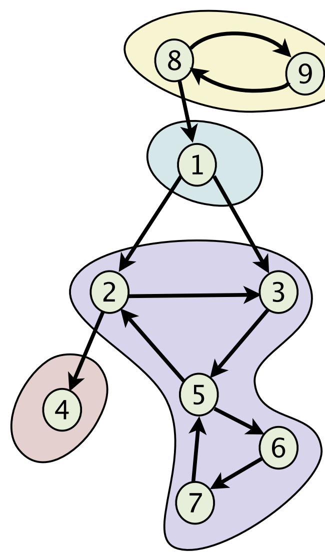
Elvi algoritmus

- 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.



1. ábra.

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 $s \rightsquigarrow u \rightarrow v$ 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.

Példa

Tegyük fel, hogy egy hivatalban minden hivatalnok megvesztegethető. Nem mindegyiküket lehet közvetlenül megvesztegetni a magasabb hivatalban levő embereket csak az vesztegetheti meg, aki rendelkezik megfelelő ajánlólevéllel, amit az ajánlólevél irójának megvesztegetésével lehet megszerezni. Egy jól értesült vállalkozó tudja kit mennyibe kerül megvesztegetni és azt is, hogy az egyes emberek milyen ajánlóleveleket fogadnak el. A hivatal vezetőjét szeretné megvesztegetni a lehető legkisebb teljes költség megfizetése mellett. Adjunk meg egy eljárást, amellyel meghatározhatja kiket kell megvesztegetnie!

Megoldás

Vegyünk egy gráfot, amelynek pontjai a vállalkozó és a tisztviselők. A vállalkozóból él megy azokba a tiszviselőkbe, akiket közvetlenül megvesztegethet, továbbá minden tisztviselőből él megy azokba a tisztviselőkbe, akik elfogadják az ajánlólevelét. A gráfban a csúcsoknak van súlya, a vállalkozóé 0, a tisztviselőké a megvesztegetés összege. A cél a vállalkozóból a hivatal vezetőjébe egy legrövidebb út megkeresése, ahol az út hossza a benne levő csúcsok súlyainak összege.

Ez megoldható a legrövidebb út algoritmusok módosításával is, de visszavezethetjük a legrövidebb út problémára. Minden csúcsot helyettesítsünk egy éllel, amelynek súlya a csúcs súlya. Az él kezdőpontjába mennek azok az élek, amelyek a csúcsba mentek, az él végpontjából mennek azok az élek, amelyek a csúcsból mentek. (Szemléletesen ez azt jelenti, hogy az új csúcsok a megvesztegetések kezdetei és végei, a kezdet előfeltétele az ajánlólevél, és ajánlólevelet az ember csak a vesztegetés végén kap.

Ford-Bellman algoritmus.

Az algoritmus egy negatív súlyú kört nem tartalmazó G=(V,E) súlyozott gráfban határozza meg egy s kezdőpontból a legrövidebb utak hosszát és a legrövidebb utak feszítőfáját. Ha a gráf tartalmaz negatív kört hamis értéket ad vissza egyébként igazat.

```
FordBellman(G,s)
 for(v in V)
    {d(v):=INF
    Apa(v):=0}
 d(s):=0
 for i=1 to |V|-1
    {for (u,v) in E
       {if d(v)>d(u)+c(u,v)
           then {d(v):=d(u)+c(u,v)
                 apa(v):=u}}}
 for (u,v) in E
       {if d(v)>d(u)+c(u,v)
```

```
           then return False}
 return True
```

Legrövidebb utakat meghatározó algoritmus körmentes irányított gráfokban.

Elsőként a csúcsoknak vesszük a topologikus rendezés szerinti sorrendjét. Ekkor egy s pontból csak azon pontokba vezet út, amelyek a sorrendben s után jönnek. s-ből az utána jövő csúcsokba a legrövidebb utak fáját a következő algoritmussal kaphatjuk meg.

```
Kormentes(G,s)
for(v in V)
    {d(v):=INF
    Apa(v):=0}
 d(s):=0
 for(u in V a topologikus sorrendben)
    for (v in Ki(u))
        {if d(v)>d(u)+c(u,v)
           then {d(v):=d(u)+c(u,v)
                Apa(v):=u}}
```

Helyesség: Indukcióval látható, hogy minden pont esetén $d(u) = \delta(s,u)$ fennáll, amikor a külső for ciklusban sorra kerül.

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 $V = 1\dots n$ és $c(i,j) = \infty$, ha $(i,j) \notin E$, legyen $G(i,j) = c(i,j)$ ha $i \neq j$ és $G(i,i) = 0$, $i = 1,\dots,n$.

Részproblémákra bontás:

$\forall i,j \in \{1,\dots,n\}$-ra és $\forall k \in \{0,1,\dots,n\}$-re legyen $D^k(i,j)$=az i-ből j-be vezető olyan utak hosszának minimuma, amelyek legfeljebb az $\{1,\dots,k\}$ 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 $G = (V, E, c)$, $c : E \to R^+$ 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 $c(u,v) \le c(x,y)$ miatt $C(F') \le C(F)$, 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

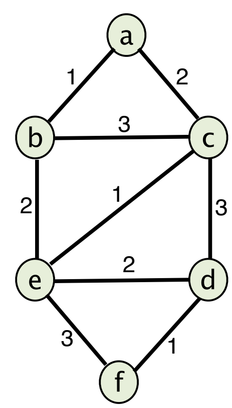
If $Halmazt - Keres(u) \neq Halmazt - Keres(v)$
$A := A \cup \{(u,v)\}$
Egyesít(u,v)

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: $O(|E| \cdot \log |V|)$

Példa



2. ábra.

A Kruskal algoritmus a következő sorrendben választja be az éleket:
$(a,b),(c,e),(d,f),(a,c),(d,e)$,

Kiskérdések jövő hétre

- Dijsskstra algoritmus
- Ford Bellman algoritmus
- Floyd Warshall algoritmus

Szorgalmi

Útvonalak Adott egy gráf és annak egy kiválasztott éle. A feladat az összes olyan pontpár meghatározása a gráfban, amelyek között csak olyan út van, ami átmegy a kiválasztott élen. Adunk egy $O(n^3)$ futási idejű algoritmust, amely ezeket a pontpárokat meghatározza.

Beküldés: cimreh@inf.u-szeged.hu,

Pszeudókód +magyarázat + futási idő elemzés

- első öt megoldó 8-8 pont
- a második öt megoldó 5-5 pont

A szerzett plusszpontok a vizsga minimumkövetelményébe nem számítanak bele.