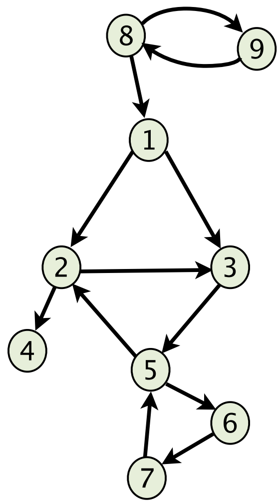
Mélységi keresés kibővítve

Az algoritmus elemzéséhez és egyes alkalmazásainak helyességének bizonyításához hasznos, ha minden pontra definiáljuk a d elérési és f elhagyási idejét.

```
MelykeresBovit(G)
 for(u in G)
   {szin(u):=feher
    Apa(u):=0}
 ido:=0
 for(u in G)
    {if szin(u)=feher
      then MBejarBovit(u)}
MBejarBovit(u)
 szin(u):=szurke
 ido:=ido+1
 d(u):=ido
 for(v in Kiel(G,u))
    {if szin(v)=feher
       Then {Apa(v):=u
             MBejarBovit(v)}}
 szin(u):=fekete
 ido:=ido+1
 f(u):=ido
```

- MBejarBovit(1)
- szín(1):=szürke
- d(1)=1
- Apa(2):=1
- MBejarBovit(2)
- szín(2):=szürke
- d(2)=2
- Apa(3):=2
- MBejarBovit(3)
- szín(3):=szürke
- d(3)=3
- Apa(5):=3
- MBejarBovit(5)
- szín(5):=szürke



1. ábra.

- d(5)=4
- Apa(6):=5
- MBejarBovit(6)
- szín(6):=szürke
- d(6)=5
- Apa(7):=6
- MBejarBovit(7)
- szín(7):=szürke
- d(7)=6
- MBejarBovit(7) vége, mert 5 már nem fehér
- szín(7):=fekete
- f(7)=7
- MBejarBovit(6) vége
- szín(6):=fekete

- f(6)=8
- MBejarBovit(5) vége
- szín(5):=fekete
- f(5)=9
- MBejarBovit(3) vége
- szín(3):=fekete
- f(3)=10
- Apa(4):=2
- MBejarBovit(4)
- szín(4):=szürke
- d(4)=11
- MBejarBovit(4) vége
- szín(4):=fekete
- f(4)=12
- MBejarBovit(2) vége
- szín(2):=fekete
- f(2)=13
- MBejarBovit(1) vége
- szín(1):=fekete
- f(1)=14
- MBejarBovit(8)
- szín(8):=szürke
- d(8)=15
- Apa(9):=8
- MBejarBovit(9)
- szín(9):=szürke
- d(9)=16
- MBejarBovit(9) vége
- szín(9):=fekete

- f(9)=17
- MBejarBovit(8) vége
- szín(8):=fekete
- f(8)=18

1. táblázat. A keresés során kapott értékek

pont	1	2	3	4	5	6	7	8	9
apa	0	1	2	2	3	5	6	0	8
d	1	2	3	11	4	5	6	15	16
f	14	13	10	12	9	8	7	18	17

Az algoritmus egy mélységi feszítő erdőt (MFE) ad eredményül, a csúcshalmaz V az élek pedig $F = \{(p,q) : Apa(q) = p\}$.

Tétel (Zárójelezési tétel) Mélységi keresést alkalmazva a G = (V,E) gráfra, a következő három feltétel közül pontosan az egyike teljesül minden u és v pontra:

- A $[d(u), f(u)]$ és $[d(v), f(v)]$ intervallumok diszjunktak és az u és v pontok egyike sem leszármazottja a másiknak a MFE-ben.
- $[d(v), f(v)] \subseteq [d(u), f(u)]$ és v leszármazottja u-nak a MFE-ben.
- $[d(u), f(u)] \subseteq [d(v), f(v)]$ és u leszármazottja v-nek a MFE-ben.

