Általános alapelvek.

A pontozási útmutató célja, hogy a javítók a dolgozatokat egységesen értékeljék. Az útmutatónak ebből kifolyólag nem célja a feladatok teljes értékű megoldásának részletes leírása.

Az útmutatóban feltüntetett részpontszámok csak akkor járnak a megoldónak, ha a kapcsolódó gondolat egy áttekinthető, világosan leírt és megindokolt megoldás egy lépéseként szerepel a dolgozatban. A részpontszámok szükség esetén tovább is oszthatók.

Annak mérlegelése, hogy az útmutatóban feltüntetett pontszám a fentiek figyelembevételével a megoldónak (részben vagy egészében) jár-e, teljes mértékben a javító hatásköre. Részpontszám jár minden olyan ötletért, részmegoldásért, amely egy megoldásban érdemi szerephez juthat és amelyből a dolgozatban leírt gondolatmenet alkalmas kiegészítésével a feladat hibátlan megoldása volna kapható. Nem ér pontot azonban például az anyagban szereplő ismeretek, definíciók, tételek puszta leírása azok alkalmazása nélkül (még akkor sem, ha egyébként valamelyik leírt tény a megoldásban valóban szerephez jut).

Az útmutatóban leírttól eltérő jó megoldás természetesen maximális pontot ér, de bizonyítás nélkül csak az előadáson szereplő tételekre és állításokra lehet hivatkozni.

Ha egy megoldó egy feladatra több, egymástól lényegesen különböző megoldást is elkezd, akkor legfeljebb az egyikre adható pontszám. Ha mindegyik leírt megoldás vagy megoldásrészlet helyes vagy helyessé kiegészíthető, akkor a legtöbb részpontot érő megoldáskezdeményt értékeljük. Ha azonban több megoldási kísérlet között van helyes és (lényeges) hibát tartalmazó is, továbbá a dolgozatból nem derül ki, hogy a megoldó melyiket tartotta helyesnek, akkor a kevesebb pontot érő megoldáskezdeményt értékeljük (akkor is, ha ez a pontszám 0).

1. Egy rendezés során a bemeneti 3, 8, 2, 7, 4, 5, 6, 1 tömbből valahány lépés után a 2, 3, 7, 8, 4, 5, 6, 1 állapot állt elő. Megtörténhetett-e ez

   (a) az összefésüléses rendezésnél; (b) a gyorsrendezésnél?

   (Az említett algoritmusokat az előadáson tanult módon futtatjuk.)

Egy lehetséges helyes megoldás

(a) Igen: a tömb első felét már rendeztük (és ezután fogjuk még rendezni a tömb második felét, majd pedig összefésülni a két rendezett felet). *(5 pont)*

(b) Nem: bármelyik elem(ek) szerint is válogattuk szét a többit „kicsikre” és „nagyokra”, az 1-es nem maradhatott a tömb végén (akárhány fázison is vagyunk már túl). *(5 pont)*

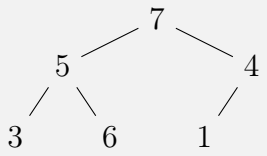
Pusztán annak megállapítására, hogy az (a) kérdésre igen, a (b) kérdésre pedig nem a válasz, nem jár pont. Ha egy megoldó nem tudja helyesen megindokolni az (a) és/vagy a (b) kérdésre a választ, de a megoldásból egyértelműen kiderül, hogy tudja, hogy hogyan működik az összefésüléses, illetve a gyorsrendezés, az $2 + 2$ pontot kapjon.

2. (a) Építsünk kupacot a tanult $O(n)$ lépésszámú eljárással a következő tömbből. (Egyértelműen derüljön ki, mikor mit csinál az algoritmus.)

$$7,\ 5,\ 4,\ 3,\ 6,\ 1$$

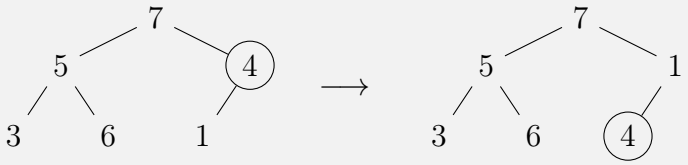
(b) Igaz-e, hogy az így kapott kupacon egy MINTÖR műveletet végrehajtva, majd a törölt elemet újra beszúrva visszakapjuk az eredeti kupacot?

(a) A tömb bináris fa alakja a következő.

