# A számítástudomány alapjai 2016. I. félév

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

## Tudnivalók

**Def:** A $G = (V, E)$ (irányított vagy irányítatlan) gráf egy *bejárásán* a $V$-beli csúcsok végiglátogatását értjük, ahol a alábbiak szerint. Az újonnan elért csúcsot (ha lehetséges) mindig már korábban elért csúcsból kell egy oda vezető él mentén elérni. Ha ez nem lehetséges, és még van eléretlen csúcs, akkor tetszőleges csúccsal lehet a következőnek elért csúcs. (Irányítatlan gráf esetén minden élt oda-vissza irányított élnek tekintünk.) A bejárás során minden csúcsot elérünk egyszer (ez adja az elérési sorrendet), és minden csúcs bejárását befejezzük egyszer, mégpedig akkor, amikor észrevesszük, hogy nem érhető el belőle újabb eléretlen csúcs. Minden csúcshoz megjegyezzük azt is, hogy melyik élen értük el. Ez utóbbi élek alkotják a bejárás *fáját*, élei a bejáráshoz tartozó *faélek*. A fában ősből leszármazottba vezető él az *előreél*, a leszármazottból ősbe vezető a *visszaél*, a többi pedig a *keresztél*.

**Mélységi bejárás:** Input: $G = (V, E)$ (ir) gráf és $v \in V$ Output: a $G$ egy bejárása, azaz
(1) egy *mélységi fa* (a bejárás fája),
(2) minden $u$ csúcshoz egy, az elérési sorrendet meghatározó $m(u)$ *mélységi szám*, valamint $u \neq v$ esetén egy $u$-hoz tartozó mutató az $u$ fabeli *ősére* (arra a csúcsra, ahonnan $u$-t elértük),
(3) minden $u$ csúcshoz egy, a befejezési sorrendet meghatározó $b(u)$ *befejezési szám*,
(4) $G$ éleinek osztályozása. Az $xy$ él faél, ha $y$-hoz $x$-et jegyeztük fel. Az $xy$ él előreél, ha $m(x) < m(y)$ és $b(x) > b(y)$, visszaél, ha $m(x) > m(y)$ és $b(x) < b(y)$, és keresztél, ha $m(x) > m(y)$ és $b(x) > b(y)$.

Működés: Üres veremmel indítunk. Ha üres a verem és $G$-nek van eléretlen $v$ csúcsa, akkor $v$-t elértnek nyilvánítjuk, betesszük a verembe, és $v$ megkapja a soron következő mélységi számot. Ha a verem nemüres és a verem tetején levő $x$ csúcsnak van eléretlen szomszédja (mondjuk $y$), akkor $y$-t a verem tetejére tesszük és elértnek nyilvánítjuk. Az $y$ csúcs megkapja a soron következő mélységi számot és feljegyezzük hozzá az elérését biztosító $xy$ élt. Ha a verem tetején levő $x$-nek nincs eléretlen szomszédja, akkor akkor az $x$ csúcsot befejezzük, $x$ megkapja a soron következő befejezési számot és $x$-et kidobjuk a veremből. Ha a verem kiürült és van még csúcs, akkor tetszőleges eléretlen csúcs kerül a verembe, ha már minden csúcsot elértünk, akkor végül osztályozzuk $G$ éleit.

**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 $Mb(v)$ azzal ér véget, hogy ha van még eléretlen $w$ csúcs, akkor meghívja az $Mb(w)$ eljárást.

**Állítás:** 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.

**Megfigyelés:** Ha $uv$ előreél, akkor $u$ elérésekor $v$ még nem lett elérve, de (mivel $u$ befejezésekor már minden $u$-ból elérhető csúcsot elértünk) $u$-ból vezet $v$-be irányított út a mélységi fában. Ha $uv$ visszaél, akkor van a mélységi fában $vu$ út, ami az $uv$ éllel együtt kört alkot. Ha $uv$ keresztél, akkor $v$ nem leszármazottja $u$-nak, ezért $u$-t be kellett fejezni $v$ elérése előtt.

**Köv.:** Irányítatlan gráf mélységi bejárása után egyetlen él sem lesz keresztél.

**Def:** A $G = (V, E)$ irányított gráf *aciklikus (DAG)*, ha nincs benne irányított kör. A $G$ csúcsainak $v_1, v_2, \ldots, v_n$ sorrendje *topologikus sorrend*, ha él csak kisebb indexű csúcsból futhat nagyobb indexűbe: $v_i v_j \in E \Rightarrow i < j$.

