A Számítástudomány alapjai

Leszámoló kombinatorika

Leszámlálási alapfogalmak, permutációk, variációk és kombinációk (ismétlés nélkül és ismétléssel), példákkal, kiszámításuk, binomiális együtthatók közti egyszerű összefüggések, a binomiális tétel.

Faktoriális

Definíció: Az egész számok 1-től n-ig való összeszorzás n faktoriális. Jele: n!

$$n! = \prod_{i=1}^{n} i$$

$$0! = 1$$

Variációk

~kiválasztás és sorba rendezés

Ismétlés nélküli variáció

Definíció: n elem k-ad osztályú ismétlés nélküli variációja: n különböző elemből k kiválasztása és sorba rendezése.

Jele: $V_{(n,k)}$

$$V_{(n,k)} = \binom{n}{k} k! = \frac{n!}{(n-k)!}$$

Megfigyelés:

$$V_{(n,k)} = n * V_{(n-1\,,\;k-1)}$$

Bizonyítás: A sorrend első eleme n féle lehet (n különböző eset). A maradék k-1 elem sorrendje meghatározza az eredetei ismétlés nélküli variációt.

$V_{(n,k)}$= az egyesesetekben megadható a maradék sorrendek összes száma. Minden esetben $V_{(n-1\,,\;k-1)}$ a lehetséges sorrendek összes száma.

Következésképpen:

$$V_{(n,k)} = n * V_{(n-1\,,\;k-1)} = V_{(n,k)} = n * (n-1) * V_{(n-2\,,\;k-2)} = \prod_{i=0}^{k-1} (n-i) = \frac{n!}{(n-k)!}$$

Ismétléses variáció

Definíció: n elem k-ad osztályú ismétléses variációja: n különböző elemből egy k hosszú sorrend, ahol az ismétlés korlátlanul megengedett.

Jele: $V_{\mathrm{ism}\,(n,k)}$

$$V_{ism\,(n,k)} = n * V_{(n\,,\;k-1)}$$

Következésképpen: $V_{\mathrm{ism}\,(n,k)} = n^i * V_{(n\,,\;k-i)} = n^k$

Permutációk

~sorba rendezés

Ismétlés nélküli permutáció

Definíció: n elem összes lehetséges sorrendje n ismétlés nélküli permutációi. Ezek száma n!

$$V_{(n,n)} = \frac{n!}{(n-n)!} = \frac{n!}{0!} = n!$$

Ismétléses permutáció

Definíció: Legyen $n = \sum_{i=1}^{l} k_i = k_1 + k_2 + \cdots + k_l$. n elem ismétlés permutációinak száma n elem sorrendje, ahol az elemek mindegyike az l típus valamelyikéből kerül ki, úgy, hogy az 1. típusból k_1, a 2. típusból k_2, … az l. típusból k_l db van.

Tétel: Az ismétléses permutációk száma:

$$\frac{n!}{\prod_{i=1}^{l} k_i!} = \frac{n!}{k_1! * k_2! * \dots * k_l!}$$

Bizonyítás: Különböztessük meg az azonos típusú elemeket, ekkor n! a lehetséges sorrendek száma. Ugyanaz az ismétlés nélküli permutáció pontosan $\prod_{i=1}^{l} k_i! = k_1! * k_2! * \dots * k_l!$ ismétléses permutációhoz tartozik. Így az ismétléses permutációk száma:

$$\frac{n!}{\prod_{i=1}^{l} k_i!} = \frac{n!}{k_1! * k_2! * \dots * k_l!}$$

Kombinációk

~kiválasztás

Ismétlés nélküli kombináció

Definíció: n elem k-ad osztályú ismétlés nélküli kombinációja: n elem közül k kiválasztása, úgy, hogy a sorrend nem számít.

Jele: $C_{(n,k)}$

Megfigyelés: $V_{(n,k)} = k! * C_{(n,k)}$

Bizonyítás: Bármely kombinációhoz pontosan k! variáció tartozik.

Következésképp:

$$C_{(n,k)} = \frac{V_{(n,k)}}{k!} = \frac{n!}{(n-k)! * k!} = \binom{n}{k}$$

Megfigyelés:

$$\binom{n}{k} = \binom{n}{n-k}$$

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

Ismétléses kombináció

Definíció: n elem k-ad osztályú ismétléses kombinációja: n különböző elemből k kiválasztása, úgy, hogy a típus ismétlése korlátlanul megengedett.

