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. Adott egy $2k+1$ darab egész számot tartalmazó rendezett T tömb. Adjunk $O(\log_2 k)$ lépésszámú algoritmust annak eldöntésére, hogy van-e olyan szám, ami legalább $(k+1)$-szer szerepel T-ben.

EGY LEHETSÉGES HELYES MEGOLDÁS

Ha van olyan a elem, ami a $(2k+1)$-elemű rendezett tömbben legalább $(k+1)$-szer szerepel, akkor biztosan ez az elem áll a középső (azaz a $(k+1)$-edik) cellában. *(3 pont)*
Keressük bináris kereséssel az a elem első, illetve utolsó előfordulását a tömbben. *(2 pont)*
Az első előfordulás megkereséséhez a $T[0 : k-1]$ résztömbben nézzük meg a középső elemre (ha két középső is van, akkor a kettő közül az egyikre, mondjuk a baloldalibbra), hogy hogyan viszonyul a-hoz:

- (a tömb rendezettsége miatt az nem lehet, hogy a középső elem nagyobb, mint a),
- ha a középső elem kisebb, mint a, akkor folytassuk az eljárást a tömb második felében,
- ha a középső elem egyenlő a-val, akkor egyrészt, tároljuk el a középső elem indexét amennyiben az kisebb, mint a legutóbb eltárolt index (a kezdetben eltárolt index legyen $*$), másrészt pedig folytassuk az eljárást a tömb első felében;

ha már nincs több vizsgálandó elem, akkor az utoljára eltárolt index lesz az a elem első előfordulásának a helye. Az utolsó előfordulás megkereséséhez értelemszerűen kell módosítani ezen az eljáráson: itt a $T[k+1 : 2k]$ résztömbbel dolgozunk, aminek ha a középső eleme nagyobb, mint a, akkor a tömb első felében folytatjuk az eljárást, egyenlőség esetén pedig a nagyobb indexeket jegyezzük fel, és a tömb második felében folytatjuk az eljárást. *(1 pont)*
Ha az utolsó és az első előfordulás indexének különbsége legalább k, akkor van olyan elem (nevezetesen a), ami legalább $(k+1)$-szer fordul elő a tömbben, különben viszont nincs. *(2 pont)*
Az eljárás teljes lépésszáma a bináris keresés lépésszáma miatt $2 \cdot O(\log_2 k) = O(\log_2 k)$. *(2 pont)*

Ha egy megoldó érdemi kísérletet tesz annak elmagyarázására, hogy a bináris keresést ebben az esetben hogyan alkalmazzuk, akkor az erre járó 1 pontot kapja meg.

2. (a) Építsünk beszúrásokkal bináris keresőfát az alábbi sorrendben érkező számokból:

7, 5, 8, 4, 2.

(Elég a végső bináris keresőfát felrajzolni.)

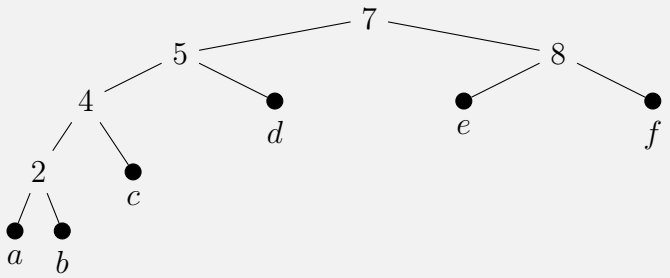
(b) Ki lehet-e színezni ennek a bináris keresőfának a csúcsait pirosra és feketére úgy, hogy egy piros-fekete fát kapjunk?

(a) A végső bináris keresőfa a következő.

(4 pont)

(b) EGY LEHETSÉGES HELYES MEGOLDÁS

Legyen az üres levelek neve a következő.



A piros-fekete fa definíciója szerint a gyökér és a levelek feketék.

Ekkor az 5-öt tartalmazó csúcsból a d-be vezető úton az 5-ös csúcsot nem számítva pontosan 1 fekete csúcs van (a d csúcs).

Vagyis a piros-fekete fa definíciója szerint az 5-ös csúcsból a levelekbe lefelé vezető összes úton az 5-ös csúcsot nem számítva pontosan 1 fekete csúcs kell hogy legyen.

Emiatt ha az 5-ös csúcsból az a-ba lefelé vezető utat nézzük, akkor azon már az a csúcs fekete, vagyis a 2-es és a 4-es csúcsoknak pirosnak kell lenniük.