**Megfigyelés:** Ha $G$ DAG, akkor tetszőleges mélységi bejárásában a csúcsok befejezési sorrendjének megfordítása topologikus sorrend.

**Köv.:** Tetszőleges $G$ irányított gráfra ekvivalensek az alábbiak: (1) $G$ DAG, (2) $G$ mélységi bejárásakor nem keletkezik visszaél (3) $G$ csúcsainak van topologikus sorrendje.

**A PERT probléma:** Input: a $G = (V, E)$ DAG és egy $c : E \to \mathbb{R}_+$ hosszfüggvény.
Output: Minden $v \in V$ csúcsra a $v$-be vezető leghosszabb irányított út hossza.

(A szokásos mesében az egyes csúcsok a projektbeli „tevékenységek”, az élhosszok pedig azt mutatják, legalább mennyi időnek kell eltelnie a két adott tevékenység megkezdése között. Az output az egyes tevékenységek legkorábbi kezdési idejét adja meg, aholis minden $uv \in E$ élre $k(v) \geq k(u) + c(uv)$ teljesül.)

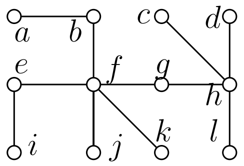
**A PERT módszer:** Meghatározzuk $G$ egy $v_1, v_2, \dots, v_n$ topologikus sorrendjét (pl DFS-sel). A $k(v_i) = \max\{0, \max\{k(v_j) + c(v_j v_i) : v_j v_i \in E\}\}$ formulával sora meghatározzuk a $k(v_1), k(v_2), \dots$ kezdési időket ill. megjelöljük mindazon $v_i v_j$ éleket, amelyek a maximumot adják.

**Def:** Ha a PERT problémát leíró $G$ gráfnak egyetlen nyelője van, akkor *kritikus út* alatt az 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.

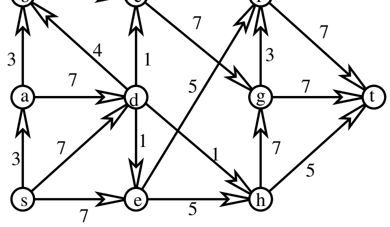
**Megfigyelés:** Egy tevékenység pontosan akkor kritikus, ha annak megkezdésében a legkisebb mértékű csúszás is a teljes projekt befejezésének késését okozza.

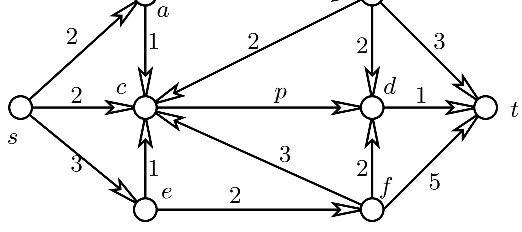
## Gyakorlatok

1. Rajzoljunk irányított gráfot, adjunk az éleknek hosszokat és szélességeket, valamint jelöljünk ki egy $r$ gyökérpontot. Keressünk $r$-ből minden más csúcsba legszélesebb utat a Dijkstra algoritmus célszerűen módosított változatával. Adott $u$ és $v$ csúcsokra döntsük el, érdemes-e egy kellően nagy kapacitású $uv$ élt bevenni a gráfba ahhoz, hogy legalább egy olyan $x$ pont legyen $G$-ben, amire a legszélesebb $rx$-út szélessége így megnövekszik. Ha érdemes az $uv$ élt megépíteni, határozzuk meg azt a legnagyobb $w$ szélességet, amit értelmes adni ennek az új élnek.
2. Mutassunk példát olyan $G$ gráfra és annak $e$ élére, hogy $e$ keresztél $G$ alkalmas mélységi bejárásánál.
3. Az 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)




4. Legyenek a 7 csúcsú $G$ gráf pontjai $v_1, v_2, v_3, v_4, v_6, v_8$ és $v_9$, valamint akkor legyen $v_i$ és $v_j$ szomszédos, ha $i$ és $j$ relatív prímek. Ekkor a $v_i v_j$ él szélessége $|i - j|$. Határozzunk meg a $v_1$ csúcsból minden más csúcsba egy-egy legszélesebb utat. (ZH '14)
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? Határozzuk meg a csúcsok befejezési számozását is. 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? (ppZH '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 alábbi PERT problémákban a legrövidebb végrehajtási időt és a kritikus tevékenységeket. Mik az egyes tevékenységekre a legutolsó időpontok, amikor azokat elkezdve a projekt még épp időben végrehajtható?




10. Adjunk példát olyan PERT feladatra, ahol minden tevékenység kritikus, mégsincs minden tevékenység egy kritikus úton.
11. 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.