Jele: $C_{\mathrm{ism}(n,k)}$

Tétel:

$$C_{ism(n,k)} = \binom{n+k-1}{k}$$

Bizonyítás: Bármely ismétléses kombináció úgy néz ki, hogy k egy n tagú nem negatív egészekből álló összeg. Írjuk fel ezt 1-es számrendszerben. Ez a felírás k db 1-est és n-1 db + jelet fog tartalmazni.

Megfigyelés: minden olyan ismétléses permutáció, ami k db 1-est és n-1 db +-t tartalmaz megfelel egy különböző ismétléses kombinációnak.

$$C_{ism(n,k)} = \frac{(k+n-1)!}{(n-1)! * k!} = \binom{n+k-1}{k}$$

Binomiális együtthatók

Binomiális tétel

Legyen $x, y \in \mathbb{R}$ és $n \in \mathbb{N}_+$.

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i$$
$$= \binom{n}{0} x^n y^0 + \binom{n}{1} x^{n-1} y^1 + \binom{n}{2} x^{n-2} y^2 + \dots + \binom{n}{n-1} x^1 y^{n-1} + \binom{n}{n} x^{n0} y^n$$

Bizonyítás

Bármely kifejezési tag n tényezős, tehát $x^{n-i}y^i$ alakú. Ilyen tagból pontosan annyi van, ahányféleképpen az n darab zárójelből kiválasztható az az i darab, ahonnan y-t választjuk a kifejtési tagba. Ez éppen $\binom{n}{i}$.

Tétel

Tétel: Minden $n \in \mathbb{N}^+$-re:

$$\sum_{i=0}^{n}\binom{n}{i} = \binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\cdots\binom{n}{n-1}+\binom{n}{n} = 2^n$$

Bizonyítás: Számoljuk ki kétféleképpen, hány n hosszú 0-1 sorozat van.

Egyrész 2 elem n-ed osztályú ismétléses variációja, amelynek száma 2^n. Másrészt számoljuk meg, hány olyan n hosszú 0-1 sorozat van, amely pontosan $i \in [0, n]$ darab egyes van. Mivel ugyanazt számoltuk meg kétfélképpen, ezért a két mennyiség egyenlő.

Következmény:

$$\sum_{i=0}^{n}(-1)^i\binom{n}{i} = \binom{n}{0}-\binom{n}{1}+\binom{n}{2}-\cdots\pm\binom{n}{n} = 0$$

Bizonyítás: Az n-1. sor összes elemét egyszer vesszük pozitív, egyszer negatív előjellel.

$$\sum_{i=0}^{n-1}\binom{n-1}{i} = \binom{n-1}{0}+\binom{n-1}{1}+\binom{n-1}{2}+\cdots\binom{n-1}{n-1} = 2^{n-1}$$

Gráfelmélet

Gráfelméleti alapfogalmak. Gráfok fokszámösszege, komponensek, utak, séták, élsorozatok, izomorfia. Fák és erdők, azok egyszerűbb tulajdonságai.

Feszítőfa, alapkörrendszer (fundamentális körrendszer), fundamentális vágásrendszer. Minimális költségű feszítőfa, Kruskal algoritmusa.

Gráfbejárás fogalma, élek osztályozása, BFS. Legrövidebb utak és a BFS tulajdonságai, legrövidebb utak fája. Élmenti javítás, Dijkstra algoritmusa.

Normál fa meg lett említve, a vizsgára nem kell.

A Dijkstra-algoritmus helyessége és lépésszáma. Ford és Floyd algoritmusai. Mélységi keresés, irányított körök keresése, aciklikus gráfok jellemzése. PERT feladat, megoldásnak algoritmusa.

Idén nem szerepel: Legszélesebb út keresése irányítatlan gráfban.

Alapfogalmak

Gráf

Definíció: Egy gráf egy rendezett pár, G=(V,E), ahol V egy nem üres halmaz, E pedig az ebből a halmazból képezhető párok egy halmaza.

$V \neq \emptyset$

$E \subseteq \binom{V}{2}$

Irányított gráf: Egy gráf irányított, ha az él kezdőpontjából szomszédos a az él végpontja, de fordítva nem.

Csúcsok és élek alapfogalmai

Csúcsok: Csúcsok/pontokhalmaza V(G). A csúcsok száma v(G)

Élek: Az élek halmaza E(G), a V(G) 2 elemű részhalmazai. Az élek száma e(G). Ha $e \in E$ a $\{v_1, v_2\}$ párnak felek meg, akkor v_1, v_2 e él végpontjai.

Ha $v_1 = v_2$, akkor e **hurokél**.

A **párhuzamos/többszörös élek,** olyan különböző nem hurokélek, amelyek végpontjai megegyeznek.