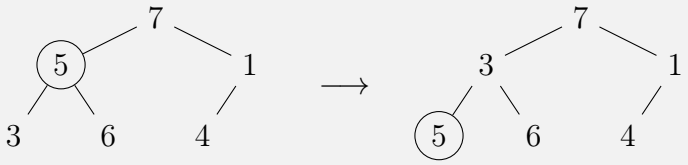


A KUPACOL(1), KUPACOL(6) és KUPACOL(3) műveletek hatására nem változik meg a fa.

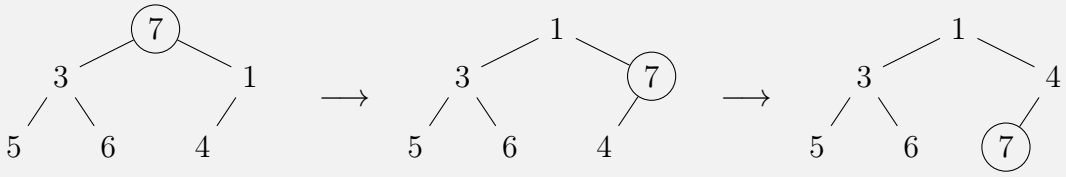
A KUPACOL(4) művelet hatása alább látható.



A KUPACOL(5) művelet hatása alább látható.

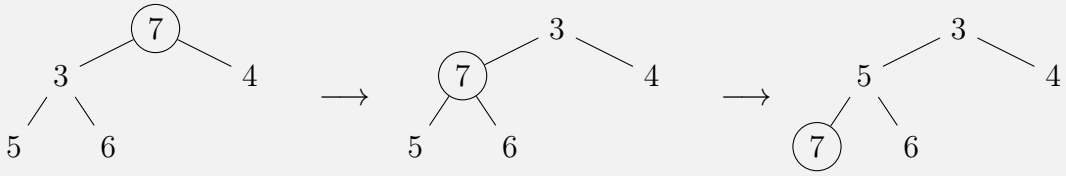


A KUPACOL(7) művelet hatása alább látható.



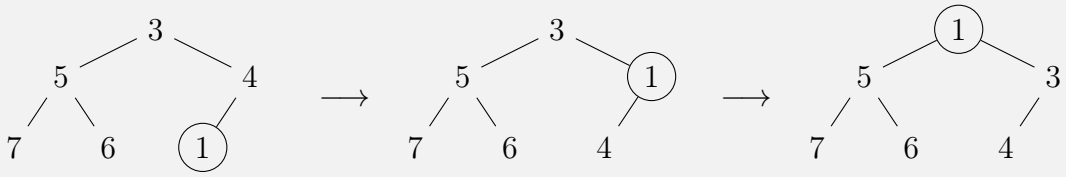
(6 pont)

(b) A MINTÖR művelet a következőképpen zajlik.



(2 pont)

A BESZÚR(1) művelet a következőképpen zajlik.



(2 pont)

Vagyis nem kaptuk vissza az eredeti kupacot. *(0 pont)*

Ha egy megoldó ismételt beszúrásokkal épít kupacot, az az első 6 pontot nem kaphatja meg.

Ha kiderül, hogy egy megoldó tudja, hogy a kupacépítés során alulról felfelé, szintenként jobbról balra hívjuk meg az elemekre a KUPACOL eljárást, de azt helytelenül hajtja végre, az az a) feladatra járó 6 pontból 1+1 pontot megkaphat; ha az is kiderül, hogy a KUPACOL eljárást is tudja, azért szintén 1 pontot kaphat. Azért ne vonjunk le pontot, ha egy megoldó nem hívja meg a levelekre (azaz most az 1-es, 6-os, 3-as csúcsokra) a KUPACOL eljárást.

Ha egy megoldó nem az órán tanult algoritmussal épít kupacot, hanem egy másképp épített kupaccal oldja meg a b) feladatot, akkor a b) feladatra járó 4 pontot megkaphatja. Amennyiben viszont egy megoldó nem egy kupaccal dolgozik a b) feladatban, akkor ezt a 4 pontot nem kaphatja meg abban az esetben sem, ha a MINTÖR és a BESZÚR műveletek működését ismeri.

Ha kiderül, hogy egy megoldó tudja, hogy a MINTÖR művelet során a legjobboldalibb levél (azaz most a 7-es) kerül fel a gyökérbe, azért 1 pontot kaphat; de bármit is rakott fel a gyökérbe, akkor ha azután helyesen állítja helyre a kupactulajdonságot, akkor azért szintén 1 pontot kaphat.