Bizonyítás. Legyen d(u) < d(v).

- Ha $d(v) < f(u)$, akkor v elérésekor u színe szürke volt, tehát v leszármazottja MFE-ben u-nak. Továbbá, v-t előbb elhagyjuk, mielőtt visszatérnénk u-hoz, tehát $f(v) < f(u)$.
- Ha $d(v) > f(u)$, akkor $d(u) < f(u) < d(v) < f(v)$, tehát a két intervallum diszjunkt. Ebből következik, hogy mind u, mind v elérésekor a másik nem lehetett szürke, tehát nem leszármazottjai egymásnak MFE-ben.

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

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

Bizonyítás

- Tegyük fel, hogy van $u \rightsquigarrow v$ az MFE-ben. Ekkor ennek tetszőleges w pontjára $d(u) < d(w)$, tehát w fehér a d(u) időben.
- Tegyük fel, hogy van olyan $u = v_1 \dots v_k = v$ út, hogy minden v_i fehér a d(u) időben, de v nem lesz leszármazottja u-nak MFE-ben. Feltehetjük, hogy $\forall i < k$-ra v_i leszármazottja lesz u-nak az MFE-ben. Mivel v_{k-1} leszármazottja u-nak, ezért $f(v_{k-1}) < f(u)$. De $d(u) < d(v) < f(v_{k-1})$, így $d(u) < d(v) < f(v_{k-1}) < f(u)$, tehát v leszármazottja u-nak MFE-ben.

Élek osztályozása

- Faél: $(u,v) \in E$ faél, ha bekerül a MFE élei közé, azaz $Apa(v) = u$.
- Visszaél: $(u,v) \in E$ visszaél, ha u leszármazottja v-nek a MFE-ben.
- Előreél: $(u,v) \in E$ előreél, ha v leszármazottja u-nek a MFE-ben és nem faél.
- Keresztél: Minden más esetben $(u,v) \in E$ keresztél.

Példa az élek osztályozására

- Faélek: (1, 2), (2, 3), (3, 5), (5, 6), (6, 7), (2, 4), (8, 9)
- Visszaélek: (5, 2), (7, 5), (9, 8)
- Előreélek: (1, 3)
- Keresztélek: (8, 1).

Tulajdonságok

- $(u,v) \in E$ akkor és csak akkor faél, ha az (u,v) él vizsgálatakor v színe fehér.
- $(u,v) \in E$ akkor és csak akkor visszaél ha az (u,v) él vizsgálatakor v színe szürke.
- $(u,v) \in E$ akkor és csak akkor előre-él ha az (u,v) él vizsgálatakor v színe fekete és $d(u) < d(v)$.
- $(u,v) \in E$ akkor és csak akkor keresztél ha az (u,v) él vizsgálatakor v színe fekete és $d(u) > d(v)$.

Topologikus rendezés

Definíció Egy G =(V,E) irányított gráf topologikus rendezésén a V pontjainak egy olyan $v_1, v_2, \ldots, v_n$ $(n = |V|)$ felsorolását értjük, amelyre teljesül, hogy minden $(u,v) \in E$ élre, u előbb van a felsorolásban, mint v.

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

Lemma A G = (V,E) irányított gráfban akkor és csak akkor van kör, ha van vissza-éle.

Bizonyítás. Tegyük fel, hogy van egy (u,v) visszaél. Ekkor hozzávéve az (u,v) visszaélt az MFE-ben levő $v \rightsquigarrow u$ úthoz, egy kört kapunk G-ben.

Tegyük fel, hogy $v_1, v_2, \ldots, v_{k-1}, v_k = v_1$ egy kör, továbbá a kör pontjai közül v_1-et érjük el először a mélységi bejárás során. Ekkor a $d(v_1)$ időben van csupa fehér pontokon 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 (v_k, v_1) visszaél lesz.