Ha $e, f \in E$ végpontjai $\{v_1, v_2\}$ illetve $\{w_1, w_2\}$, és $\{v_1, v_2\} \cap \{w_1, w_2\} \neq \emptyset$ akkor e és f szomszédos élek.

v_1, v_2 szomszédos pontok, ha $\{v_1, v_2\} \in E$.

v_1 illeszkedik e-re, ha annak egyik végpontja.

Fokszám: A pontra $V \in V(G)$ illeszkedő élek száma, plusz a pontra illeszkedő hurokélek számának kétszerese. Jele $d(v)$

Irányított gráfok esetén külön beszélhetünk **kifok**áról (=a csúcsból induló irányított élek száma) és **befok**áról (=a csúcsba érkező irányított élek száma).

Izolált pont: nincs vele szomszédos pont, fokszáma 0.

Teljes fokú csúcs (apex): egyszerű n pontú gráf n-1 fokú csúcsai teljes fokúak.

Handshake lemma (HSL) „kézfogás-segédtétel”

Tétel: Tetszőleges irányítatlan gráfra a gráf fokszámösszege az élszám összegének kétszerese. Ebből kivetkezik, hogy a gráfok fokszámösszege páros.

$$\sum_{v_i \in V(G)} d(v_i) = 2e(G)$$

Irányított esetben a gráf ki és be fokszámösszege egyenlő az élek számával.

Bizonyítás: Építsük fel G-t üresgráfból élek egyenkénti behúzásával. A HSL teljesül üres gráfra is. Minden él behúzásával a jobb és a bal oldal is kettővel növekszik.

Gráf típusok, gráf elemek - definíciók

Egyszerű gráf

Egyszerű gráfban nincs hurokél és párhuzamos élek.

Nem egyszerű gráfban van hurokél vagy párhuzamos élek.

Teljes gráf

n pontú teljes gráf bármely két pontja között vezet út. Jele K_n

Reguláris gráf

Irányítatlan G=(V,E) gráf k reguláris, ha bármely csúcsának fokszáma k.

Irányított G=(V,E) gráf k reguláris, ha bármely csúcsának kifoka és befoka is k.

Komplementer gráf

G=(V,E) gráf komplementere $\overline{G}$, amelyben azok az élek vannak behúzva, amik G-ben nincsenek.

$$\bar{G} = (V', E')$$
$$V' = V$$
$$E' = \binom{V}{2} - E$$
$$\overline{f_i} = n - 1 - f_i$$

E(G) és E($\overline{G}$,) diszjunkt ponthalmazok.

Izomorfia

A G=(V,E) és G’=(E’,V’) gráfok izomorfok, ha létezik olyan egyértelmű megfeleltetés/bijekció/számozás v és V’ között, hogy bármely i, j esetén az i. csúcsból pontosan annyi él fut a j. csúcsba G-ben, mint G’-ben.

Kör

n pontú kör bármely két pontja közt legalább 2 különböző út van. Jele: C_n

Élsorozat

Az élsorozat olyan csúcsból csúcsba menő élek sora, ahol az él és csúcs ismétlődés korlátlanul megengedett.

Egy $v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k$ sorozat élsorozat, ha e_i v_{i-1} és v_k pontokat összekötő él.

Zárt élsorozat kezdő és végpontja megegyezik. $v_0 = v_k$

Séta

A séta olyan csúcsból csúcsba menő élek sorozata, ahol az élismétlés tiltott, a csúcsismétlés viszont korlátlanul megengedett. Olyan élsorozat, amiben nincs élismétlődés.

Zárt séta kezdő és végpontja megegyezik.

Út

Az út olyan csúcsból csúcsba menő élek sorozata, ahol nem megengedett a csúcs- és élismétlődés. Olyan séta, ahol a csúcsok mind különbözőek. Jele P_n

Egyszerű gráfban $v_0, v_1, v_2, \dots, v_{k-1}, v_k$-val írjuk le.

A kör tekinthető zárt útnak.

Megfigyelés: Ha G-ben ($\exists$ uv út) $\Leftrightarrow$ ($\exists$ uv séta) $\Leftrightarrow$ ($\exists$ uv élsorozat).

Átmérő: legtávolabbi csúcsok közti legrövidebb út.

Él-és csúcstörlés

Az éltörlés két pont egy összekötöttségének megszűntetése. Ellentétes művelete az élbehúzás, amely két pont közt vezető 1 hosszú utak számát eggyel növeli.

Csúcstörlés esetén egy csúcsot, és a hozzá kapcsolódó éleket töröljük.

Véges gráf

Véges gráfban a csúcsok és az élek száma is véges.

Véges 2 reguláris gráf minden komponense kör.

Véges gráf páratlan fokszámú csúcsainak darabszáma páros. Indoklás: HSL.