Ha kiderül, hogy egy megoldó tudja, hogy a BESZÚR művelet során hova kerül az új elem (azaz most az 1-es), azért 1 pontot kaphat; ha helyesen állítja helyre a kupactulajdonságot, akkor azért szintén 1 pontot kaphat.

3. Egy 2–3 fának 4 szintje van (azaz a gyökér és a levelek szintjén kívül van még két közbülső szint). Legkevesebb hány elem lehet a fában, ha tudjuk, hogy bármely új elem beszúrásakor legalább egy csúcsvágásra van szükség, és van olyan elem, aminek a beszúrásakor két csúcsvágás is kell?

Egy lehetséges helyes megoldás

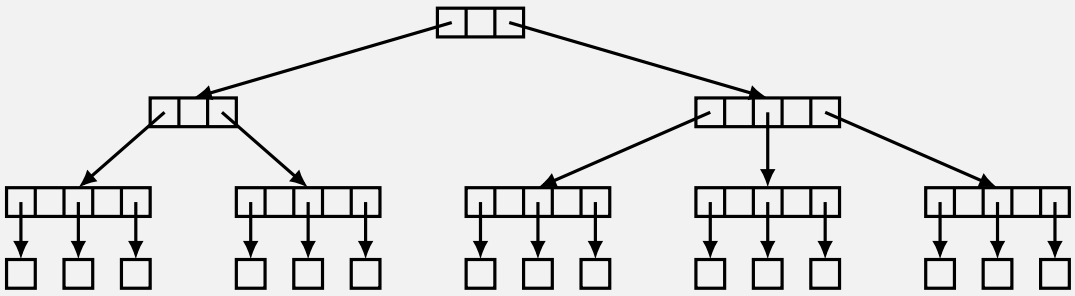
Ahhoz, hogy bármilyen elem beszúrásakor legyen csúcsvágás, közvetlenül a levelek feletti szinten minden csúcsnak három gyereke kell hogy legyen. (Hiszen ha itt valamelyik csúcsnak csak kettő gyereke lenne, akkor oda még csúcsvágás nélkül beszúrhatnánk egy új elemet harmadik gyerekként.) *(4 pont)*

Mivel egy elem beszúrásakor a szétvágandó háromgyerekes csúcsok közvetlenül egymás felett állnak, ezért kell hogy legyen a levelek feletti második szinten is egy háromgyerekes csúcs. *(4 pont)*

Ahhoz, hogy a lehető legkevesebb elemet tárolja a fa, az összes többi csúcsnak kétgyerekesnek kell lennie, *(1 pont)*

vagyis a gyökérnek két gyereke van, ezek közül az egyiknek kettő, a másiknak három gyereke, és mind a $2 + 3 = 5$ „unokának” három-három gyereke van (például az alábbi ábrán látható elrendezésben). *(0 pont)*

Tehát legkevesebb $5 \cdot 3 = 15$ elemet tárolunk egy ilyen fában. *(1 pont)*



Pusztán annak megállapítására, hogy a válasz 15, nem jár pont. Ha egy megoldó nem tudja helyesen megindokolni a választ, de a megoldásból egyértelműen kiderül, hogy tudja, hogy hogyan működik a 2–3 fába való beszúrás, az 2 pontot kapjon. Ha egy megoldó felrajzol egy megfelelő fát és elmagyarázza, hogy ez miért jó, de azt nem, hogy miért minimális, azért legfeljebb 1+2 pontot kaphat.

Ha egy megoldó leírja, hogy a levelek feletti szinten kell lennie háromgyerekes csúcsnak, azért 1 pontot kaphat, ha azt is leírja, hogy minden itteni csúcsnak három gyereke kell hogy legyen, szintén 1 pontot kaphat, ha pedig ezeket az állításokat meg is indokolja, akkor azért 2 pontot kaphat.

Ha egy megoldó leírja, hogy valahol lennie kell további háromgyerekes csúcsnak, azért 1 pontot kaphat, ha azt leírja, hogy ennek a háromgyerekes csúcsnak muszáj a levelek feletti második szinten lennie, azért is 1 pontot kaphat, ha azt is leírja, hogy minden más csúcs kétgyerekes, azért szintén 1 pontot, és mindezek indoklásáért 2 pontot kaphat.

