A számítástudomány alapjai 2020. I. félév

5. gyakorlat. Összeállította: Fleiner Tamás (`fleiner@cs.bme.hu`)

Tudnivalók

Def: *Mélységi bejárás* (avagy *DFS*) alatt olyan gráfbejárást értünk, amikor mindig a lehető legkésőbb elért csúcsból kerül elérésre a soron következőnek elért csúcs. Az elérési illetve befejezési sorrendből adódik minden v csúcshoz egy $m(v)$ *mélységi* ill. $b(v)$ *befejezési szám.*
A *verem (stack)* olyan adatszerkezet, amelyben a tárolt adatokon kétféle művelet végezhető: push esetén egy újabb adatot teszünk az eddig tároltak tetejére, pop pedig a felső adatot veszi el a veremből. (Ezekből megvalósíthtó a peek művelet, ami a felső elem kiolvasása, a verem állapotának megváltoztatása nélkül.) A *sor (queue)* adatszerkezetben az enqueue (a push megfelelője) a sor végére írja az adatot, míg a dequeue (a pop megfelelője) a sor elejéről töröl, a peek itt külön művelet, a sor elejét olvassa ki.
A verem egy LIFO (last in first out), míg a sor egy FIFO (first in first out) típusú adatszerkezet.

DFS megvalósítása veremmel: A csúcsok elérése a verembe kerülésükkel, befejezése pedig a veremből kikerülésükkel történik. Üres veremmel indítunk. (I) Üres a verem esetén (a) ha G-nek nincs eléretlen csúcsa, az algoritmus lefutott, (b) ha van eléretlen v csúcs, akkor v-t betesszük a verembe, és a soron következő mélységi számot kapja. (II) Ha a verem nem üres, akkor (a) ha a verem tetején levő v csúcsnak van eléretlen u szomszédja, akkor u-t a verembe tesszük, u megkapja a soron következő mélységi számot és vu faél lesz, (b) ha nincs v-nek eléretlen szomszédja, akkor v kikerül a veremből és a soron következő befejezési számot kapja.

Megjegyzés: (1) Verem helyett FIFO sorral dolgozva a szélességi bejárást végeznénk el.
(2) A mélységi bejárás önmagát meghívó rekurzív algoritmusként is felfogható. A v-ből indított $Mb(v)$ bejárás abból áll, hogy mindaddig, amíg van v-nek eléretlen u szomszédja $Mb(v)$ meghívja az $Mb(u)$ eljárást. Ha nincs ilyen szomszéd, akkor $Mb(v)$ véget ér, és az az $Mb(v)$-t meghívó $Mb(w)$ bejárás folyatódik. Ha $Mb(v)$-t nem másik bejárás hívta meg, akkor az algoritmus elindít egy $Mb(z)$ bejárást egy eléretlen z csúcsra, ha van még eléretlen z, különben véget ér.

Megfigyelés: (1) A mélységi bejárás lépésszáma lineáris, azaz van olyan c konstans, hogy tetszőleges n csúcsú, m élű gráf mélységi bejárásához legfeljebb $c(n+m)$ lépés szükséges.
(2) Ha uv faél vagy előreél, akkor $m(u) < m(v)$ és $b(u) > b(v)$, ha uv visszaél, akkor $m(u) > m(v)$ és $b(u) < b(v)$, ha pedig uv keresztél, akkor $m(u) > m(v)$ és $b(u) > b(v)$.
(3) Következmény: irányítatlan gráf DFS bejárása után nincs keresztél.
(4) Ha G-ben van visszaél, akkor G tartalmaz irányított kört.
(5) Ha G-ben nincs irányított kör, akkor nincs visszaél, így tetszőleges $uv \in E$ esetén $b(u) > b(v)$.