Részgráf – definíciók

Részgráf

G' részgráfja G-nek, ha megkapható G-ből él- és csúcstörlésekkel.

G'=(V',E')akkor részgráfja G=(V,E)-nek, ha $V' \subseteq V$ és $E' \subseteq E$.

Feszített részgráf

G' feszített részgráfja G-nek, ha megkapható G-ből csúcstörlésekkel.

G'=(V',E')akkor feszített részgráfja G=(V,E)-nek, ha $V' \subseteq V$ és E' az összes olyan G beli élet tartalmazza, ami G' élei közt fut..

Feszítő részgráf

G' feszítő részgráfja G-nek, ha megkapható G-ből éltörlésekkel.

G'=(V',E')akkor feszítő részgráfja G=(V,E)-nek, ha $V' = V$ és $E' \subseteq E$.

Komponensek és összefüggőség

Komponens definíció

K a G gráf komponense, ha bármely két K-n belüli csúcs között vezet út, de K-n kívüli csúcsba nem vezet út.

Állítás

Bármely G gráf egyértelműen bontható fel komponensekre, és a G bármely két különböző komponense diszjunkt ponthalmaz (metszetük $\emptyset$)

Összefüggő gráf

G összefüggő, ha pontosan 1 komponense van, azaz bármely 2 csúcsa között vezet út.

Irányított esetben G **erősen összefüggő,** ha bármely csúcsból bármely csúcsba vezet irányított út, **gyengén összefüggő**, ha irányítás nélkül összefüggő.

Él hozzáadási lemma

Állítás

Tegyük fel, hogy G tetszőleges irányítatlan gráf. G+e jelöljön egy olyan gráfot, amit G-ből egy él behúzásával kapunk. Az alábbiak közül pontosan egy teljesül:

1) {G+e körök}={G körök} és G+e-nek eggyel kevesebb komponense van, mint G-nek
2) G+e-nek ugyanannyi komponense van, mint G-nek, és G+e-ben több kör van, mint G-ben.

Bizonyítás

Vizsgáljuk G komponenseit:

1) Az e él két különböző komponens között fut:
   - nem keletkezik kör
   - a komponensek száma eggyel csökken
2) Az e él G ugyanazon komponensének két pontja között fut
   - a komponensek nem változnak
   - legalább egy új kör keletkezik

Éltörlési lemma

Tegyük fel, hogy G tetszőleges irányítatlan gráf. G-e jelöljön egy olyan gráfot, amit G-ből egy él törlésével kapunk. Az alábbiak közül pontosan egy teljesül:

1) {G-e körök}={G körök} és G-e-nek eggyel több komponense van, mint G-nek
2) G-e-nek ugyanannyi komponense van, mint G-nek, s G-e-ben kevesebb kör van, mint G-ben.

Élvágó halmaz, vágás - definíció

Élvágó halmaz

Egy X⊆E élhalmaz élvágó halmaz, ha az X-beli élek elhagyásával nő a gráf komponenseinek száma.

Vágás

X vágás, ha élvágó, de semelyik valódi részhalmaza nem az.

Egyértelmű vágás

Az egyértelmű vágások élvágó élek.

Irányított gráfok

Irányított gráfok élei (v_1, v_2)rendezett párok, ahol v_1 az e él kezdőpontja, $v_2$pedig a végpontja.

Forrás: egyetlen élnek sem végpontja

Nyelő: egyetlen élnek sem a kezdőpontja

Állítás

Bármely irányítatlan gráfnak létezik aciklikus (körmentes) irányítása

Bizonyítás: Tetszőleges pontot nyelővé teszünk, ez elhagyható. Az így kapott részgráfban újra nyelővé teszünk egy pontot, és elhagyjuk. Ezt folytatjuk egészen addig, amíg csak egy csúcs marad a gráfban

Fák és tulajdonságaik

Definíciók

Erdő

G erdő, ha G körmentes

Fa

G fa, ha összefüggő erdő. / G fa, ha G összefüggő, körmentes gráf.

Levél

Fa elsőfokú csúcsa

Erdő és fa élszáma

Tétel

Ha G egy n pontú erdő és k komponense van, akkor éleinek száma n-k.

$$|E(G)| = n - k$$

Bizonyítás
Építsük fel G-t $\overline{K_n}$-ből élek egyenkénti behúzásával. Mivel kört sosem hozunk létre, ezért minden élbehúzással eggyel csökkentjük a komponensek számát. Végül k komponense van a gráfnak, így a behúzott élek száma n-k.

Következmény
Mivel a fa összefüggő, tehát komponenseinek száma egy, ezért az n pontú fa éleinek száma n-1.