4. Mutassuk meg, hogy az alábbi eldöntési probléma NP-beli.
   Bemenet: egy egyszerű, irányítatlan G gráf szomszédossági mátrixa.
   Kérdés: igaz-e, hogy G bármely csúcsát elhagyva a kapott gráfban lesz teljes párosítás?

Megfelelő tanú megadása:	*4 pont*
A bemenet méretének megállapítása:	*1 pont*
A tanú méretének megállapítása:	*2 pont*
A tanú helyességének ellenőrzése lépésszámelemzéssel:	*3 pont*

Ha egy megoldó külön kitér arra is, hogy a megadott tanú pontosan azon bemenetek esetén létezik, amire a válasz igenlő, az 1 extra pontot kaphat, amennyiben ezzel a feladatért járó összpontszám nem haladja meg a 10-et. Ha kiderül, hogy egy megoldó ismeri az NP fogalmát, de helytelen tanút ad meg az igenlő válaszra, akkor a bemenet méretének megállapításáért járó 1 ponton kívül legfeljebb további 2 pontot kaphat.

EGY LEHETSÉGES HELYES MEGOLDÁS

Az igen válaszra megfelelő tanú n darab párosítás (ahol n a gráf csúcsainak számát jelöli) úgy, hogy minden $i \in \{1, \ldots, n\}$ esetén az i-edik párosítás egy teljes párosítás abban a gráfban, amit G-ből az i-edik csúcs elhagyásával kapunk. Nyilván pontosan azon bemenetek esetén létezik n darab ilyen párosítás, amikre a válasz igenlő.

A bemenet mérete $\Theta(n^2)$ (avagy pontosabban n^2).

A tanú leírható n darab $n \times n$-es mátrix segítségével (aminek a sorai és oszlopai is a bemeneti gráf csúcsaival vannak indexelve), ahol a mátrix elemei 1-ek vagy 0-k aszerint, hogy az adott él benne van-e a párosításban. Ennek a mérete $n \cdot O(n^2) = O(n^3)$, azaz a bemenet méretében nézve polinomiális.

A tanú helyességéhez azt kell ellenőriznünk, hogy (1) valóban párosításokat adtunk meg, (2) a megadott párosítások valóban élei a bemeneti gráfnak, (3) minden $i \in \{1, \ldots, n\}$ esetén az i-edik párosítás az i-edik csúcs kivételével minden más csúcsot fed.

(1) A tanúban szereplő mindegyik mátrixban megnézzük, hogy minden sorban és minden oszlopban legfeljebb egy darab 1-es szerepel-e (és a főátló minden eleme 0); ez $n \cdot O(n^2) = O(n^3)$ lépés.

(2) A tanúban szereplő mindegyik mátrixban megnézzük, hogy ahol 1-es áll, ott a bemeneti mátrixban is 1-es szerepel-e; ez $n \cdot O(n^2) = O(n^3)$ lépés.

(3) A tanúban szereplő minden $i \in \{1, \ldots, n\}$ esetén megnézzük, hogy az i-edik mátrixnak az i-edik sorában és oszlopában nincsen 1-es, de minden más sorában és oszlopában van; ez $n \cdot O(n^2) = O(n^3)$ lépés.

Vagyis az ellenőrzés teljes lépésszáma $O(n^3)+O(n^3)+O(n^3) = O(n^3)$, ami a bemenet méretében nézve polinomiális.

Természetesen a tanú megadható más alakban is, például: a párosítások leírhatók csúcspárok segítségével is, illetve nem kell ragaszkodni ahhoz, hogy párosítások „sorrendbe legyenek rendezve". Ugyan egy csúcs neve csak $O(\log n)$ biten írható le, ne vonjunk le pontot azért, ha egy megoldó ezt $O(1)$-nek veszi.

Ha egy megoldó egyetlen csúcs elhagyása esetére oldja meg a feladatot, akkor összesen legfeljebb 4 pontot kaphat.

Ha egy megoldó azt mondja, hogy a probléma P-beli, és így $\mathrm{P} \subseteq \mathrm{NP}$ miatt a probléma NP-beli is, akkor azért legfeljebb 4 pontot kaphat (mert az az állítás nem hangzott el, hogy tetszőleges gráfban is tudunk polinomidőben teljes párosítást keresni).

