Algoritmusok és adatszerkezetek 2.

Fekete István jegyzetéből (részlet)

9. gyakorlat

Forrás: http://people.inf.elte.hu/fekete/ Algoritmusok 2 / gráfalgoritmusok

1. *Mélységi bejárás és alkalmazásai*

1.1 Mélységi bejárás

Az **algoritmus elvét** a *Bejárási/keresési stratégiák* című fejezetben már láttuk, most foglaljuk össze röviden. Egy kezdőpontból kiindulva addig megyünk egy él mentén, ameddig el nem jutunk egy olyan csúcsba, amelyből már nem tudunk tovább menni, mivel nincs már meg nem látogatott szomszédja. Ekkor visszamegyünk az út utolsó előtti csúcsához, és onnan próbálunk egy másik él mentén tovább menni. Ha ezen az ágon is minden csúcsot már bejártunk, ismét visszamegyünk egy csúcsot, és így tovább.

Most vizsgáljuk meg a bejárást **<u>ADS szinten</u>** egy példán:

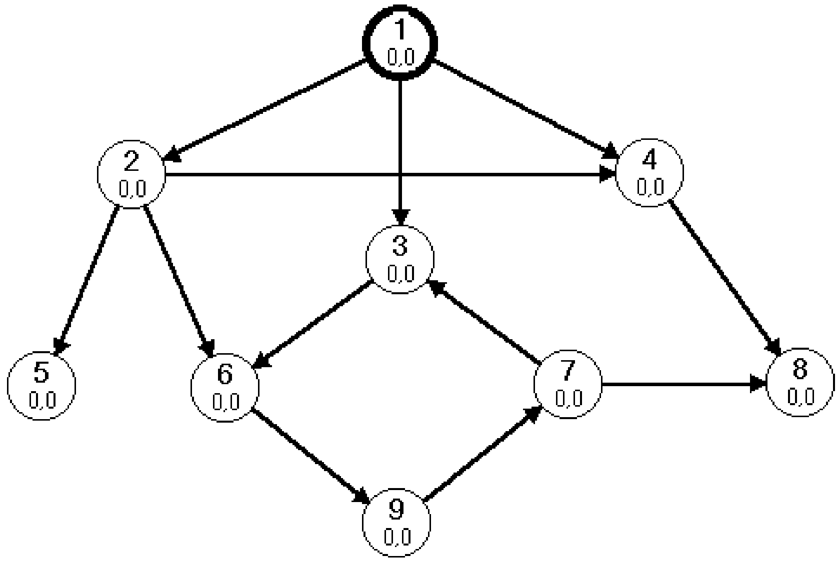
Színezzük a csúcsokat attól függően, hogy az illető csúcsra vonatkozóan a bejárás milyen fázisban van.

- Egy csúcs legyen fehér, ha még nem jutottunk el hozzá a bejárás során (kezdetben minden csúcs fehér).
- Egy csúcs legyen szürke, ha a bejárás során már elértük a csúcsot, de még nem állíthatjuk, hogy az illető csúcsból elérhető összes csúcsot meglátogattuk.
- A csúcs legyen fekete, ha azt mondhatjuk, hogy az illető csúcsból elérhető összes csúcsot már meglátogattuk és visszamehetünk (vagy már visszamentünk) az idevezető út megelőző csúcsára

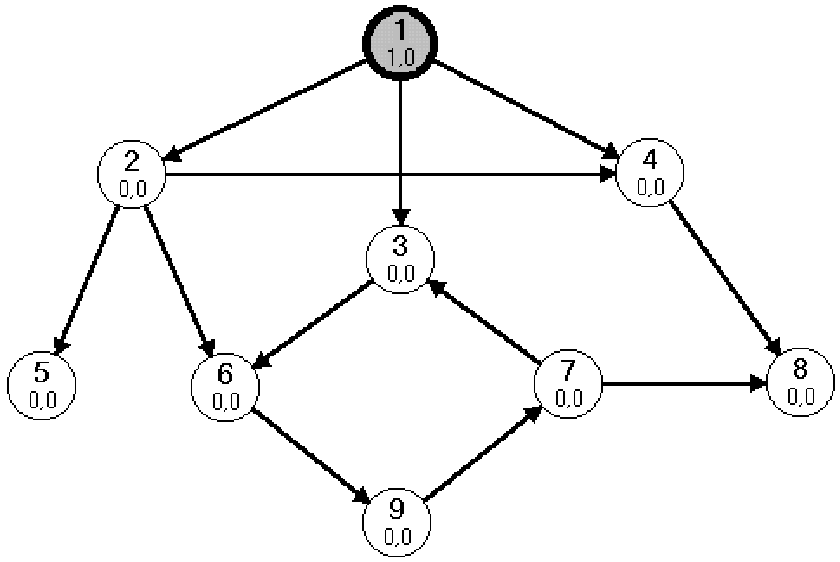
A bejárás során tároljuk el, hogy egy csúcsot hányadikként értünk el, azaz hányadikként lett szürke és tároljuk el, hogy hányadikként fejeztük be a csúcs, és a belőle elérhető csúcsok bejárását, azaz a csúcs hányadikként lett fekete. Az említett számokat nevezzük mélységi, illetve befejezési számnak és az ábrákon a csúcsok címkéi alatt fogjuk megjeleníteni.

A példában egy csúcsból kimenő élek feldolgozási sorrendje legyen a szomszéd csúcsok címkéje szerint növekedően rendezett (pl.: láncolt ábrázolásnál az éllista a csúcsok címkéje szerint rendezett).

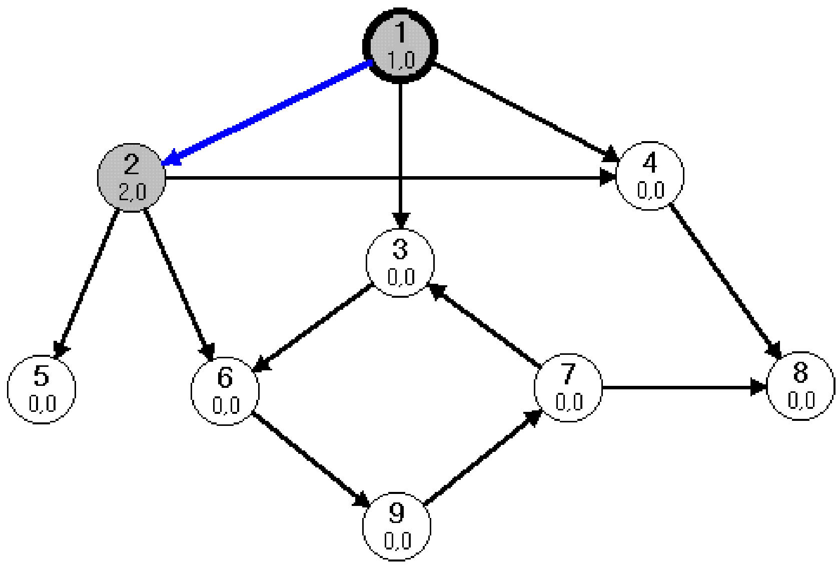
Tehát nézzük az alábbi példát, amelyben a kezdőcsúcs legyen az 1-es csúcs. Legyen kezdetben minden csúcs fehér, és a mélységi és befejezési számuk is legyen az extremális 0.