Fa tulajdonságai

Állítás
Minden G gráf esetén az alábbi 3 tulajdonság közül bármely 2 maga után vonja a harmadikat.

1) G összefüggő
2) G körmentes
3) $|E(G)| = |V(G)| - 1$

Tételek
Bármely $e \in E(F)$ él törlésével kapott F-e gráfnak 2 komponense van.

Bármely $u, v \in V(F)$ pontokhoz $\exists!$ (egyértelműen létezik) uv-út F-ben.

Mivel F összefüggő, ezért létezik uv út. Tegyük fel, hogy van olyan e él, amit csak az egyik uv út tartalmaz, ekkor e él törlésével F-e összefüggő maradna, így ellentmondásra jutottunk.

Élhozzáadás során F+e-nek pontosan 1 köre keletkezik.

F+e-ben létezik kör, hiszen a komponensek száma nem csökkenthető. e két végpontja között pontosan 1 út vezet, ezért a kör egyértelmű.

Ha $|V(F)| \geq 2$ akkor F-nek van legalább 2 levele.

Bizonyítás: Tekintsük a fában található leghosszabb $v_1 v_k$ utat. Tegyük fel, hogy v_k nem elsőfokú, ekkor létezik v_{k+1} csúcs, amely szomszédos v_k-val. Ellentmondásra jutottunk, hiszen $v_1 v_{k+1}$ út hosszabb, mint $v_1 v_k$.

Feszítőfa

Szemléltetés
Legyen G tetszőleges gráf. Építsük fel G-t élek egyenkénti behúzásával. A komponensek számát csökkentő éleket színezzük zöldre, a kört létrehozó éleket színezzük pirosra.

Ha G összefüggő volt, akkor n-1 zöld él keletkezett. Hiszen n-ről 1-re csökkent a komponensek száma, a piros élek pedig nem befolyásolják a komponensek számát. A zöld élek körmentes összefüggő gráfot, fát alkotnak n csúcson.

Feszítőfa
G feszítőfája G olyan feszítő részgráfja, ami fa.

Megfigyelés: Ha G-nek létezik feszítőfája, akkor G összefüggő.

Állítás: Ha G összefüggő, akkor létezik feszítőfája.

Alapkör
Legyen G tetszőleges gráf, és F egy feszítőfája. Minden $e \in E(G) \setminus E(F)$ élhez (piros) egyértelműen létezik F+e-nek egy köre, amit e alapkörének hívunk.

Alapvágás
Minden $e \in E(F)$ élhez (zöld) tartozik G-nek egy alapvágása, amit F-e 2 komponense határoz meg.

Feszítő erdő
F a G gráf feszítőerdője, ha minden komponense G megfelelő komponensének feszítőfája

A mohó algoritmus

Mohó algoritmus

Egy algoritmus mohó algoritmus, ha minden lépésben az éppen legjobbnak tűnő lehetőséget választja, nem törődve azzal, hogy most rosszabbnak tűnő választással végül jobb eredményt kaphatnánk.

Költségfüggvény

Adott G=(V,E) gráf, és az éleihez rendelt $k: E \to \mathbb{R}_+$ költségfüggvény/súly. G tetszőleges E' élhalmazának k(E') költsége a E' -beli élek összköltsége

Minimális költségű feszítőfa

Az F ⊆ E élhalmaz minimális költségű feszítőfa, ha (V, F) a G feszítőfája, és nem drágább a G egyetlen feszítőfájánál sem, azaz k(F) ≤ k(F') teljesül G minden (V, F') feszítőfájára

Költségminimalizálás - tétel

A cél olyan algoritmus megadása, amely megadja G minimális költségű feszítőfáját.

Legyen k értékkészlete $c_1 < c_2 < c_3 < \cdots < c_{n-1} < c_n$.

A költségfüggvény értékeihez rendeljünk élhalmazt, amely az adott költségnél nem nagyobb súlyú éleket tartalmazza.

$$E_i = \{e \in E(G) \mid k(e) \leq c_i\}$$
$$G_i = (V, E_i)$$

Egy (V,F) feszítőfát gyártunk úgy, hogy G éleit vagy bevesszük faélnek (zöld) vagy nem (piros). A feszítőfába először E_1, E_2, E_3… vesszük be az összes olyan élet, amely nem hoz létre kört. Az így kapott feszítőfa élhalmaza legyen F.

Megfigyelés: Az i. lépésben az így kapott $F \cap E_i$ feszítőerdője G_i-nek.

Következésképp: F minden E_i-ből a lehető legtöbb élt tartalmazza.