5. Tegyük fel, hogy P $\neq$ NP.
   (a) Létezik-e olyan X eldöntési probléma, amire $X \in$ P és RH $\prec X$ teljesül?
   (b) Létezik-e olyan Y eldöntési probléma, amire $Y \in$ P és $Y \prec$ RH teljesül?

   (Az RH a részhalmaz-összeg eldöntési változatát jelöli.)

EGY LEHETSÉGES HELYES MEGOLDÁS

(a) Legyen $X \in P$ egy tetszőleges probléma. Megmutatjuk, hogy nem létezik RH $\prec X$ Karp-redukció. Indirekt tegyük fel, hogy létezik. *(0 pont)*
Ekkor $X \in$ P és a Karp-redukció tulajdonságai miatt RH is P-beli. *(2 pont)*
Az előadáson tanultak szerint az RH probléma NP-nehéz, *(1 pont)*
vagyis az előadáson tanultak szerint P = NP, *(2 pont)*
ami ellentmondás. Tehát RH $\not\prec X$. *(0 pont)*

(b) **I. megoldás.**
Legyen $Y \in P$ egy tetszőleges probléma. Megmutatjuk, hogy létezik $Y \prec$ RH. *(0 pont)*
Az előadáson tanultak szerint P $\subseteq$ NP, *(1 pont)*
vagyis az Y probléma NP-beli. *(2 pont)*
A korábbi megállapításunk szerint az RH probléma NP-nehéz, *(0 pont)*
így az NP-nehézség definíciója szerint létezik $Y \prec$ RH Karp-redukció. *(2 pont)*

Bár a megoldásból az derült ki, hogy az RH problémára bármely P-beli probléma Karp-redukálható, ezt elég lett volna egyetlen P-beli probléma esetén belátni.

(b) **II. megoldás.**

Legyen Y egy tetszőleges P-beli probléma. *(1 pont)*

Az alábbiakban megadunk egy $Y \prec$ RH Karp-redukciót. *(0 pont)*

Mivel $Y \in P$, ezért tetszőleges bemenetről polinomidőben el tudjuk dönteni, hogy a válasz igenlő vagy nemleges. Ha a válasz igenlő, akkor ehhez a bemenethez rendeljük hozzá az RH probléma azon $(s'_1, \ldots, s'_m; b')$ bemenetét, ahol $m = 1$ és $s'_1 = 1$ és $b' = 1$. Ha viszont a válasz nemleges, akkor rendeljük hozzá az RH probléma azon $(s'_1, \ldots, s'_m; b')$ bemenetét, ahol $m = 1$ és $s'_1 = 1$ és $b' = 2$. *(2 pont)*

Ekkor az Y tetszőleges bemenetéhez az RH probléma megfelelő bemenete polinomidőben elkészíthető: az Y probléma bemenetéről polinomidőben el tudjuk dönteni, hogy a válasz igenlő vagy nemleges, és ezután az RH probléma megfelelő bemenetét konstans lépésben fel tudjuk írni, ami összesen polinomidő. *(1 pont)*

Nyilván az Y probléma egy bemenetére pontosan akkor lesz igenlő a válasz, ha az RH probléma megfelelő bemenetre is igenlő a válasz. (Hiszen az $s'_1, \ldots, s'_m$ számok közül – ami jelen esetben csak az s'_1 – pontosan akkor lehet néhányat kiválasztani úgy, hogy a kiválasztott számok összege $b' = 1$ legyen, ha $s'_1 = 1$, azaz az Y probléma bemenetére a válasz igenlő.) *(1 pont)*

Tehát valóban létezik $Y \prec$ RH Karp-redukció. *(0 pont)*

III. megoldás.

Legyen Y az a probléma, amiben egy adott a egész számról kell eldöntenünk, hogy egyenlő-e eggyel. Ekkor a bemenet mérete $\Theta(\log a)$, és a bemenetről nyilván $O(1)$ lépésben el tudjuk dönteni, hogy eggyel egyenlő-e, tehát az Y probléma P-beli. *(1 pont)*

Az alábbiakban megadunk egy $Y \prec$ RH Karp-redukciót. *(0 pont)*

Vegyük az Y probléma egy tetszőleges a bemenetét, és ehhez rendeljük hozzá az RH probléma azon $(s'_1, \ldots, s'_m; b')$ bemenetét, ahol $m = 1$ és $s'_1 = a$ és $b' = 1$. *(2 pont)*

Ekkor az $(s'_1, \ldots, s'_m; b')$ elkészítésének lépésszáma $O(\log a)$, ami az Y probléma bemenetének méretében nézve polinomiális. *(1 pont)*

