(G, b) + 1 = BeFok(G, b)$ és minden $v \neq a, b \in V$ pontra $KiFok(G, v) = BeFok(G, v)$.

11.15. tétel. A $G = (V, E)$ irányított gráfban akkor és csak akkor van Euler-kör, ha G összefüggő, és minden $v \in V$ pontra $KiFok(G, v) = BeFok(G, v)$.

```

Program EulerKor; {Euler-kör keresés irányított gráfra}
Const
    MaxP=100000;           {a pontok max. száma}
Type
    Pont=0..MaxP;
    Lanc=^Cella;
    Cella=Record
        P:Pont;
        csat:Lanc
    End;
    Graf=Array[1..MaxP] Of Lanc; { a gráfábrázolás típusa }
Var
    N,                      { a pontok száma }
    E:Longint;              { az élek száma }
    G:Graf;
    KiFok, BeFok:Array[1..MaxP] of 0..MaxP;
    El:Array[1..MaxP] of Lanc; { El[p] az első aktív él }
    Kor:Lanc;               { a körséta pontjai }

Procedure Beolvas; Var InF:Text;
    u,v:Pont;
    i:Longint;
    Guv:Lanc;
Begin
    Assign(InF, 'eulerkor.be'); Reset(InF);
    ReadLn(InF, N, E);
    For u:=1 To N Do Begin      {Inicializálás}

```

```

    G[u]:=Nil;
    KiFok[u]:=0;
    BeFok[u]:=0;
End;
For i:=1 To E Do Begin      {az input beolvasása}
    ReadLn(InF, u,v);
    New(Guv);
    Guv^.p:=v;
    Guv^.csat:=G[u];
    G[u]:=Guv;
    Inc(KiFok[u]);  Inc(BeFok[v]);
End{for i};

Close(InF);
End{Beolvas};

Procedure KiIr; Var
    KiF:Text;
    L:Lanc;
Begin
    Assign(KiF,'eulerkor.ki'); Rewrite(KiF);
    L:=Kor;
    While L<> Nil Do Begin
        Write(KiF, L^.P,' ');
        L:=L^.csat;
    End;
    Writeln(kiF);
    Close(KiF);
End{KiIr};

Procedure EulerKorEpit;
Var
    i:Longint;
    VanKor:Boolean;
    EInd:Longint;

Procedure Seta(p:Pont);
Var q:Pont;
    L:Lanc;
Begin
    While El[p]<>Nil Do Begin {amíg van p->q benemjárt él}
        q:=El[p]^P;
        El[p]:=El[p]^csat;    {q->p passzívá tétele}
        Seta(q);
    End{while};              {tovább a p->q élen}

    New(L);
    L^.P:=p; L^.csat:=Kor;    {p bevétele a sétába}
    Kor:=L;
    Inc(EInd);                {a bejárt élek számának növelése}
End{Seta};

Begin{EulerKorEpit}
    VanKor:=True;
    For i:=1 To N Do Begin
        If (KiFok[i]=0) Or (KiFok[i]<>BeFok[i]) Then

```

```

        VanKor:=False;
        El[i]:=G[i];           {nincs Euler-kör}
    End;
    Kor:=Nil;
    EInd:=0;
    If VanKor Then Begin
        Seta(1);               {Eurel-kör építés}

        VanKor:=EInd=E+1;      {összefüggő-e a gráf?}
    End;
    If Not VanKor Then Kor:=Nil;
    KiIr;
End{EulerKorEpit};

Begin{Program}
    Beolvas;

    EulerKorEpit;
End.

```

EULERKOREPIT algoritmus futási ideje: $\Theta(V + E)$

Euler-kör keresés irányítatlan gráfban.

```

Program EulerKor; {Euler-kör keresés irányítatlan gráfra} Const
    MaxP=100000;           {a pontok max. száma}
Type
    Pont=0..MaxP;
    Lanc=^Cella;
    Cella=Record
        P:Pont;
        Aktiv:Boolean;
        Veg,csat:
            Lanc
        End;
    Graf=Array[1..MaxP] Of Lanc; { a gráfábrázolás típusa }
Var
    N,                      { a pontok száma }
    E:Longint;              { az élek száma }
    G:Graf;                 { a gráf }
    Fok:Array[1..MaxP] of 0..MAXP; { fokszám }
    El:Array[1..MaxP] of Lanc; { El[p] az első aktív él }
    Kor:Lanc;               { a körséta pontjai }

Procedure Beolvas; Var InF:Text;
    u,v:Pont;
    i:Longint;
    Guv,Gvu:Lanc;
Begin
    Assign(InF, 'eulerkor.be'); Reset(InF);
    ReadLn(InF, N, E);
    For u:=1 To N Do Begin {Inicializálás}
        G[u]:=Nil;
        Fok[u]:=0;
    End;
    For i:=1 To E Do Begin {az input beolvasása}

```

```

    ReadLn(InF, u,v);
    New(Guv); New(Gvu);
    Guv^.p:=v; Guv^.Aktiv:=True; Guv^.Veg:=Gvu;
    Gvu^.p:=u; Gvu^.Aktiv:=True; Gvu^.Veg:=Guv;
    Guv^.csat:=G[u]; Gvu^.csat:=G[v];
    G[u]:=Guv; G[v]:=Gvu;
    Inc(Fok[u]); Inc(Fok[v]);
End{for i};

Close(InF);
End{Beolvas};

Procedure KiIr; Var
    KiF:Text;
    L:Lanc;
Begin
    Assign(KiF,'eulerkor.ki'); Rewrite(KiF);
    L:=Kor;
    While L<> Nil Do Begin
        Write(KiF, L^.P,' ');
        L:=L^.csat;
    End;
    Writeln(kiF);
    Close(KiF);
End{KiIr};

Procedure EulerKorEpit;
Var
    i:Longint;
    VanKor:Boolean;
    EInd:Longint;

Procedure Seta(p:Pont);
{Global: G, El}
Var q:Pont;
    L,Lq:Lanc;
Begin
    While El[p]<>Nil Do Begin {amíg van p->q benemjárt él}
        If El[p]^Aktiv Then Begin{p->El[p]^P aktív él}
            q:=El[p]^P;
            El[p]^Aktiv:=False; {p->q passzívá tétele}
            Lq:=El[p]^Veg;
            Lq^.Aktiv:=False; {q->p passzívá tétele}
            El[p]:=El[p]^csat;
            Seta(q); {tovább a p->q élen}
        End Else {p->El[p]^P passzív él}
            El[p]:=El[p]^csat;
        End{while};

        New(L);
        L^.P:=p; L^.csat:=Kor; {p bevétele a sétába}
        Kor:=L;
        Inc(EInd); {a bejárt élek sz. növelése}
    End{Seta};

Begin{EulerKorEpit}

```

```

VanKor:=True;
For i:=1 To N Do Begin
  If (Fok[i]=0)Or Odd(Fok[i]) Then
    VanKor:=False;          {nincs Euler-kör}
  El[i]:=G[i];
End;
Kor:=Nil;
EInd:=0;
If VanKor Then Begin
  Seta(1);                  {Eurel-kör építés}

  VanKor:=EInd=E+1;         {összefüggő-e a gráf?}
End;
If Not VanKor Then Kor:=Nil;
KiIr;
End{EulerKorEpit};

Begin{Program}
  Beolvas;

  EulerKorEpit;
End.

```

EULERKOREPIT algoritmus futási ideje: $\Theta(V + E)$