Legyen F' G tetszőleges feszítőfája.

$$F=\{f_1, f_2, f_3, \dots, f_n\}$$
$$F'=\{f_1', f_2', f_3', \dots, f_n'\}$$
$$k: f_1 \leq f_2 \leq f_3 \leq \cdots \leq f_n$$
$$k: f_1' \leq f_2' \leq f_3' \leq \cdots \leq f_n'\}$$

Következésképp: $|F \cap E_i| \geq |F' \cap E_i|$

Következésképp: F minimális költségű feszítőfa

Tétel

F minimális költségű feszítőfa akkor és csak akkor, ha $F \cap E_i$ feszítőerdője G_i-nek, ahol G_i=(V, E_i), és $E_i = \{e \in E(G) \mid k(e) \leq c_i\}$ minden $e \in \mathbb{R}_+$-re.

Kruskal-algoritmus

Input: G=(V, E) összefüggő gráf és $k: E \to \mathbb{R}_+$ költségfüggvény.

Output: a G egy minimális költségű feszítőfájának F élhalmaza.

Működés:

Legyen $\mathrm{F}_0 = \emptyset$ és $E = (e_1, e_2, e_3, \dots, e_n)$, ahol $k(e_1) \leq k(e_2) \leq k(e_3) \dots \leq k(e_n)$.

$$F_i = \begin{cases} F_i \cup e_1 & ha\ F_i \cup e_1\ körmentes \\ F_i & ha\ F_i \cup e_1\ tartalmaz\ kört \end{cases}$$
$$F_n = F_{min}$$

Tétel

A Kruskal-algoritmus által kiszámított F élhalmaz a G egy minimális költségű feszítőfája.

Bizonyítás

Az algoritmus végén a kiválasztott élek F feszítőerdőt alkotnak. Tegyük fel, hogy létezik F_0 minimális költségű feszítőerdő, amire k(F_0)<k(F). Ha több ilyen ellenpélda van, válasszuk azt aminek a legtöbb közös éle van F-fel. Legyen $\mathrm{e}_0 \in E(\mathrm{F}_0) \setminus E(F)$.

Ha e_0-t hozzátesszük F-hez kapunk egy C alapkört. Ha valamely $e \in E(C)$ élre k(e)>K(e_0) teljesülne, akkor az algoritmus e helyett e_0-t választottuk volna.

$F_0 - e_0$ két komponensből áll, és van legalább egy olyan $e_1 \in (E(C) - \{e_0\}) \subseteq E(F)$, melynek végpontjai $F_0 - e_0$ két külön komponenséhez tartozik. De nem lehet k(e_1)<k(e_0), mert ekkor k(F_1)<k(F_0), ami ellentmond F_0 minimalitásának. k(e_1)=k(e_0) esetén pedig F_1-nek egyel több közös éle lenne F-fel, mint F_0-nak. Így ellentmondásra jutottunk.

Gráfbejárás

A gráfbejárás

G=(V,E) gráf bejárása során $\forall v \in V$: éléretlen→elért→befejezett. A gráfbejárás során keletkezik egy elérési és egy befejezési sorrend

Általános bejárási lépések

1) ∃ elért csúcs. Legyen u egy elért csúcs.
   a) ∃ u-nak eléretlen szomszédja: v. Ekkor v elértté válik az uv út mentén.
   b) ∄ u-nak eléretlen szomszédja. Ekkor u befejezetté válik.
2) ∄ elért csúcs
   a) ∃ eléretlen csúcs: u. u elértté válik.
   b) ∀ csúcs befejezett. END

A bejárás fája

Azon élek összessége, amelyek mentén valamely csúcs elértté válik.

Élek osztályozása bejárt gráfban

Faél

uv faél, ha v az u-ból vált elértté.

Előreél

uv előreél, ha v az u leszármazottja a bejárás fáján.

Visszaél

uv visszaél, ha v az u őse a bejárás fáján.

Keresztél

uv keresztél, ha nem ős és leszármazott között fut.

Szélességi bejárás – Breath First Search BFS

Szabály

Az általános lépésben u mindig a legkorábban elért csúcs. A legkisebb n elérési indexű csúcsból kell elérni, a vele szomszédos csúcsokat. Ha nincs több ilyen, akkor az n+1 elérési indexű csúcsból folytatódik a bejárás

Megfigyelés

BFS után az elérési sorrend megegyezik a befejezési sorrenddel.

FIFO „First in, First out”

Gráfél nem ugorhat át faélt.

Ha i<j<k és $v_i v_j$ és $v_j v_k$ faél akkor $v_i v_k$ ∉E(G), hiszen, ha $v_i v_k$ él lenne, akkor v_k-t legkésőbb v_i-ből elértük volna. v_k őse nem követhetné v_i-t.