Tétel Ha a G irányított gráf körmentes, akkor pontjainak minden olyan $v_1, \ldots, v_n$ felsorolása, amelyre $f(v_i) > f(v_{i+1})$, $i = 1, \ldots, n-1$ G egy topologikus rendezése lesz bármely mélységi bejárással kiszámított f elhagyási függvényre.

Bizonyítás Legyen $(u,v) \in E$ tetszőleges él.

- Ha $d(u) < d(v)$, azaz u-t előbb érjük el, mint v-t, akkor a fehér út tétel miatt v leszármazottja lesz u-nak a MFE-ben, tehát $d(u) < d(v) < f(v) < f(u)$.
- Tegyük fel, hogy $d(u) > d(v)$, azaz v-t előbb érjük el, mint u-t. Mivel nincs $v \rightsquigarrow u$ út G-ben, ezért v-t befejezzük, mielőtt u-t elérnénk, tehát $f(v) < d(u) < f(u)$.

Elvi algoritmus

- A mélységi keresés algoritmusát hajtsuk végre a gráfra.
- Az egyes csúcsok elhagyásakor beszúrjuk őket egy láncolt lista elejére.
- A csúcsok láncolt listája adja meg a rendezési sorrendet.

Az algoritmus futási ideje megegyezik a mélységi keresésével, azaz $\theta(V+E)$. (Feltéve, hogy a ki iteráció lineáris időben megvalósítható.)

Erősen összefüggő komponensek

Definíció $u \sim v$ ha van $u \rightsquigarrow v$ út és $v \rightsquigarrow u$ út is a gráfban.

Lemma A $\sim$ reláció egy ekvivalencia reláció, így egy osztályozást definiál.

Definíció A $\sim$ reláció által definiált osztályozás osztályai a gráf erősen összefüggő komponensei. Azaz egy u pontot tartalmazó erősen összefüggő komponens: $C(u) = \{v \in V : u \sim v\}$.

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

Definíció Egy G = (V,E) gráf komponensgráfjának hívjuk azt a gráfot, amelynek csúcsai az erősen összefüggő komponensek, és C komponensből akkor vezet él egy D komponensbe, ha C valamely pontjából G-ben vezet él D valamely pontjába.

Lemma A komponensgráf körmentes.
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 MBEJÁR eljárást f szerint csökkenő sorrendben hívjuk.
- 3. A 2. pontban az egy mélységi feszítőfába kerülő pontok alkotnak egy erősen összefüggő komponenst.

Megvalósítás: A 2-es pontban nem rendezést hajtunk végre az f értékek szerint, hanem a pontokat a befejezésük időpontjában berakjuk egy verembe, és abból vesszük a pontokat a második bejárásnál.

Helyesség

Terjesszük ki a d és f függvényeket V részhalmazaira, $U \subseteq V$:
$d(U) = \min_{u \in U}\{d(u)\}$
$f(U) = \max_{u \in U}\{f(u)\}$

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.

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

Bizonyítás. Ha $d(C) < d(C')$, akkor legyen $x \in C$, amelyre d(x) minimális, azaz az elsőnek elért pont C-ben. Ekkor bármely $w \in C'$ pontra a d(x) időben létezik csupa fehér pontokon át haladó $x \rightsquigarrow w$ út. Tehát a fehér út tétel miatt C és C' minden pontja leszármazottja lesz x-nek a MFE-ben, amiből adódik az állítás.

Ha $d(C) > d(C')$, akkor legyen $y \in C'$, amelyre d(y) minimális, azaz az elsőnek elért pont C'-ben. Ekkor belőle mindenki elérhető fehér pontokon C'-ben, így $f(y) = f(C')$. Mivel van (u,v) él, ezért 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 $w \in C$.

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

Helyesség

Tétel Az algoritmus az erősen összefüggő komponenseket adja meg.

Bizonyítás Legyenek G erősen összefüggő komponensei $C(r_1),\ldots,C(r_k)$, ahol $f(r_1) > f(r_2) > \cdots > f(r_k)$ és $f(r_i) = f(C(r_i))$, i = 1,...,k.