Nyilván az Y probléma a bemenetére pontosan akkor lesz igenlő a válasz, ha a az RH probléma megfelelő bemenetre is igenlő a válasz. (Hiszen az $s'_1, \ldots, s'_m$ számok közül – ami jelen esetben egyedül csak az a – pontosan akkor lehet néhányat kiválasztani úgy, hogy a kiválasztott számok összege $b' = 1$ legyen, ha $a = 1$, azaz az Y probléma bemenetére a válasz igenlő.) *(1 pont)*

Tehát valóban létezik $Y \prec$ RH Karp-redukció. *(0 pont)*

6. Adjunk meg egy 3SZÍN $\prec X$ Karp-redukciót, ahol X az alábbi eldöntési probléma.

   Bemenet: egy egyszerű, irányítatlan G gráf szomszédossági mátrixa.

   Kérdés: el lehet-e hagyni két csúcsot G-ből úgy, hogy a kapott gráf csúcsai helyesen kiszínezhetők legyenek 3 színnel?

Helyes a Karp-redukció iránya és értelmezési tartománya:	*1+1 pont*
Helyes konstrukció:	*4 pont*
A konstrukció elkészítésének lépésszáma:	*1 pont*
A konstrukció helyességének belátása:	*3 pont*

EGY LEHETSÉGES HELYES MEGOLDÁS

Legyen a G gráf a 3SZÍN probléma egy tetszőleges bemenete, és rendeljük hozzá ehhez az X problémának a következő G' bemenetét. Legyen G' az a gráf, amit úgy kapunk, hogy a G gráf mellé felveszünk külön komponensként egy K_5-öt.

A G gráf szomszédossági mátrixának mérete $\Theta(n^2)$ (avagy pontosabban n^2), ahol n a G gráf csúcsainak száma. A G' gráf szomszédossági mátrixa a következőképpen kapható meg: legyen egy $(n+5)\times(n+5)$-ös mátrix bal felső $n\times n$-es részmátrixa G szomszédossági mátrixa, a jobb alsó 5×5-ös részmátrixa pedig nézzen ki úgy, aminek a főátlójában minden elem 0, a maradék 20 eleme pedig 1-es, a mátrix összes többi eleme pedig 0. Nyilván G' szomszédossági mátrixának elkészítésének lépésszáma $O(n^2)$, ami polinomiális.

Belátjuk, hogy a 3SZÍN problémában a G gráfra pontosan akkor lesz igenlő a válasz, ha az X problémában a G' gráfra is igenlő a válasz.

Először tegyük fel, hogy a 3SZÍN problémában a G gráfra igenlő a válasz, azaz a G gráf csúcsai helyesen kiszínezhetők 3 színnel. Hagyjunk el a G' gráfból két új csúcsot (az öt közül bármelyik kettőt): ekkor a G gráf mellett marad még egy háromszög. A feltevés miatt a G gráf csúcsai helyesen kiszínezhetők 3 színnel, a háromszög csúcsai szintén (a három csúcsot három különböző színre színezve), ezért az X problémában a G' gráfra is igenlő a válasz.

Fordítva, tegyük fel, hogy az X problémában a G' gráfra igenlő a válasz, azaz a G' gráfból el lehet hagyni két csúcsot úgy, hogy a kapott gráf csúcsai helyesen kiszínezhetők legyenek 3 színnel. Ekkor muszáj az öt új csúcs közül kettőt elhagyni (hiszen sem a K_5, sem a K_4 csúcsai nem színezhetők helyesen három színnel). Így mivel a megmaradt gráfnak részgráfja a G gráf, ezért ekkor G csúcsai is helyesen kiszínezhetők 3 színnel, vagyis a 3SZÍN problémában a G gráfra is igenlő a válasz.

Természetesen más helyes konstrukciót is meg lehet adni, például azt, amikor a G gráfhoz hozzáveszünk két új csúcsot, amiket aztán egymással és G minden csúcsával is összekötünk.