1
0,0
2
0,0
3
0,0
4
0,0
5
0,0
6
0,0
7
0,0
8
0,0
9
0,0

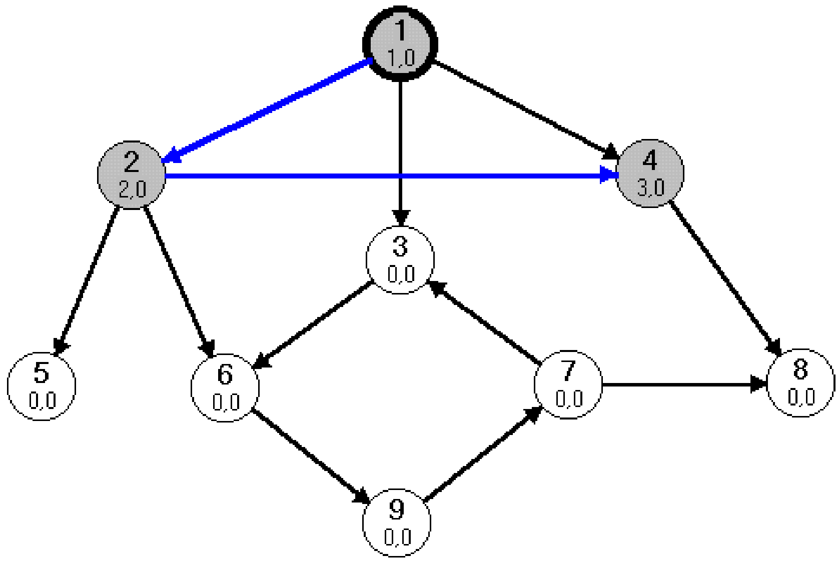
A kezdőcsúcsot érjük el elsőként, tehát színezzük szürkére, és a mélységi számát állítsuk be 1-re



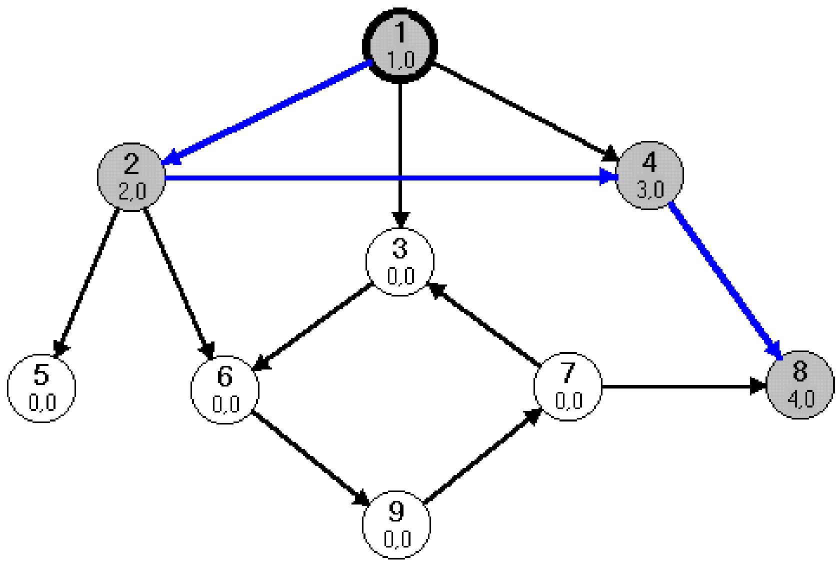
Az 1-es csúcsból három él vezet ki, azaz három él mentén indulhatnánk el, de a kikötöttük feltételként, hogy az élek feldolgozási sorrendje legyen a szomszéd csúcsok címkéje szerint növekedően rendezett. Tehát a 2-es csúcsot érjük el másodikként.



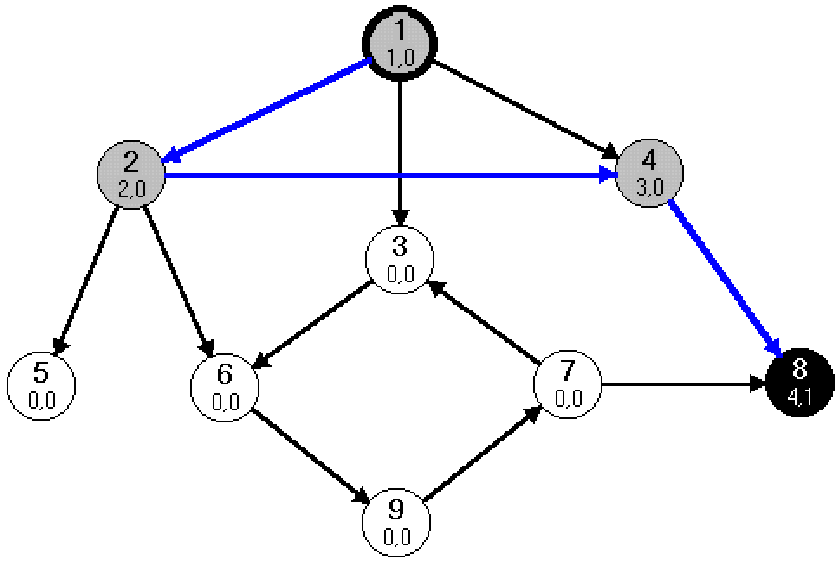
Ezután elérjük harmadikként a 4-es csúcsot.


1
1,0
2
2,0
4
3,0
3
0,0
5
0,0
6
0,0
7
0,0
8
0,0
9
0,0

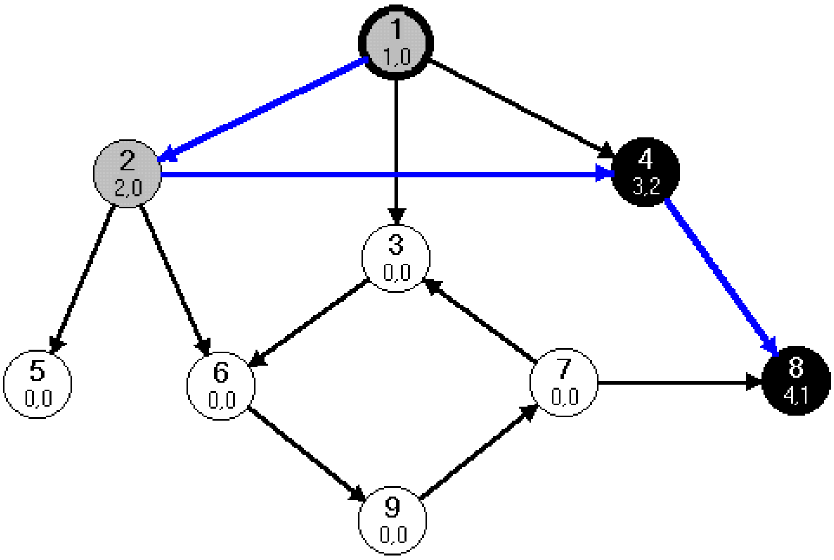
Negyedikként a 8-as csúcsot érjük el.



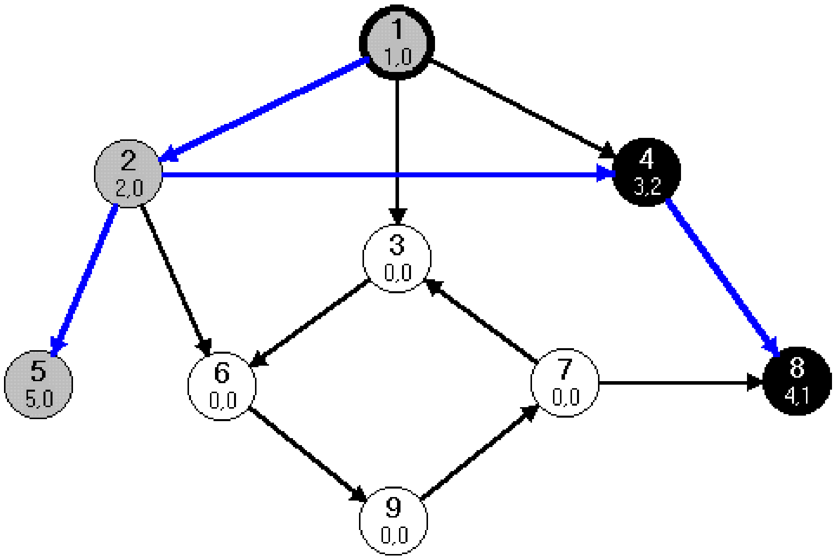
Mivel a 8-as csúcsnak nincs olyan szomszédja, amit még nem látogattunk volna meg (nincs egyáltalán szomszédja), a 8-as csúcs bejárását befejeztük, a csúcsot színezzük feketére. Mivel a bejárás során a 8-as csúcs lett elsőként fekete, így az ő befejezési száma legyen az egyes.