Def: A $G = (V, E)$ irányított gráf *aciklikus* avagy *DAG (directed acyclic graph)*, ha G-ben nincs irányított kör. A $v_1, v_2, \ldots, v_n$ *topologikus sorrend*, ha $V = \{v_1, v_2, \ldots, v_n\}$ és $v_iv_j \in E \Rightarrow i < j$, azaz ha G minden éle „jobbra" mutat.

Köv.: (1) Ha G DAG, akkor a DFS utáni befejezési sorend megfordítása topologikus sorrend.
(2) Tetszőleges $G = (V, E)$ irányított gráfra ekvivalensek az alábbi állítások:
(a) G DAG, (b) G csúcsainak van topologikus sorrendje, (c) G DFS bejárása után nincs visszaél.

A PERT probléma: Input: a $G = (V, E)$ DAG és egy $c : E \to \mathbb{R}_+$ hosszfüggvény.
Output: Minden $v \in V$ csúcsra egy v-be vezető leghosszabb irányított út és annak a hossza.
(Az órai mese szerint a csúcsok „projekttevékenységek", a $c(uv)$ „élhossz" pedig azt mutatja, legalább mennyi időnek kell eltelnie u megkezdése után ahhoz, hogy v elkezdődhessen. Ezért ha a v tevékenység legkorábbi kezdési idejét $k(v)$ jelöli, akkor $k(v) \geq k(u) + c(uv)$ teljesül minden $uv \in E$ élre. Következésképp $k(v)$ minden v csúcs esetén megegyezik a v-ben végződő leghosszabb út hosszával. (A projektmenedzsment szakirodalom máshogyan tekint a feladatra: a csúcsok a projekt mérföldkövei, az élek az egyes projekttevékenységek, $c(e)$ pedig az e tevékenység végrehajtási ideje. Ebben a terminológiában $k(v)$ az a legkorábbi időpont, amikor a v mérföldkő elérhető.))

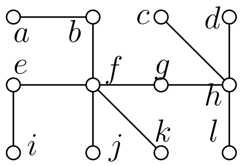
A PERT módszer: Meghatározzuk G egy $v_1, v_2, \ldots, v_n$ topologikus sorrendjét (pl DFS-sel). A $k(v_i) = \max\left(\{k(v_j) + c(v_jv_i) : v_jv_i \in E\} \cup \{0\}\right)$ formulával sora kiszámítjuk a $k(v_1), k(v_2), \ldots$ kezdési időket ill. megjelöljük mindazon v_iv_j éleket, amelyek mentén a maximum eléretik.

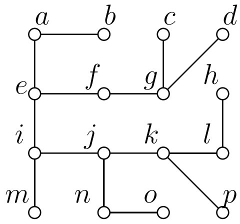
Def: Ha a PERT problémát leíró G gráfnak egyetlen nyelője van, akkor *kritikus út* alatt egy ezen nyelőbe vezető leghosszabb utat értünk. (Több kritikus út is lehet.) A kritikus utak minden élét megjelöltük a PERT módszer során. *Kritikus tevékenység* pedig minden olyan csúcs, ami a nyelőbe vezető kritikus utak valamelyikének csúcsa.

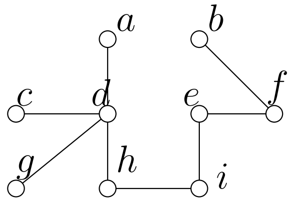
Megfigyelés: Egy tevékenység pontosan akkor kritikus, ha annak megkezdésében a legkisebb mértékű csúszás is a teljes projekt befejezésének késését okozza.