Mivel azonban a piros-fekete fa definíciója szerint a piros csúcsok gyerekei feketék, ezért nem lehet kiszínezni ennek a bináris keresőfának a csúcsait pirosra és feketére úgy, hogy egy piros-fekete fát kapjunk. *(6 pont)*

Pusztán annak megállapítására, hogy a válasz nem, vagy a piros-fekete fa definíciójának alkalmazás nélküli kimondására nem jár pont. Ha egy megoldó nem tudja helyesen megindokolni a (b) kérdésre a választ, de a megoldásból egyértelműen kiderül, hogy értelmesen használni akarja a piros-fekete fa tulajdonságait, az 2 pontot kapjon.

3. Tervezzünk olyan adatstruktúrát, amiben különböző pozitív egész számokat tárolhatunk és amiben a következő műveleteket lehet elvégezni.
   - LEGKISEBB_PÁROS: visszaadja a tárolt számok közül a legkisebb párosat;
   - LEGNAGYOBB_PÁRATLAN: visszaadja a tárolt számok közül a legnagyobb páratlant;
   - KERES(x): megadja, hogy az x számot tároljuk-e az adatstruktúrában;
   - NÖVEL(x): megnöveli az x elem értékét 1-gyel, ha az $x+1$ szám nincs benne az adatstruktúrában, egyébként nem csinál semmit.

   Mindegyik művelet lépésszáma $O(\log_2 n)$ legyen, ahol n a tárolt elemek számát jelöli.

EGY LEHETSÉGES HELYES MEGOLDÁS

Tároljuk a páros, és a páratlan számokat is egy-egy piros-fekete fában (vagy 2–3 fában). *(3 pont)*

A KERES(x) művelet során az x paritásának megfelelő fában a piros-fekete fa KERES műveletével keressük x-et, ami a tanultak szerint $O(\log_2 n)$ lépés. (De azt is megtehetjük, hogy mindkét fában keressük x-et, mert az $2 \cdot O(\log_2 n) = O(\log_2 n)$ lépés.) *(2 pont)*

A LEGKISEBB_PÁROS művelethez meg kell keresnünk a páros számokat tároló fában a legkisebb elemet, ami a tanultak szerint $O(\log_2 n)$ lépés, a LEGNAGYOBB_PÁRATLAN művelethez pedig a páratlan számokat tároló fában a legnagyobb elemet, ami szintén $O(\log_2 n)$ lépés. *(2 pont)*

A NÖVEL(x) művelethez megkeressük az x és az $x+1$ elemeket az adatstruktúrában, ami az előzőek szerint $O(\log_2 n)$ lépés. Ha az x benne van és $x+1$ nincsen, akkor a piros-fekete fa TÖRÖL műveletével töröljük a megfelelő piros-fekete fából az x elemet és beszúrjuk a másik fába az $x+1$ elemet. Minden más esetben nem csinálunk semmit. Ez a tanultak szerint $4 \cdot O(\log_2 n)$ lépés. *(3 pont)*

Ha egy megoldó kiegyensúlyozott fa helyett bináris keresőfát használ, akkor legfeljebb 6 pontot kaphat. Ha egy megoldó a NÖVEL(x) művelet során elfelejti törölni az x elemet az adatstruktúrából, akkor ezért 1 pontot veszítsen el.

4. Egy 7-méretű hash-táblában kettős hash-elést alkalmazunk, amihez a $h(x) = x \pmod 7$ hash-függvényt és a $h'(x) = 1 + \big(x \pmod 6\big)$ másodlagos hash-függvényt használjuk. Néhány elem beszúrása után (miközben törlés nem történt) az alábbi táblát kaptuk.

0	1	2	3	4	5	6
0	8		10		5	6

(a) Ebben a táblában végrehajtjuk a TÖRÖL(6), majd a TÖRÖL(0) műveleteket. Rajzoljuk le a tábla állapotát a két művelet elvégzése után.

(b) Az így kapott táblában mely cellákat vizsgáljuk meg a KERES(3) művelet során?

(c) A kapott táblában hogyan zajlik a BESZÚR(3) művelet?

(a)

0	1	2	3	4	5	6
$*$	8		10		5	$*$

(2 pont)

(b) (Mivel $h(3) = 3$ és $h'(3) = 4$, ezért a 3-as indexű cellából indulva 4-esével ugrálunk balra ciklikusan értve, amíg nem találunk egy üres cellát.)