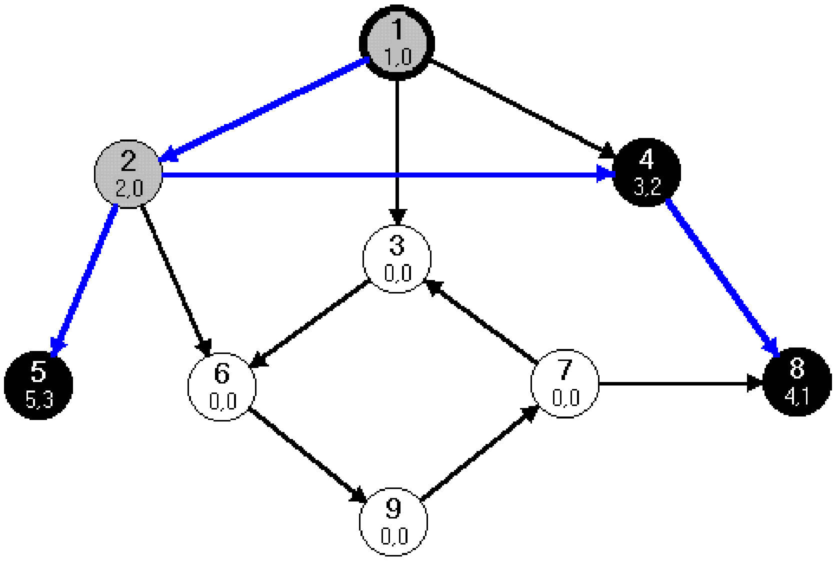
A bejárás során a megtett utunk $\langle 1,2,4,8 \rangle$. Most menjünk vissza az utolsó előtti csúcsra, a 4-es csúcsra. Mivel a 4-es csúcsnak sincs még meg nem látogatott szomszédja, így a 4-es csúcs bejárását is befejeztük, színezzük a csúcsot feketére, és a bejárási számát állítsuk be kettőre.


1
1,0
2
2,0
4
3,2
3
0,0
5
0,0
6
0,0
7
0,0
8
4,1
9
0,0

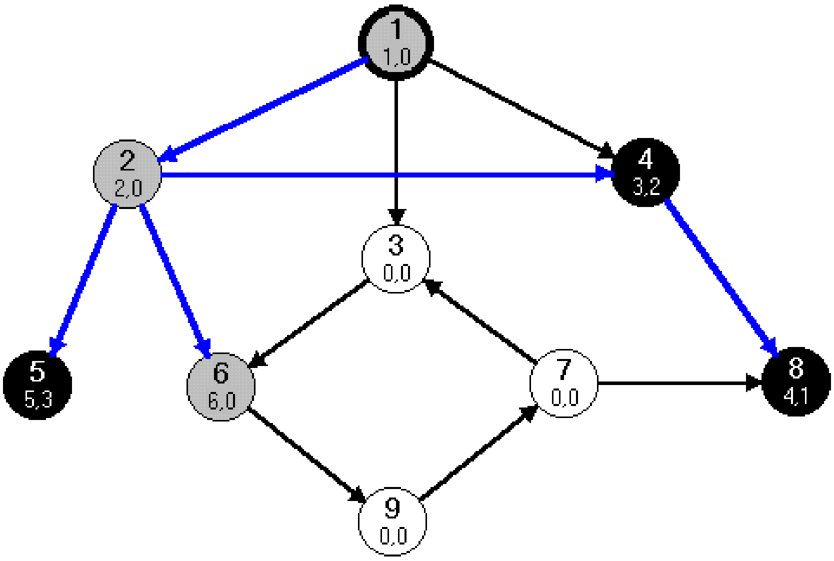
Menjünk vissza a 2-es csúcshoz. A 2-es csúcsnak két olyan szomszédja is van, amelyet még nem látogattunk meg. Látogassuk meg a kisebb címkéjű csúcsot.



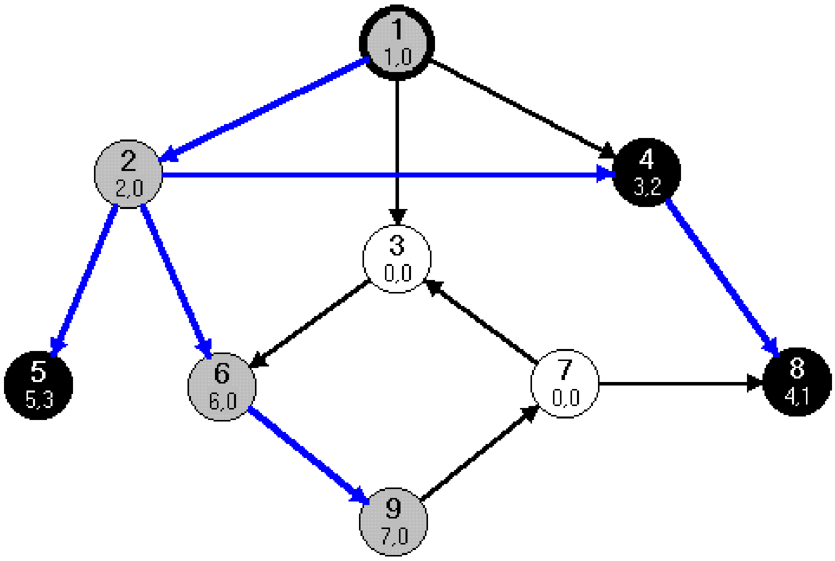
Az 5-ös csúcs bejárását harmadikként befejeztük.



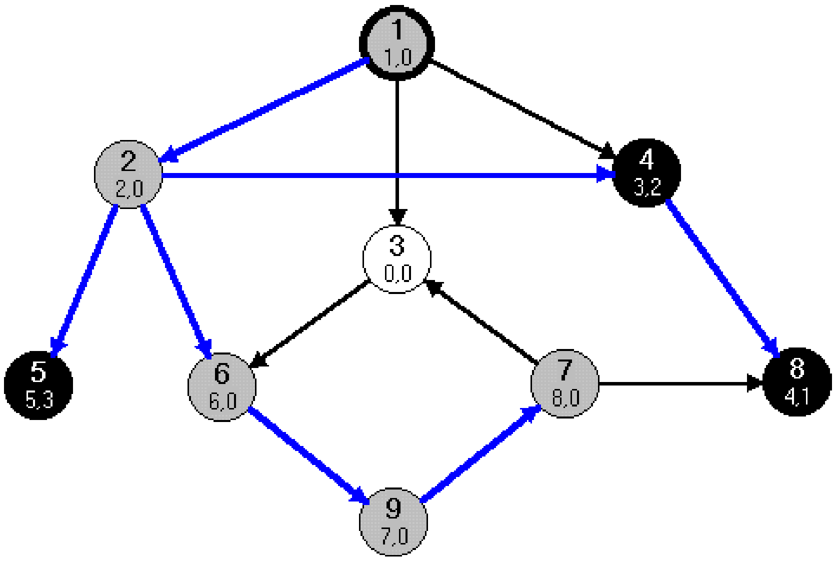
A 2-es csúcsból a bejárást a 6-os csúcs irányába folytatjuk.


1
1,0
2
2,0
3
0,0
4
3,2
5
5,3
6
6,0
7
0,0
8
4,1
9
0,0

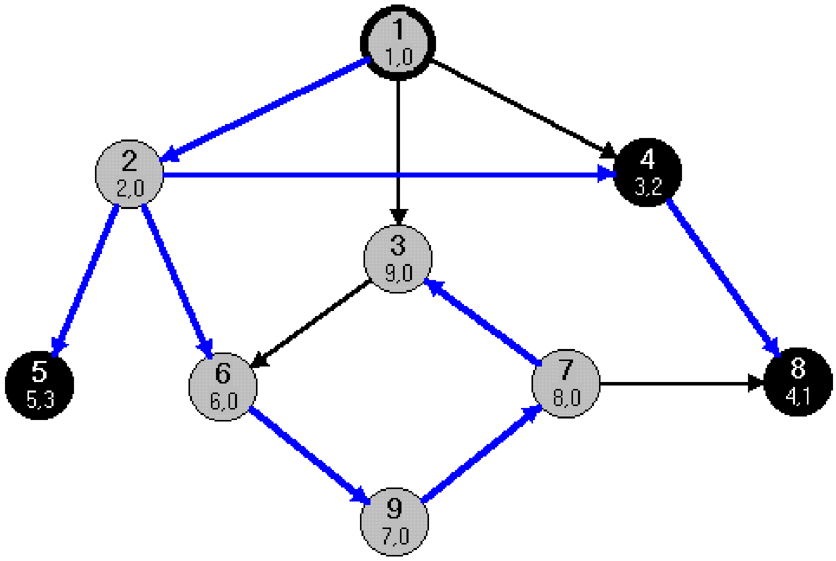
Tovább haladva hetedikként elérjük a 9-es csúcsot.