Gyakorlatok

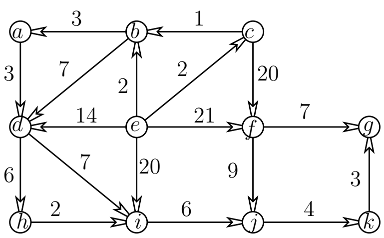
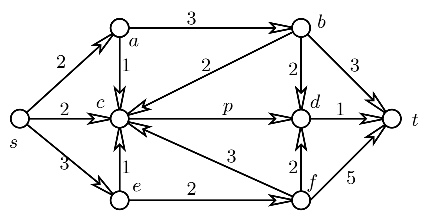
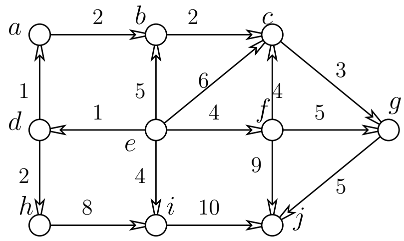
1. Mutassunk példát olyan G gráfra és annak e élére, hogy e keresztél G alkalmas mélységi bejárásánál. (✓)
2. A bal oldali ábrán látható a G gráf egy mélységi fája. Honnan indulhatott a bejárás, ha tudjuk, hogy b és c ill. a és e szomszédosak G-ben? (✓) (ZH '14)







3. A középső ábrán látható a G irányítatlan gráfnak egy i gyökérű DFS fája (azaz egy i-ből indított mélységi bejárása után kapott feszítőfa). Tudjuk, hogy $d_G(e) = 7$. Határozzuk meg a G gráf e-ből induló éleit.
4. Tegyük fel, hogy a jobb oldali ábrán látható F fa a G gráfnak egyszerre h-gyökerű BFS fája és d-gyökerű DFS fája. Legfeljebb hány éle lehet G-nek?
5. Rajzoljunk egy irányított gráfot, végezzük el a mélységi bejárását. Ha a mélységi fa minden élét meg kell hagyni, akkor legalább hány élét kell törölni G-nek, hogy DAG-ot kapjunk? Mik a törlendő élek? Mi a helyzet akkor, ha nem a mélységi fából indulunk ki?
6. Igaz-e, hogy minden aciklikus, irányított G gráf csúcsainak pontosan egy topologikus sorrendje van? (pZH '14)
7. Igaz-e, hogy ha egy n csúcsú, aciklikus, irányított G gráfban van egy $n-1$ élű irányított út, akkor G csúcsainak pontosan egy topologikus sorrendje van? (ppZH '14)
8. Legyen G DAG, és tegyük fel, hogy az u és v csúcsok között egyik irányban sincs irányított út G-ben. Mutassuk meg, hogy G-nek van olyat topologikus sorrendje, amelyben u megelőzi v-t, és olyan is, amelyben v előzi meg u-t. (!)
9. Határozzuk meg az első két ábrán látható PERT problémában a legrövidebb végrehajtási időt és a kritikus tevékenységeket. (✓)

10. A jobb oldali ábrán látható G gráf egyes éleire írt számok azt jelentik, hogy hány kincset tudunk összegyűjteni az adott élen. Határozzuk meg, mennyi az összesen összegyűjthető kincsek száma, ha a gráf tetszőleges pontjából indulhatunk, de csak irányított élek mentén haladhatunk. (!)
11. Adjunk példát olyan PERT feladatra, ahol minden tevékenység kritikus, mégsincs minden tevékenység egy kritikus úton. (✓!)
12. Adjunk olyan eljárást, amely tetszőleges PERT probléma esetén minden tevékenységhez meghatározza azt a legkésőbbi időpontot, amikor az adott tevékenységet elkezdve a teljes PERT feladat legrövidebb idő alatti végrehajtása még éppen nem kerül veszélybe.(!)
13. Adott a PERT problémát leíró G DAG és a G egy $e = uv$ éle. Tudjuk, hogy x összeg kifizetésével a $c(e)$ érték x-szel csökken. (A többi c érték adott, azokra nincs ráhatásunk.) Adjunk olyan eljárást, amelynek segítségével meghatározható az a legkisebb x érték, aminek kifizetésével a PERT feladat a lehető legrövidebb idő alatt végrehajthatóvá válik.(*)