Legyen továbbá $F(r_i)$, $(i = 1,\ldots,k)$ a második bejárás során kapott r_i-gyökerű mélységi feszítőfa pontjainak halmaza. Ekkor $C(r_i) = F(r_i)$. A bizonyítást i-szerinti indukcióval adható meg. Tegyük fel, hogy az első i-1 komponensre az állítás igaz. Ekkor $C(r_i - 1)$ bejárása után egy $MBejar(r_i)$ hívás következik a transzponált gráfban. Mivel $f(r_i) = f(C(r_i))$, ezért a fehér út tétel miatt (G^T -ben alkalmazva a második bejárásra) $C(r_i)$ minden pontja bekerül $F(r_i)$-be. Másrészt a fenti következmény miatt minden él, amely $C(r_i)$-ből kivezet, csak olyan C' komponensbeli pontba vezethet, amelyre $f(C) > f(C(r_i))$, és ezeket már korábban bejárta az algoritmus.

Legrövidebb utak súlyozott gráfokban

A feladat egy súlyozott gráfban egy adott pontból kiinduló legrövidebb utak megkeresése. Az input a súlyozott gráf és a kiindulási s pont. Outputként egy legrövidebb utak fáját adunk vissza, egy Apa függvény által, továbbá a legrövidebb utak hosszait egy d függvény által. A feladatot nemnegatív élsúlyok esetén a következő Dijsktra algoritmussal oldhatjuk meg. A pontokat egy d érték szerinti Q módosítható prioritási sorban tároljuk.

Dijskstra algoritmusa

```
Kezd(G,s)
 for(v in V)
    {d(v):=INF
     Apa(v):=0
     Kesz(v):=0}
 d(s):=0

Kozelit(G,u,v,Q)
 if d(v)>d(u)+c(u,v)
    then {d(v):=d(u)+c(u,v)
          Modosit(Q,v)
          Apa(v):=u}

Dijkstra(G,s)
 Kezd(G,s)
 Letesit(Q: ModPrisor)
 for(v in V)
    SorBa(Q,v)
 while(Elemszam(Q)>0)
    {SorBol(Q,u)
    Kesz(u):=1
    for(v in KiEl(G,u))
        {if Kesz(v)=0
           then Kozelit(u,v)}}
```

Példa

Hajtsuk végre az 1 pontból a Dijkstra algoritmust az alábbi gráfra. (A mátrixban a c_{ij} érték az (i,j) él hossza, ∞ ha nincs él.)

$$\begin{pmatrix} \infty & 2 & \infty & 1 & 8 & \infty \\ 2 & \infty & 1 & 4 & 1 & 5 \\ 2 & \infty & 2 & 1 & 1 & 7 \\ \infty & 1 & 4 & \infty & 2 & 7 \\ 1 & \infty & 1 & 4 & \infty & 1 \\ \infty & 1 & \infty & 4 & 2 & 1 \end{pmatrix}$$

$Kesz = \{1\}, d(2) = 2, Apa(2) = 1, d(4) = 1, Apa(4) = 1, d(5) = 8, Apa(5) = 1,$
$Kesz = \{1,4\}, d(3) = 5, Apa(3) = 4, d(5) = 3, Apa(5) = 4, d(6) = 8, Apa(6) = 4,$
$Kesz = \{1,4,2\}, d(3) = 3, Apa(3) = 2, d(6) = 6, Apa(6) = 2,$
$Kesz = \{1,4,2,3\},$
$Kesz = \{1,4,2,3,5\} d(6) = 4, Apa(6) = 5,$

Tulajdonságok

Tegyük fel, hogy a G = (V,E,c) irányított, súlyozott gráfra és s kezdőpontra végrehajtottuk a Kezd eljárást, majd valahány Kozelit műveletet. Ekkor az alábbi összefüggések teljesülnek.

Háromszög egyenlőtlenség: $(\forall (u,v) \in E)(\delta(s,v) \leq \delta(s,u) + c(u,v))$

Bizonyítás A legrövidebb $s \rightsquigarrow u$ utat folytatva az (u,v) éllel egy $s \rightsquigarrow v$ utat kapunk.