Nyolcadikként a 7-es csúcsot érjük el.

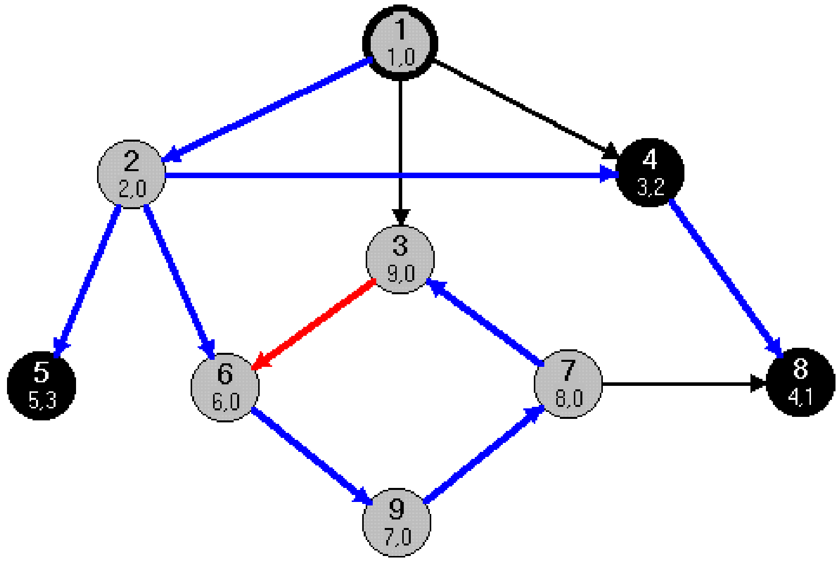


Majd a 7-es csúcsból elsőként megvizsgáljuk a 3-as csúcsba vezető él mentén a lehetőségeket. Mivel a 3-as csúcs még fehér, kilencedikként elérjük a 3-as csúcsot.

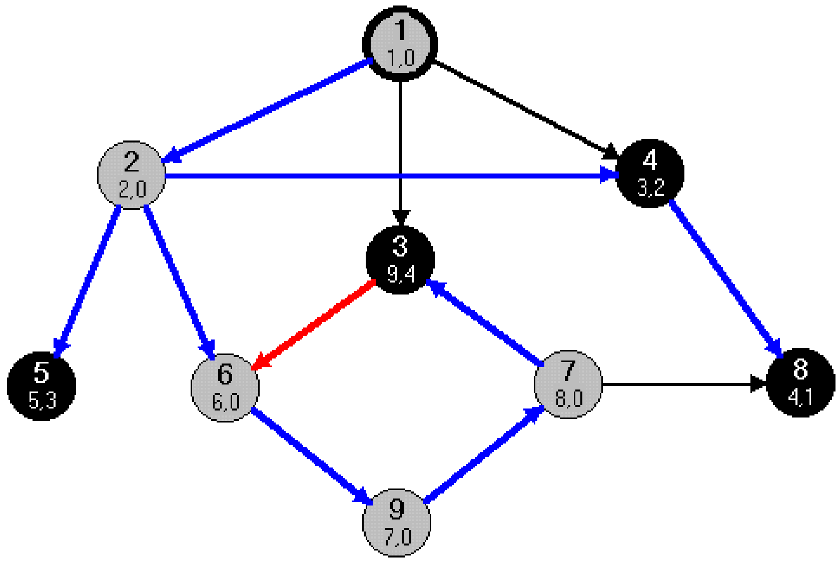


1
1,0
2
2,0
3
9,0
4
3,2
5
5,3
6
6,0
7
8,0
8
4,1
9
7,0

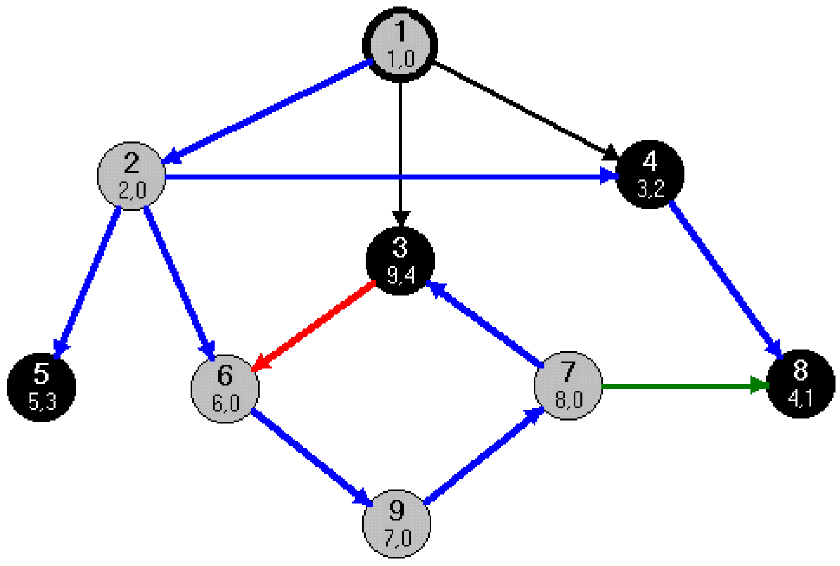
A 3-as csúcsból a 6-os csúcsba vezet él, azonban a 6-os csúcsot már meglátogattuk, azaz a színe már nem fehér, azaz erre már nem folytatjuk a bejárást (különben a körön végtelen sokáig keringhetnénk).



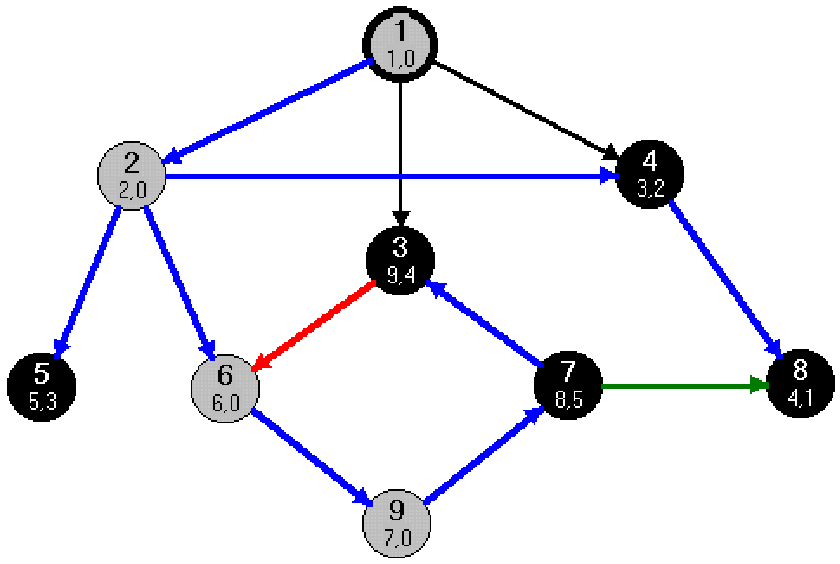
Mivel a 3-as csúcsból már nem vezet él még meg nem látogatott csúcsba, így a 3-as csúcs bejárását is befejeztük.