BFS után ∄ előre él.

Mélységi bejárás – Depth First Search DFS

Szabály

Az általános lépésben u mindig a legkésőbb elért csúcs. A legnagyobb n elérési indexű csúcsból kell elérni, a vele szomszédos csúcsokat. Ha nincs több ilyen, akkor az n-1 elérési indexű csúcsból folytatódik a bejárás.

Definíció

A mélységi szám (m) megadja, hogy egy csúcs hányadikként vált elértté.

A befejezés szám (b) megadja, hogy egy csúcs hányadikként vált befejezetté.

Megfigyelés

Ha uv faél	m(u)<m(v)	b(u)>b(v)
Ha uv előreél	m(u)<m(v)	b(u)>b(v)
Ha uv visszaél	m(u)>m(v)	b(u)<b(v)
Ha uv keresztél	m(u)>m(v)	b(u)>b(v)

Megfigyelés

Irányítatlan DFS után nincs keresztél

LIFO „Last in First out"

Irányított körmentes gráf

Definíció - DAG

G irányított gráf DAG Directed Aciclic Graf – Irányított körmentes gráf, ha G-ben ∄ irányított kör.

Definíció Topologikus sorrend

G(V,E) irányított grág csúcsainak $v_0, v_1, v_2, \dots, v_{k-1}, v_k$ topologikus sorrendje, ha $\forall v_i, v_j \in E$ -re i < j , azaz minden él „jobbra" mutat.

Megfigyelés

Ha G-nek létezik topologikus sorrendje, akkor G DAG.

Állítás

Ha G DAG akkor létezik topologikus sorrendje.

Megfigyelés

Ha DFS után van visszaél, akkor G-ben van irányított kör. Következésképp

Ha G DAG, DFS után nincs visszaél, bármely uv él esetén b(u)>b(v).

DFS befejezési sorrendjének megfordítottja minden DAG esetén jó topologikus sorrend.

Legrövidebb út keresése

Élsúlyozatlan esetben

Definíciók

Csúcsok távolsága

G-ben az u és v csúcsok távolsága az uv utak élszámának minimuma.

Legrövidebb utak fája

$F \in E(G)$ r gyökérrel a G egy legrövideb utak fája, r-ből bármely másik csúcsba vezet G egy legrövidebb útja F élein és F fa.

Állítás

BFS után a faélek a gyökérből egy legrövidebb utak fáját alkotják.

Bizonyítás

Tetszőleges r gyökérből induló út i. éleu_{i+1} előtt végződik, különben az $u_i u_{i+1}$ élet átugraná. Következésképpen az i-1. él vége $u_i = v$ előtt van.

Dÿjkstra/Dijkstra algoritmus

Távolságfüggvény definíció

G=(V,E) (ir) gráf és egy l függvény, amely az élekhez valós számokat rendel l: E→ ℝ.

Az út hossza az út éleinek összhossza.

dist(u,v)= a legrövidebb uv út hossza.

Az l hosszfüggvény konzervatív, ha ∄ negatív összhosszúságú kör.

Felső becslés, élmenti javítás

Felső becslés

d: v→ ℝ (r, l) felső becslés, ha d(r,l) ≥ dist(r,v) ∀v∈V.

Élmenti javítás

Ha d(v) > d(u)+l(u,v), akkor d(v)= d(u)+l(u,v).

Megfigyelés

Él menti javítás után is d(r,l) felső becslést kapunk.

Ha d(r,l) felső becskésre ∄ élmenti javítás, akkor a d(r,l) felső becslés pontos. d(v)=dist(r,v) ∀v∈V. d(r)=0.

Bizonyítás

Tekintsünk a végső d(r,l) felső becslést. Cél megmutatni, hogy egyik út mentén sem ∄ élmenti javítás.

Tegyük fel, hogy $u_i, u_j \in V$.

Ha i>j, akkor d(u_i)d(u_j), ezért $u_i u_j$ él mentén nem lehet érdemben javítani.

Ha i<j, akkor az i. fázisban már próbáltunk $u_i u_j$ mentén javítani.

Következésképp egyik él mentén sem lehet javítani. Tehát a Dijkstra helyesen adja meg a csúcsok gyökértől mért távolságát.

Megfigyelés – Legrövidebb utak fája LUF

Ha ∀ csúcsra megjelöljük, hogy melyik él állította be a végső d(v)-t, akkor egy legrövidebb utak fáját kapunk.