Felső korlát tulajdonság: $d(v) \geq \delta(s,v)$ és ha egyszer $d(v) = \delta(s,v)$, akkor ezután mindig teljesül az egyenlőség.

Bizonyítás Az egyenlőtlenség a Kozelit lépések száma szerinti indukcióval igazolható, felhasználva a háromszög egyenlőtlenséget. Ha egyszer létrejön az egyenlőség az utána nem változhat, mivel d értéke az eljárás során nem növekedhet, és az egyenlőtlenség miatt nem is csökken tovább.

Nincs-út tulajdonság: Ha nincs $s \rightsquigarrow v$ út, akkor mindvégig $d(v) = Inf$ teljesül.

Bizonyítás A felső korlát tulajdonságból azonnal következik.

Konvergencia tulajdonság: Ha $s \rightsquigarrow u \to v$ egy legrövidebb út és $d(u) = \delta(s,u)$ Kozelit(G,u,v,Q) végrehajtása előtt, akkor Kozelit(G,u,v,Q) után $d[v] = \delta(s,v)$.

Bizonyítás Kozelit(G,u,v,Q) után $d(v) \leq d(u) + c(u,v) = \delta(s,u) + c(u,v) = \delta(s,v)$, így a felső korlát tulajdonság alapján egyenlőség kell fennálljon.

Út-közelítés tulajdonság: Ha $p = \{v_0, v_1, \ldots, v_k\}$ egy $s = v_0 \rightsquigarrow v_k$ legrövidebb út, akkor a $(v_0,v_1),(v_1,v_2),\ldots,(v_{k-1},v_k)$ élekre ebben a sorrendben végrehajtott Kozelit eljárások után $d(v_k) = \delta(s,v_k)$.

Bizonyítás A konvergencia tulajdonságból következik.

LUF tulajdonság: Tegyük fel, hogy G nem tartalmaz negatív súlyú éleket és minden $v \in V$ pontra $d[v] = \delta(s,v)$. Ekkor az $F = \{(Apa(v),v) : v \in V, Apa(v) \neq 0\}$ élhalmaz G-nek s gyökerű LUF-ja lesz.

Bizonyítás: A nincs út tulajdonság alapján a fába pontosan azon pontok kerülnek, amelyek elérhetőek s-ből. Továbbá könnyen látható, hogy minden v pontra az F fában az adott pontba s-ből vezető út hossza $d(v)$, mivel d(v)-t minden esetben a $d(apa(v)) + c(apa(v),v)$ érték alapján kapjuk.

Helyesség

Tétel A DIJKSTRA algoritmust nemnegatív él-súlyozott irányított G = (V,E,c) gráfra és s kezdőpontra végrehajtva, minden $v \in V$ pontra teljesül $d(v) = \delta(s,v)$.

Bizonyítás: Bizonyítás a Kesz halmazba kerülés szerinti indukcióval. Az első pont, amely kikerül a Q prioritási sorból és bekerül Kesz-be az s pont, amire az állítás nyilvánvaló.

Legyen u az első pont, amire nem teljesül az állítás, akkor a bekerülésekor a felső korlát tulajdonság miatt $d(u) > \delta(s,u)$. Így $\delta(s,u) \neq \infty$, tehát van egy legrövidebb út s-ből u-ba. Ezen az úton u bekerülésekor a kezdőpont a $Kesz$ halmazban van, a végpont nincs, így van két olyan egymást követő pont (x,y) hogy $x \in Kesz$ és $y \notin Kesz$.

Ekkor x még u előtt került be a Kesz halmazba, így $d(x) = \delta(s,x)$. Továbbá x bekerülésekor végrehajtottuk a $Kozelit(G,x,y,Q)$ műveletet, így a konvergencia tulajdonság miatt $d(y) = \delta(s,y)$.