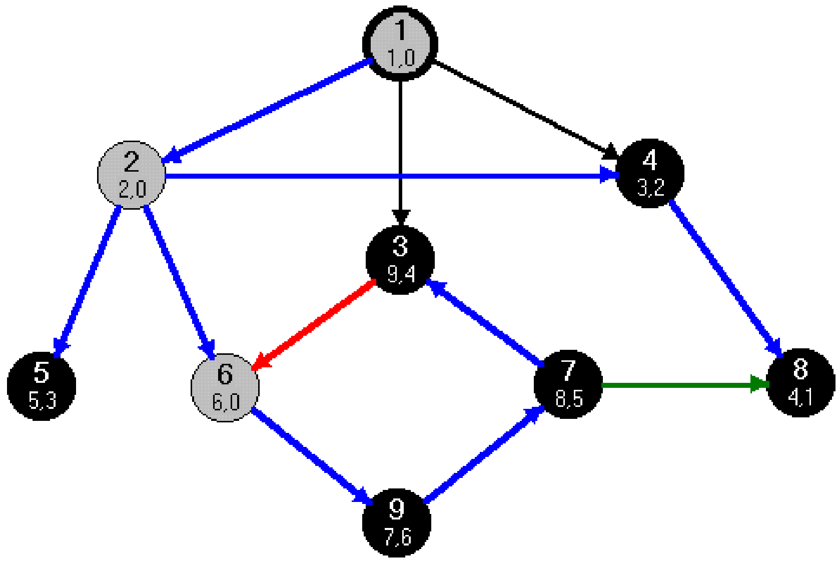
Visszamegyünk a 7-es csúcsba, ahol a sorrendben következő él, a (7,8) mentén vizsgálódunk. Azonban a 8-as csúcs színe már fekete, tehát már befejeztük a bejárását, így erre már felesleges volna mennünk.


1
1,0
2
2,0
3
9,4
4
3,2
5
5,3
6
6,0
7
8,0
8
4,1
9
7,0

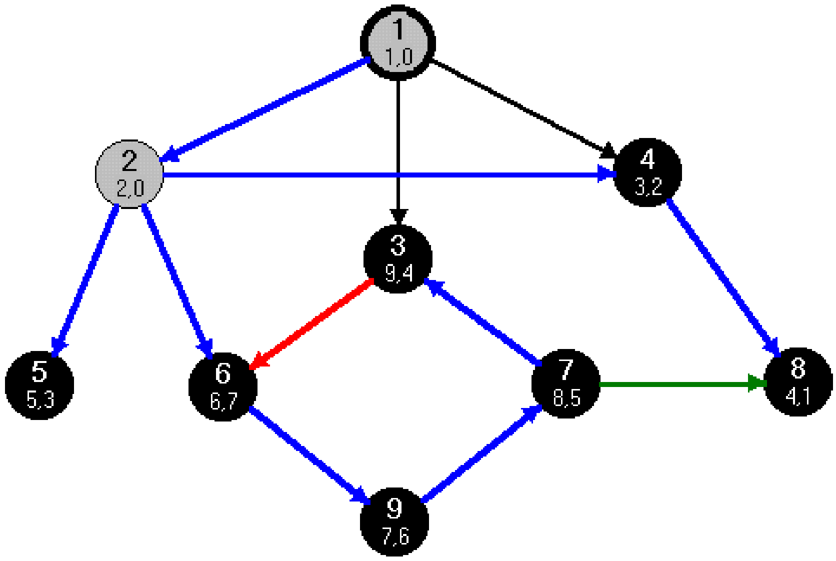
Mivel a 7-es csúcsnak nincs fehér szomszédja, így ötödikként befejeztük a bejárását.



A 9-es csúcsnak a bejárását is befejeztük.

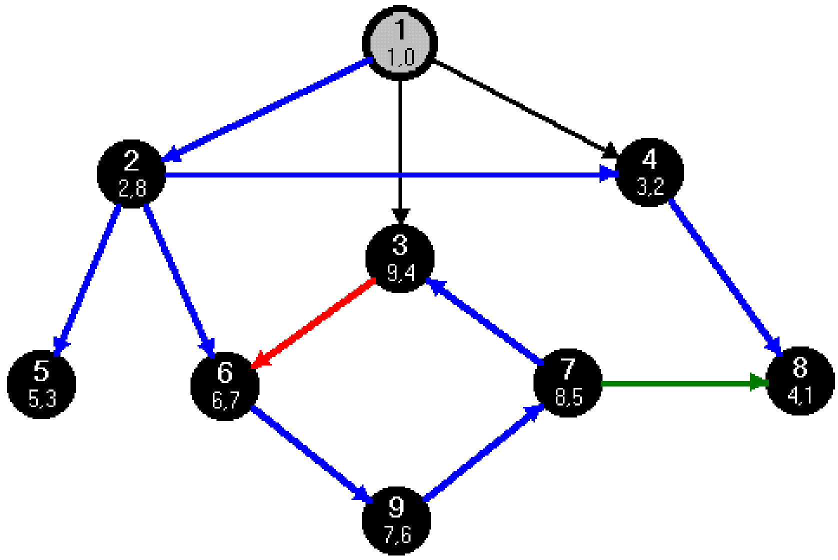


Az úton ismét egy csúccsal visszamegyünk és befejezzük a 6-os csúcsot is.

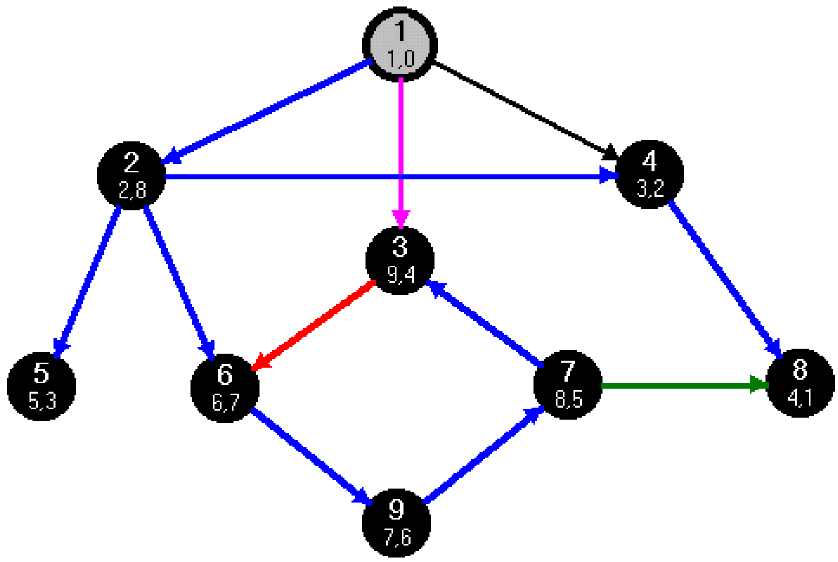


1
1,0
2
2,0
3
9,4
4
3,2
5
5,3
6
6,7
7
8,5
8
4,1
9
7,6

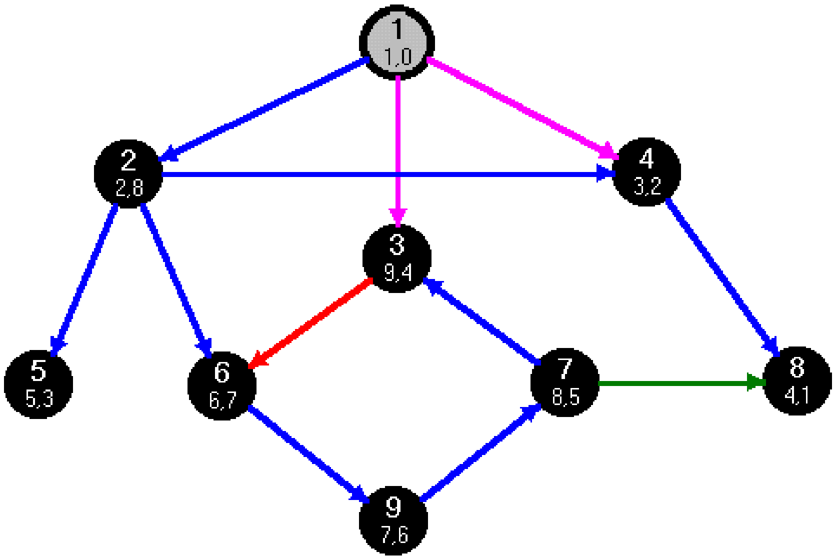
A 2-es csúcsra lépve, látható, hogy minden kimenő éle mentén már próbálkoztunk, így nyolcadikként befejezzük őt is.