Állítás

A Dijkstra lépésszáma a csúcsok számának négyzetével arányos. $\sim n^2$

A Dijkstra algoritmus működése

Input

G=(V,E) (ir)

l: E→ $\mathbb{R}_+$ hosszfüggvény

r∈V gyökér

Inicializálás

$U_0 = \emptyset$

d(r)=0

d(v)= ∞ ∀v$\neq r$

Műveletek

i=1,2,…,$|V|$

1) u_i legyen az a v, amire d(v) minimális a V\\U_{i-i} halmazon
2) Az elért halmaz bővítése u_i-vel $U_i = U_{i-i} \cup u_i$
3) Élmenti javítást végzünk ∀u_iv élen, ahol v∉ U_i.

Output

dis(r,v) ∀v∈V

d(v) az utolsó $|V|$. fázis végén

Ford algoritmus

Működés

Input

G=(V,E) (ir)

l: E→ $\mathbb{R}$ konzervatív hosszfüggvény

r∈V gyökér

Inicializálás

Kiindulási felsőbecslés:

d(r)=0

d(v)= ∞ ∀v$\neq r$

Legyen $\mathrm{E} = \{e_1, e_2, e_3, \dots, e_m\}$ az élek rögzített sorrendje.

Műveletek

A Ford algoritmus maximum n-1 fázisból áll, de ha egy fázisban nem tudunk több él menti javítást végezni, akkor megkaptuk a minimális dist(r,v) távolságfüggvényt. END.

Az i. fázisban élmenti javítást végzünk rendre $e_1, e_2, e_3, \dots, e_m$ éle mentén.

Output

dis(r,v) $\forall v \in V$

a legrövidebb utak fája r gyökérből

Helyesség bizonyítása

Az i. fázis végén d(r.l) felső becslés pontos lesz minden olyan csúcsra, amibe vezet legfeljebb i élből álló legrövidebb út. Ebből következik, hogy az n-1. fázis végére pontos lesz a becslés minden csúcsra.

Ha az n- fázisban is tudnánk végezni él menti javítást, akkor az l hosszfüggvény nem lenne konzervatív.

Állítás

A Ford algoritmus lépésszáma a csúcsok számának köbével arányos. $\sim n^3$

Floyd algoritmus

A Floyd algoritmus G=(V,E) gráf minden $u, v \in \mathrm{V}$ pár távolságát meghatározza.

Input

G=(V,E) (ir)

l: E→ $\mathbb{R}$ konzervatív hosszfüggvény

Output

dis(u,v) $\forall(v,u) \in V$

Inicializálás

Legyen $\mathrm{V} = \{v_1, v_2, v_3, \dots, v_n\}$ a csúcsok rögzített sorrendje.

Definíció

$d^{(k)}(i,j)$ a legrövidebb olyan v_i, v_j út hossza, amelynek belső csúcsai $v_1, v_2, v_3, \dots, v_k$ lehetnek (nem feltétlen eleme mind).

Megfigyelés

$d^{(0)}(i,j) = \mathrm{l(ij)}$ ha $(v_i, v_j) \in E$, tehát v_i és v_j szomszédos csúcsok. Különben $d^{(0)}(i,j) = \infty$.

Műveletek

A k. fázisban $d^{(k-1)}$ értékekből meghatározza a $d^{(k)}$ értékeket, a következő módon.

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

Ha k=n+1, END.

Állítás

A Floyd algoritmus lépésszáma a csúcsok számának köbével arányos. $\sim n^3$

PERT-módszer (Project Evaluation and Review Technique)

Input

G=(V,E) DAG

l: E→ $\mathbb{R}_+$ időkorlát

Output

Megadja minden csúcsra, az oda vezető leghosszabb út hosszát, amely a projekt legkorábbi kezdési idejével egyenlő. Ezek közül a leghosszabb út a teljes projekt végrehajtásához szükséges minimális idővel egyenlő.

Működés

1. lépés – Topologikus sorrend keresés

Az i. lépésben talált forrás lesz a topologikus sorrend i. eleme. Az i. elem letörlése után maradt/keletkező forrás az i+1. elem a topologikus sorrendben.

Eredmény: $v_1, v_2, v_3, \dots, v_n$ topologikus sorrend

2. lépés

$k(v_1), \mathrm{k}(v_2), \mathrm{k}(v_3), \dots, \mathrm{k}(v_n)$ időfüggvények meghatározása, a következő módon.

$$k(v_i) = max\,\{k(v_j) + l(v_i, v_j)\,\}$$

Definíció – Kritikus út

A gráfban található leghosszabb út/utak, amelyek minimális csúszása is a teljes projekt csúszását eredményezik. Kritikus tevékenységek azok, amelyek rajta vannak egy kritikus úton.

∃∄∀∈∉