Minden él súlya nemnegatív, y rajta van az u-ba vezető legrövidebb úton, így $\delta(s,y) \leq \delta(s,u)$. Továbbá a prioritási sorból u-t hamarabb választjuk ki, mint y-t, így $d(u) \leq d(y)$.

Következésképpen azt kaptuk, hogy $d(y) = \delta(s,y) \leq \delta(s,u) < d(u) \leq d(y)$, ami ellentmondás, így a tételt igazoltuk.

Következmény: A tételből a LUF tulajdonság alapján következik az algoritmus helyessége.

Floyd Warshall algoritmus

A feladat egy súlyozott gráfban minden pontpárra a legrövidebb utak megkeresése. Az input a súlyozott gráf. Outputként egy mátrixot adunk meg, amely a legrövidebb utakat tartalmazza, továbbá egy segédmátrixot, amely minden pontpárra tartalmazza a legrövidebb út első pontját, így a mátrix alapján a legrövidebb út egyszerűen megkapható.

A feladatot dinamikus programozással oldjuk meg. Legyen $V = 1 \ldots n$ és $c(i,j) = \infty$, ha $(i,j) \notin E$, legyen $G(i,j) = c(i,j)$ ha $i \neq j$ és $G(i,i) = 0$, $i = 1,\ldots,n$.

Részproblémákra bontás:

$\forall i,j \in \{1,\ldots,n\}$-ra és $\forall k \in \{0,1,\ldots,n\}$-re legyen $D^k(i,j)$=az i-ből j-be vezető olyan utak hosszának minimuma, amelyek legfeljebb az $\{1,\ldots,k\}$ belső pontokon mennek keresztül. Ekkor $D^n(i,j) = \delta(i,j)$ és $D^0(i,j) = G(i,j)$.

Rekurzív összefüggés ($k \geq 1$):

$$D^k(i,j) = \min\{D^{k-1}(i,j), D^{k-1}(i,k) + D^{k-1}(k,j)\}$$

,

Megjegyzés: A rekurzió kiszámítása egyetlen tömbben megoldható, mert $D^{k-1}(i,k) = D^k(i,k)$ és $D^{k-1}(k,j) = D^k(k,j)$

Floyd Warshall algoritmus

```
FloydWarshall(G)
for i=1 to n
   {for j=1 to n
      {D[i,j]:=G[i,j]
      if D[i,j]<INF then Elso[i,j]:=j}}
for k=1 to n
    {for i=1 to n
        {for j=1 to n
           If D[i,k]+D[k,j]<D[i,j]
              then {D[i,j]:=D[i,k]+D[k,j]
                    Elso[i,j]:=Elso[i,k]}}}
```

Futási idő: $\Theta(n^3)$

A G = (V,E) gráf tranzitív lezártja az a $G^* = (V,E^*)$ gráf, ahol $E^* = \{(u,v) : u \rightsquigarrow v\}$.

Egy lehetséges algoritmus G^* kiszámítására: vegyük azt a gráfot, amelyben minden létező él súlya 1, és alkalmazzuk a Floyd-Warshall algoritmust. Az i és j pontok között akkor és csak akkor van út G-ben, ha távolságuk nem ∞.

Megjegyzés: Azonban a Floyd-Warshall algoritmus egyszerű módosításával (Boolean értékeket használva a távolságok helyett) hatékonyabb megoldást kapunk.

Minimális feszítőfák

Legyen $G = (V,E,c)$, $c : E \rightarrow R^+$ egy súlyozott irányítatlan gráf. Terjesszük ki a súlyfüggvényt a $T \subseteq E$ élhalmazokra: $C(T) = \sum_{(u,v)\in T} c(u,v)$

Az F = (V,T) gráf minimális feszitőfája G-nek, ha

- F feszitőfája G-nek, és
- C(T) minimális

Legyen $A \subseteq F$ valamely (V,F) minimális feszítőfára, és $(u,v) \in E$.
Definíció (u,v) biztonságos él A-ra nézve, ha $A \cup \{(u,v)\}$ is része valamely minimális feszítőfának.