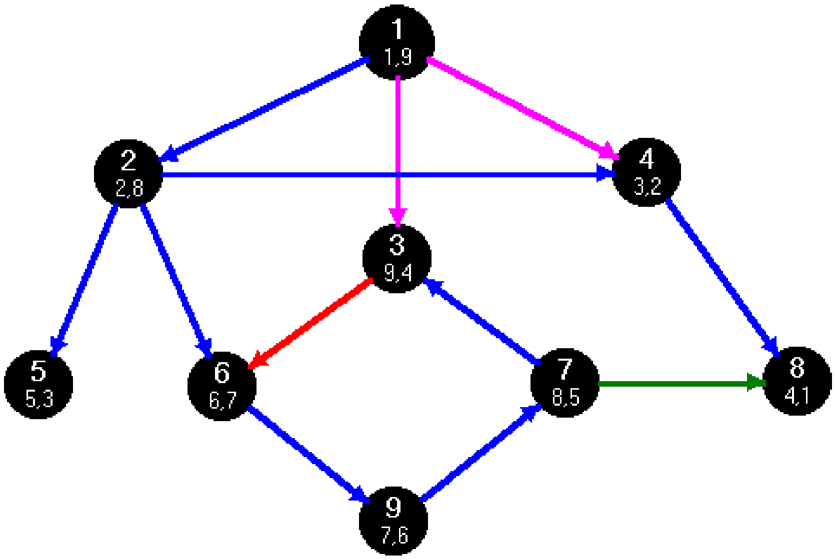
Az 1-es csúcsra lépve megvizsgáljuk a 3-as csúcsot, de látva, hogy színe nem fehér, arra nem megyünk tovább.



Ezután az előzőhöz hasonlóan még megvizsgáljuk a maradék (1,4) él mentén a 4-es csúcsot, de annak színe sem fehér.


1
1,0
2
2,8
3
9,4
4
3,2
5
5,3
6
6,7
7
8,5
8
4,1
9
7,6

Végül az 1-esből kimenő összes él mentén már megvizsgáltuk a bejárási lehetőségeket, így az 1-es csúcsot is befejeztük utolsóként.

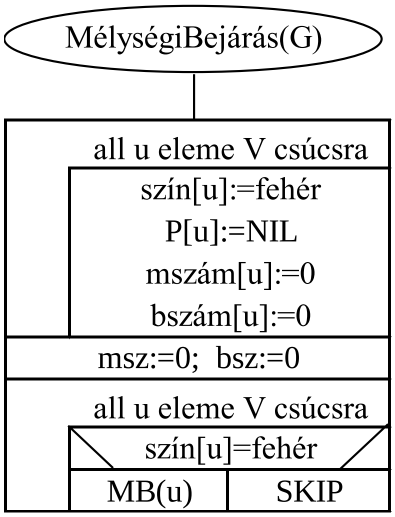


Az algoritmus **<u>ADT szintű</u>** leírása:

Legyen *G=(V,E)* irányított vagy irányítatlan, véges gráf, ahol *V={1,2,…,n}*. Továbbá definiáljuk az alábbi vektorokat:

- *szín*[*1..n*]: az ADS színtű színezés megvalósítására
- *mszám*[*1..n*] és *bszám*[*1..n*]: az ADS szinten említett mélységi és befejezési számok nyilvántartására
- *P*[*1..n*]: a bejárás során, egy csúcs megelőző csúcsának a nyilvántartására (a korábban látottakhoz hasonlóan, pl.: szélességi bejárás vagy Dijkstra algoritmus).

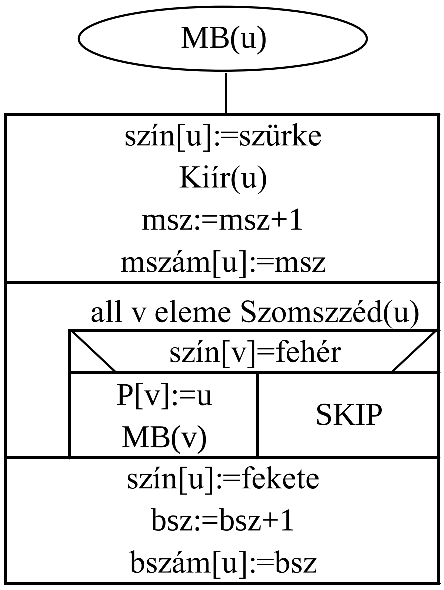
Az előző példában úgy kezdtük a bejárást, hogy kijelöltünk egy kezdőcsúcsot és a kezdőcsúcsból elérhető csúcsokat (a példában az összes csúcs ilyen volt) jártuk be. Egy másik példán előfordulhatna, hogy lennének olyan csúcsok, amelyeket egyáltalán nem látogatnánk meg (a szélességi keresés is így működik). Azonban a későbbi alkalmazások érdekében, a bejárásunk legyen olyan, hogy minden csúcsot meglátogatunk. Tekintsünk egy kezdőcsúcsot, és innen indítsunk el egy bejárást. Miután visszajutottunk az említett kezdőcsúcsba (azaz a kezdőcsúcsból elérhető csúcsokat bejártuk), nem fejezzük be az algoritmust, hanem keresünk egy eddig még meg nem látogatott (azaz fehér) csúcsot és innen újra indítjuk a bejárást. Ezt eddig folytatjuk, amíg van fehér csúcsunk. Nyilván minden ilyen menetben legalább egy csúcsot átszínezünk feketére, tehát véges számú menet után elfogynak a fehér csúcsok. Elegendő a csúcsok halmazán egyszer végigmenni (a gyakorlatban a csúcsok címkéje szerinti növekedően), és ha egy csúcs színe fehér, akkor onnan indítsunk egy bejárást. Tehát az algoritmust bontsuk két részre. Az egyik része egy kezdőcsúcsból indítja a bejárást, ez lesz az *MB(u)* eljárás, a másik pedig végigmegy a csúcsok halmazán, és egy fehér csúcsot találva elindítja az előbb említett eljárást. A kezdőcsúcsból induló mélységi bejárásra (*MB(u)*) egyszerű rekurzív definíciót adni, miszerint az *u* csúcsot akkor jártuk be, ha az összes szomszédját már bejártuk (vagy legalább elértük, a kör miatti végtelen ciklus elkerülése). Tehát az *MB(u)* eljárást most rekurzíven adjuk meg.



Inicializáló lépés: kezdetben minden csúcs fehér, a nincs megelőzője, és a mélységi és befejezési száma 0.

Globális számlálok a mélységi és befejezési számok meghatározására

A csúcsokat végignézve, a fehér csúcsokból indítjuk a rekurzív bejáró eljárást.

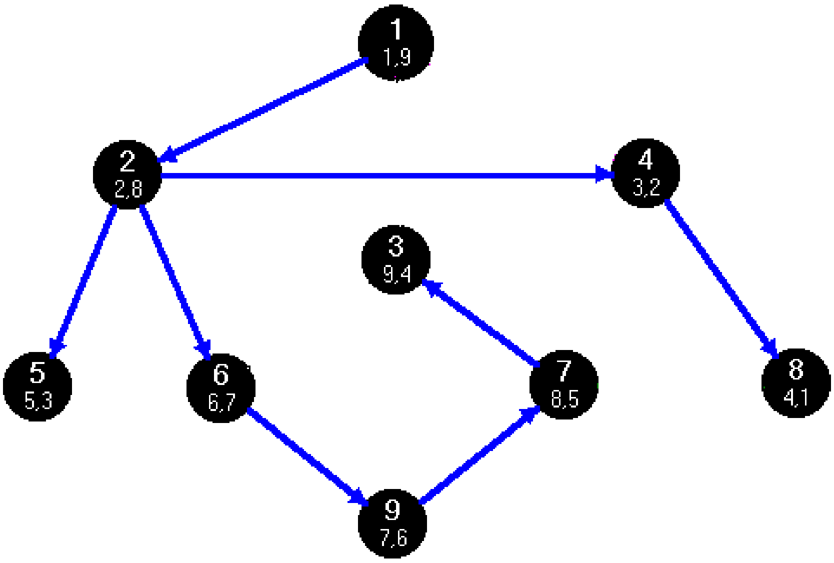


Itt látogatjuk meg (érjük el először) az u -t, átszínezzük fehérről szürkére, és a mélységi számát is beállítjuk.