7. Egy könyvtárban szeretnénk megfigyelni, hogy melyik könyvet milyen gyakran kölcsönzik ki. Tervezzünk olyan adatstruktúrát, amiben a következő műveleteket lehet elvégezni.
   - ÚJ(x): felvesz az adatbázisba egy új (eddig még ki nem kölcsönzött) könyvet, aminek a kódja x lesz;
   - ÚJ_KÖLCSÖNZÉS(x): rögzíti az x kódú könyv egy újabb kikölcsönzését;
   - HÁNY_KÖLCSÖNZÉS(x): megadja, hogy eddig hány alkalommal kölcsönözték ki az x kódú könyvet;
   - DARAB(k): megadja, hogy hány olyan könyv van a könyvtárban, amit eddig pontosan k alkalommal kölcsönöztek ki.

   Mindegyik művelet lépésszáma $O(\log_2 n)$ legyen, ahol n a könyvtárban lévő könyvek számát jelöli.

Tároljuk a könyvek kódját egy piros-fekete (vagy egy 2–3) fában. Egy másik piros-fekete (vagy 2–3) fában tároljuk el a különböző kikölcsönzési számokat úgy, hogy a fa csúcsaiba egy extra számlálót is rakunk, ami azt mutatja, hogy hány könyvet kölcsönöztek ki ennyiszer. A két piros-fekete fa csúcsait mutatókkal kössük össze aszerint, hogy azok mutassák, hogy az adott kódú könyvet hány alkalommal kölcsönözték ki. *(3 pont)*

Az ÚJ(x) művelet során először az első fába a piros-fekete fa BESZÚR műveletével beszúrjuk az x elemet. Majd a második fában pedig a piros-fekete fa KERES műveletével megnézzük, hogy szerepel-e már a 0 elem: ha igen, akkor a számlálóját megnöveljük eggyel, ha nem, akkor pedig a piros-fekete fa BESZÚR műveletével beszúrjuk a 0-t, és az új csúcs számlálóját 1-re állítjuk. Végül felveszünk egy új mutatót az első fa x-et tartalmazó csúcsából a második fa 0-t tartalmazó csúcsára. *(1 pont)*

Az ÚJ_KÖLCSÖNZÉS(x) művelet során először a piros-fekete fa KERES műveletével megkeressük az első fában az x elemet, majd a mutatójával átlépünk a második fába, hogy megtudjuk, hogy hányszor kölcsönözték ki ezt a könyvet; tegyük fel, hogy k alkalommal. Ha a k-t tartalmazó csúcsnak a számlálója 1-nél nagyobb, akkor csökkentsük ezt a számlálót eggyel, különben meg a piros-fekete fa TÖRÖL műveletével töröljük a k-t. Ezután a piros-fekete fa KERES műveletével keressük a második fában a $k+1$ elemet: ha benne van, akkor a számlálóját növeljük meg eggyel, ha viszont nem, akkor pedig a piros-fekete fa BESZÚR műveletével beszúrjuk a $k+1$ elemet, és az új csúcs számlálóját 1-re állítjuk. Végül az első fa x-et tartalmazó csúcsának a mutatóját átállítjuk, hogy a második fa $(k+1)$-et tartalmazó csúcsára mutasson. *(2 pont)*

A HÁNY_KÖLCSÖNZÉS(x) művelet során először a piros-fekete fa KERES műveletével megkeressük az első fában az x elemet, majd a mutatójával átlépünk a második fába, hogy megtudjuk, hogy hányszor kölcsönözték ki ezt a könyvet. *(1 pont)*

A DARAB(k) művelet során a piros-fekete fa KERES műveletével megkeressük a második fában a k elemet, és megnézzük a számlálójának az értékét; ha pedig k nincs benne a második fában, akkor 0 értékkel térünk vissza. *(1 pont)*

Mivel az első piros-fekete fában n darab elemet, a másodikban pedig legfeljebb n darab elemet tárolunk, ezért mindkét piros-fekete fában a KERES, BESZÚR és TÖRÖL műveletek lépésszáma a tanultak szerint $O(\log_2 n)$. *(1 pont)*

Az adatstruktúránk mindegyik műveletének során a piros-fekete fa KERES, BESZÚR és TÖRÖL műveleteit használjuk $O(1)$-szer, valamint még $O(1)$ lépést végzünk a mutatókkal. Így az adatstruktúránk mindegyik műveletének a lépésszáma $O(1) \cdot O(\log_2 n) + O(1) = O(\log_2 n)$. *(1 pont)*

Ha egy megoldó a könyvek kódját tárolja el egy kiegyensúlyozott fában, amibe valahogyan a kölcsönzések számát is beleírja, de a DARAB műveletet nem tudja $O(\log_2 n)$ lépésben megvalósítani, az legfeljebb 3 pontot kaphat.