Elvi algoritmus
$A := \emptyset$
While A nem feszítőfa
        (u,v) biztonságos él keresése;
        $A := A \cup \{(u,v)\}$

Definíció: A G = (V,E) gráf vágása: $(S, V \setminus S)$, ahol $S \subseteq V$;

Definíció: $(u,v) \in E$ keresztél az $(S, V \setminus S)$ vágásra, ha $u \in S$ és $v \in V \setminus S$, vagy $u \in V \setminus S$ és $v \in S$. Az $(S, V \setminus S)$ vágás elkerüli az $A \subseteq E$ élhalmazt, ha A-ban nincs keresztél.

Definíció: (u,v) könnyű él az $(S, V \setminus S)$ vágásra, ha a legkisebb c-értékű (súlyú) keresztél.

Tétel: Ha A része a G = (V,E,c) valamely minimális feszítőfájának és elkerüli az $(S, V \setminus S)$ vágást, továbbá (u,v) könnyű él az $(S, V \setminus S)$ vágásra, akkor (u,v) biztonságos él A-ra nézve.

Bizonyítás: Legyen T = (V,F) egy olyan minimális feszítőfa, amelyre $A \subseteq F$. Ha $(u,v) \in F$, akkor az állítás nyilvánvaló.
Ha $(u,v) \notin F$, akkor (u,v)-t hozzá véve az F éleihez kört kapunk. Mivel u és v az S vágás különböző oldalán vannak, ezért van a körben egy másik (x,y) keresztél. Ekkor az $F' := F \setminus \{(x,y)\} \cup \{(u,v)\}$ élhalmaz is feszítőfája lesz G-nek. Továbbá $c(u,v) \leq c(x,y)$ miatt $C(F') \leq C(F)$, így szintén minimális.

Kruskal algoritmusa

Kruskal(G,w)
 Letesit(A: halmaz)
 for $(v \in V)$
     $Halmazt - Keszit(v)$
 rendezzük E éleit w szerint növekvő sorrendbe
 for $(u,v) \in E$ esetén a súly szerinti sorrendben
     If $Halmazt - Keres(u) \neq Halmazt - Keres(v)$
         $A := A \cup \{(u,v)\}$
         Egyesít(u,v)

Megvalósítás: Unio Holvan adattípussal.

Helyesség: Az általános tétel alapján, vágásnak olyan vágást használva, ahol az egyik halmaz az u-tartalmazó részfa az aktuális feszítő erdőből.

Futási idő: $O(|E| \cdot \log |V|)$

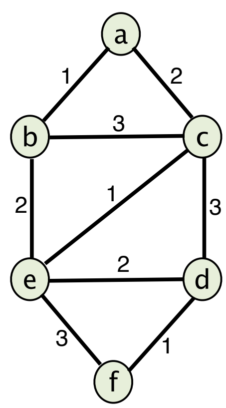
Prím algoritmusa

```
Prim(G,c,r)
 for(v in V)
     {d(v):=Inf
     apa(v):=0
     Bent(v):=0}
 d(r):=0
 Letesit(Q: ModPrisor)
 for(v in V)
    Sorba(Q,v)
 while(Elemszam(Q)>0)
   {Sorbol(Q,u)
    Bent(u):=1
    for(v in Kiel(G,u))
        {If (Bent(v)=0) and (c(u,v)<d(v))
           then {apa(v):=u
                 d(v):=c(u,v)}}}
```

Helyesség: Az általános tétel alapján, vágásnak olyan vágást használva, ahol az egyik halmaz azon v pontokat tartalmazza, akikre Bent(v)=1.

Futási idő: $O(|E|+|V|\cdot\log|V|)$

Példa



2. ábra.

A Kruskal algoritmus a következő sorrendben választja be az éleket:

$(a,b),(c,e),(d,f),(a,c),(d,e),$

A Prim algoritmus sorrendje
$(a,b),(a,c),\ (c,e),\ (d,e),(d,f).$