Az u összes még meg nem látogatott szomszédjára feljegyezzük megelőző csúcsként az u -t, és elidítjuk a bejárást.

Befejeztük az u bejárását, beszínezzük feketére, és a befejezési számát is beállítjuk.

Az *MB(u)* eljárás futása során feljegyezzük a *P* tömbbe egy csúcs megelőzőjét, így egy *u* csúcsból kiinduló, úgynevezett **mélységi fát** kapunk. Az ADS szinten említett példában az 1-es csúcsból kiinduló mélységi faként az alábbi gráfot kaptuk:



Összességében, pedig a *P* tömbben feljegyzett megelőzési reláció, *G* egy részgráfját adja, amelyet **mélységi erdőnek** nevezünk.

<u>8.1.1. Definíció:</u> Legyen *G=(V,E)* irányított, véges gráf. A *G* mélységi bejárása után a *P*-ben keletkező megelőzési reláció által ábrázolható irányított *T* részgráfot, a *G* egy **mélységi erdőjének** nevezzük.

<u>8.1.2. Definíció:</u> Legyen *T* a *G=(V,E)* gráf egy mélységi erdője, és $x, y \in V$ csúcsok. Az *y* **leszármazottja** *x*-nek *T*-ben, ha $\exists\, x \leadsto y$ *T*-beli irányított út.

<u>8.1.3. Állítás:</u> (**szükséges feltétel**) Legyen *T* a *G=(V,E)* gráf egy mélységi erdője, $x, y \in V$ csúcsok, $x \neq y$, és *y* **leszármazottja** *x*-nek *T*-ben, akkor *mszám*[*y*]>*mszám*[*x*].

<u>Bizonyítás:</u> *y* **leszármazottja** *x*-nek, tehát $\exists\, x \leadsto y$ út *T*-ben. Ez az út úgy keletkezik, hogy az út *(u,v)* éleinek vizsgálatakor, *szín*[*v*]=*fehér* csúcsok esetén a *P* tömbbe bejegyzésre kerül a megelőző *u* csúcs, majd meghívjuk az *MB(v)* eljárást, ahol a *v* csúcs legalább egyel megnövelt mélységi szám értéket fog kapni.

<u>8.1.4. Állítás:</u> (**szükséges és elégséges feltétel**) Legyen *T* a *G=(V,E)* gráf egy mélységi erdője, $x, y \in V$ csúcsok, $x \neq y$. Ha *mszám*[*y*]>*mszám*[*x*] és *bszám*[*y*]<*bszám*[*x*] $\Leftrightarrow$ *y* **leszármazottja** *x*-nek *T*-ben.

<u>Bizonyítás:</u>

A mélységi és befejezési számok viszonyából következik, hogy előbb meghívtuk az *MB(x)* eljárást, majd még mielőtt végett ért volna az *x* szomszédainak rekurzív bejárását végző ciklus, meghívtuk *MB(y)*-t, amely teljes egészében lefutott, majd csak ez után terminálhatott az említett ciklus. Ez akkor lehetséges, ha létezik egy olyan $z \in V$ csúcs, hogy az *MB(z)* eljárásban hívtuk meg *MB(y)*-ot, és *P*[*y*]=*z* bejegyzésre kerül. Továbbá a *z* csúcs olyan, hogy *z=x* (azaz *y* szomszédja *x*-nek) vagy *MB(z)* lefutása is teljesül, amit az *MB(y)*-ról az előbb említettünk. Ezt a gondolatmenetet addig folytathatjuk, míg *z=x* nem lesz, miközben láthatjuk, hogy a *P* tömb bejegyzései által reprezentált irányított úton haladunk visszafelé (a rekurzív hívási fán haladunk felfelé). Tehát ez az út igazolja, hogy *y* leszármazottja *x*-nek *T*-ben.

<u>8.1.5. Tétel (Fehér út tétel):</u> Legyen *T* a *G=(V,E)* gráf egy mélységi erdője, $x, y \in V$ csúcsok, $x \neq y$. *y* **leszármazottja** *x*-nek *T*-ben. $\Leftrightarrow$ *x* elérésekor az *y* elérhető az *x*-ből, az *x* kivételével csak fehér csúcsokat tartalmazó úton.

<u>Bizonyítás:</u>

<u>$\Rightarrow$</u>: *y* leszármazottja *x*-nek $\Rightarrow$ $\exists x \leadsto y$ *T*-beli út, amelynek mentén minden csúcs leszármazottja *x*-nek. Ezen fabeli út minden *u* csúcsára teljesül: *mszám*[*u*]>*mszám*[*x*] (8.1.3.

szükséges feltétel), azaz minden *u* csúcsot később érünk el, mint az *x*-et $\Rightarrow$ *x* elérésekor az *u* leszármazott csúcsok még fehérek.

$\underline{\Leftarrow}$: Tegyük fel, hogy *x* elérésekor létezik *y*-ba vezető fehér csúcsokból álló út, legyen *v* egy ilyen út első olyan csúcsa, amelyet az *x* szomszédait felsoroló ciklusban elérünk, azaz meghívjuk az *MB(v)* eljárást. Mivel *x* már szürke és *v* még fehér $\Rightarrow$ *v*-t később érjük el, mint *x*-et, tehát *mszám*[*v*]>*mszám*[*x*]. Továbbá *MB(x)* belsejéből hívtuk meg *MB(v)*-t tehát *MB(v)* előbb lefut, mint *MB(x),* azaz *bszám*[*v*]<*bszám*[*x*]. Tehát a 8.1.4. állítás szerint *v* leszármazottja *x*-nek. Azonban feltettük, hogy *v*-t érjük el először az *y*-hoz vezető úton, tehát az út többi csúcsa *v* elérésekor még fehér, így a $v \leadsto y$ útra hasonlóan alkalmazható a fenti rekurzív gondolatmenet (míg el nem jutunk az $y \leadsto y$ útig). Miután beláttuk, hogy *v* leszármazottja *x*-nek és *y* leszármazottja *v*-nek$\Rightarrow$ *y* leszármazottja *x*-nek.

<u>Megjegyzés:</u> ha *x*=*y*, akkor triviálisan teljesül a kölcsönös leszármazottság.

<u>8.1.6. Definíció:</u> Az élek osztályozása egy adott mélységi bejárás szerint.

Legyen *T* a *G=(V,E)* gráf egy mélységi erdője, és $x, y \in V$ csúcsok. Az $(x, y) \in E$ él

a) faél, ha $x \to y$ éle *T*-nek.
b) előreél, ha $x \to y$ nem éle *T*-nek, de *y* leszármazottja *x*-nek *T*-ben, és $x \neq y$.
c) visszaél, ha *x* leszármazottja *y*-nak T-ben ($x \to y$ nem éle *T*-nek; ide tartozik *x*=*y* hurokél is).
d) keresztél, ha *x* és *y* nem leszármazottai egymásnak T-ben.

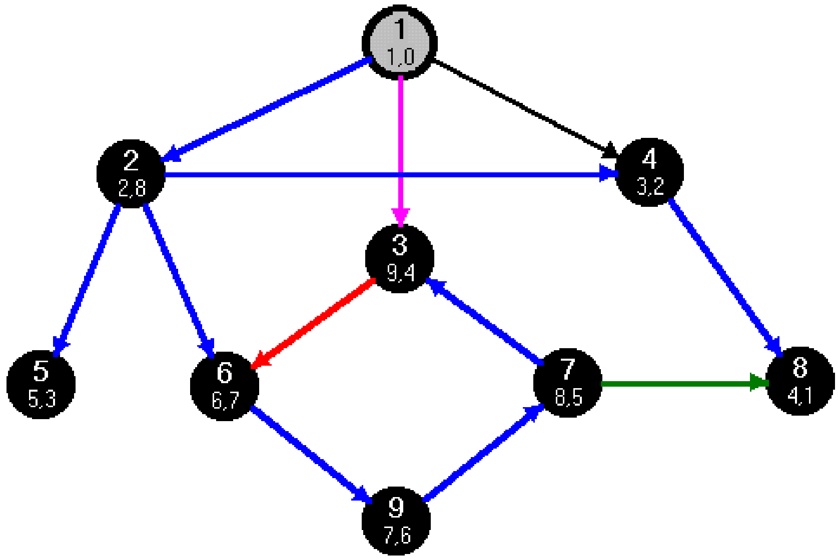
<u>8.1.7. Tétel:</u> Az éltípusok azonosítása a bejárás során.

Tegyük fel, hogy *G=(V,E*) irányított véges gráf mélység bejárása során éppen az $(x, y) \in E$ vizsgálatánál tartunk. Ekkor (x, y) él

a) faél, ha $mszám[y] = 0$.
b) előreél, ha $mszám[y] > mszám[x]$.
c) visszaél, ha $mszám[y] \leq mszám[x]$ és $bszám[y] = 0$.
d) keresztél, ha $mszám[y] < mszám[x]$ és $bszám[y] > 0$.

<u>Bizonyítás:</u>

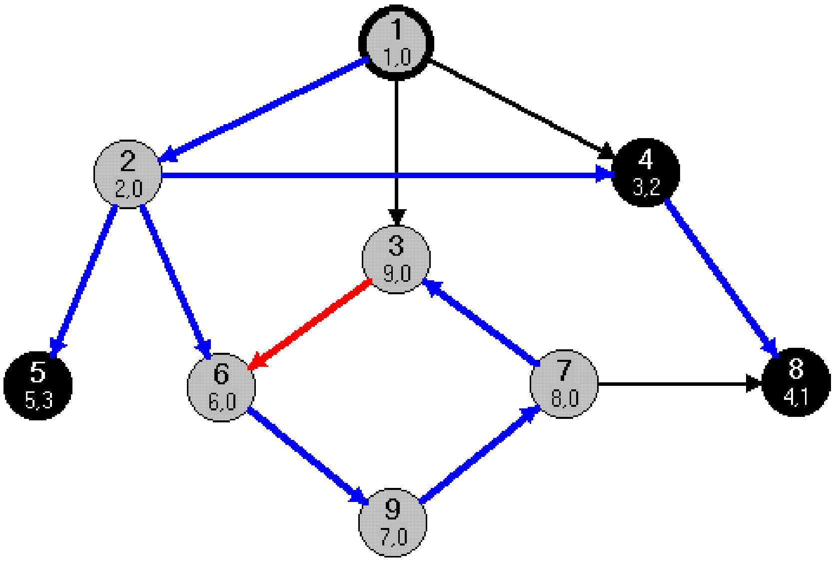
a) $mszám[y] = 0$ azt jelenti, hogy most érjük el az *y* csúcsot, azaz a csúcs még fehér. Továbbá az *(x,y)* él mentén jutunk az *y* csúcsba, mint *x* szomszédja, amely az *MB(x) x* szomszédait felsoroló ciklusában, az elágazás bal oldali ágában lehetséges, ahol valóban a *P* tömbbe bejegyzésre kerül az él, tehát faél lesz.
b) $mszám[y] > mszám[x]$ és *mszám*[*x*]>*0*, így *mszám*[*y*]>*0*, tehát *y* nem fehér (így az él nem lehet faél). Továbbá a mélységi számok viszonyából következik, hogy az *MB(y)*-t később hívtuk meg, mint az *MB(x)*-t, de az *MB(x)* még nem fejeződött be. Így az *MB(y)* meghívása, az *MB(x)* eljárás *x* szomszédait felsoroló ciklusában, valamely szomszédra meghívott *MB* eljárásban, vagy annak rekurzív leszármazottjában történt, az *(x,y)* él vizsgálata előtt. Azonban az *(x,y)* él vizsgálata a ciklus egy későbbi iterációjában történik, tehát az *MB(y)*-nak mostanra be kellet fejeződnie, de az *MB(x)* még csak ezek után fog befejeződni$\Rightarrow$ *bszám*[*y*]<*bszám*[*x*]. A 8.1.4. állítást felhasználva beláthatjuk, hogy *y* leszármazottja *x*-nek *T*-ben, és láttuk, hogy nem lehet faél, tehát előreél.
Láthatjuk az ADS szinten vizsgált példán, hogy az 1-es csúcsból a 3-as csúcsba vezető él feldolgozásakor a 3-as csúcshoz már eljutottunk az $\langle 1,2,6,9,7,3 \rangle$ úton, tehát a 3-as leszármazottja az 1-nek, és az (1,3) él nem faél, tehát előreél.


1
1,0
2
2,8
3
9,4
4
3,2
5
5,3
6
6,7
7
8,5
8
4,1
9
7,6

c) $mszám[y] \le mszám[x]$ és $bszám[y] = 0$
$mszám[y] = mszám[x] \Rightarrow$ *(x,y)* hurokél, ami triviálisan visszaél

$mszám[y] < mszám[x]$ és $bszám[y] = 0 \Rightarrow$ Az *y* bejárása elkezdődött, de még nem fejeződött be. Elkezdtük az *x* bejárását, amelynek során vizsgáljuk az *(x,y)* élt. Tehát az *y* bejárása során jutunk el az *x*-hez, azaz *x* leszármazottja *y*-nak.

(Az *MB(y)* eljárásban, vagy annak valamely rekurzív leszármazottjában hívtuk meg az *MB(x)*-et, tehát az *MB(x)*-nek előbb kell befejeződnie, mint az *MB(y)*-nak, így a 8.1.4. állítás alkalmazható). Az alábbi ábrán a (3,6) él visszaél.

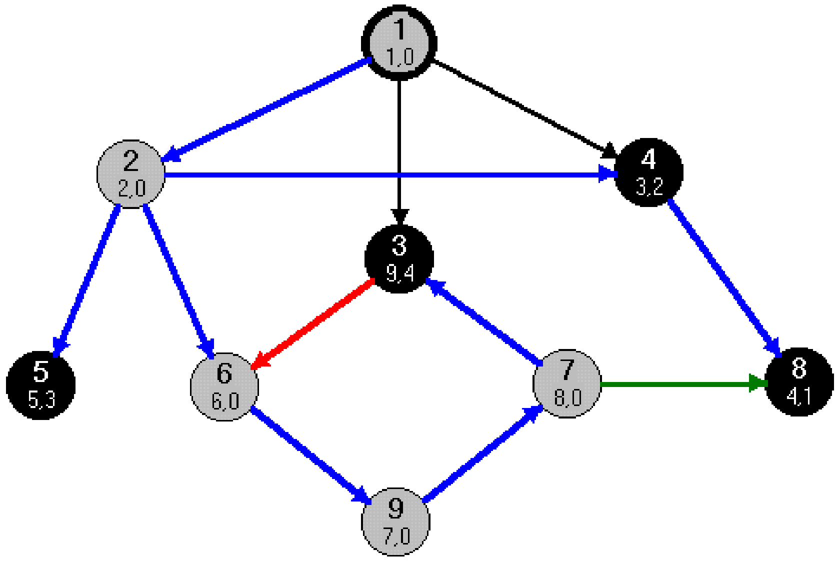


d) $mszám[y] < mszám[x]$ és $bszám[y] > 0$
$mszám[y] < mszám[x] \Rightarrow$ *y* nem leszármazottja *x*-nek, mivel a 8.1.3. állítás szerinti szükséges feltételt nem teljesíti.

$mszám[y] < mszám[x]$ szükséges feltétele, hogy *x* leszármazottja legyen *y*-nak, de az *y* bejárása befejeződött (*bszám*[y]>*0*), míg az *x* bejárása még nem, tehát a *bszám*[x] csak nagyobb lehet *bszám*[y]-nál, azaz az elégséges feltétel nem teljesül.

Tehát *x* és *y* nem leszármazottai egymásnak *T*-ben, amiből következik, hogy *(x,y)* él keresztél.

Az alábbi példán a (7,8) él keresztél.



Megjegyzés: Hogyan lehetne azonosítani az *(x,y)* él típusát az él vizsgálata során az *y* csúcs színének segítségével (lásd gyakorlaton)?

A mélységi bejárás műveletigénye: Mivel minden csúcsra pontosan egyszer hívjuk meg az *MB* eljárást, az eljárásban minden csúcsnak a szomszédait vizsgáljuk, összesen annyiszor ahány éle van a gráfnak, így *T(n)=O(n+e)*.