Vagyis a KERES(3) során a 3-as, a 6-os, és a 2-es indexű cellákat vizsgáljuk meg. *(4 pont)*

(c) (Mivel $h(3) = 3$ és $h'(3) = 4$, ezért a 3-as indexű cellából indulva 4-esével ugrálunk balra ciklikusan értve, amíg nem találunk vagy egy üres vagy egy $*$-ot tartalmazó cellát.)

0	1	2	3	4	5	6
$*$	8		10		5	3

(4 pont)

Ha egy megoldó nem jelöli meg a törölt cellákat, és a b) és c) feladatokban sem foglalkozik ezekkel, akkor összesen legfeljebb 3 pontot kaphat. Ha egy megoldó nem tudja, hogy a KERES művelet során hogyan viselkednek a $*$-gal jelölt mezők, akkor a b) feladatra járó 4 pontból legfeljebb 1-et kaphat meg. Ha egy megoldó nem tudja, hogy a BESZÚR művelet során hogyan viselkednek a $*$-gal jelölt mezők, akkor a c) feladatra járó 4 pontból legfeljebb 1-et kaphat meg. Ha egy megoldó a próbasorozatot nem balra, hanem jobbra indítja el, akkor ezért ne veszítsen pontot.

5. Mutassuk meg, hogy az alábbi eldöntési probléma coNP-beli.

   Bemenet: egy egyszerű, irányítatlan G gráf szomszédossági mátrixa, és egy $k \leq |V(G)|$ pozitív egész szám.

   Kérdés: igaz-e, hogy G-ből bármely k darab csúcsot törölve a kapott gráfnak legfeljebb k komponense lesz?

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

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

Egy lehetséges helyes megoldás

A nem válaszra megfelelő tanú k darab olyan csúcs, melynek elhagyásával legalább $k+1$ komponens keletkezik. Nyilván pontosan azon bemenetek esetén létezik k ilyen csúcs, amikre a válasz nemleges.

A bemenet mérete a $k \leq n$ feltétel miatt $\Theta(n^2 + \log k) = \Theta(n^2)$, ahol n a gráf csúcsainak számát jelöli.

A tanú leírható egy, a csúcsokkal indexelt tömb segítségével, ahol a tömb elemei 1-ek vagy 0-k aszerint, hogy az adott csúcsot el akarjuk-e hagyni. Ennek a mérete $O(n)$, azaz a bemenet méretében nézve polinomiális.

A tanú helyességéhez azt kell csak ellenőriznünk, hogy valóban k darab csúcsot hagytunk el és a megadott csúcsokat elhagyva valóban legalább $k+1$ komponens keletkezik.

Ehhez egyrészt végigmegyünk a tömbön és megszámoljuk, hogy k darab 1-es van-e benne; ez $O(n)$ lépés.

Másrészt, töröljük a megadott csúcsokat a G gráfból, azaz hagyjuk el ezen csúcsok sorait és oszlopait a G szomszédossági mátrixából. (Pontosabban másoljuk le egy új mátrixba a megmaradó csúcsok sorai és oszlopai által alkotott részmátrixot.) A kapott gráfon futtassunk szélességi (vagy mélységi) bejárást a komponensek számának meghatározására.

Az új gráf szomszédossági mátrixának elkészítése $O(n^2)$ lépés, a szélességi (vagy mélységi) bejárás futtatása pedig $O\big((n')^2\big)$ lépés, ahol n' az új gráf csúcsainak száma. Mivel $n' \leq n$, ezért az ellenőrzés teljes lépésszáma $O(n) + O(n^2) + O\big((n')^2\big) = O(n^2)$, ami a bemenet méretében nézve polinomiális.

Természetesen a tanú megadható más alakban is, például az elhagyandó csúcsok nevének felsorolásával. Ugyan egy csúcs neve csak $O(\log n)$ biten írható le, ne vonjunk le pontot azért, ha egy megoldó ezt $O(1)$-nek veszi.

6. Tegyük fel, hogy P $\neq$ NP. Létezik-e ekkor az alábbi X probléma esetén IP $\prec X$ Karp-redukció? (Az IP az egészértékű programozás eldöntési változatát jelöli.)

   Bemenet: egy egyszerű, irányítatlan G gráf szomszédossági mátrixa.

   Kérdés: igaz-e, hogy G bármely két csúcsa között vezet legfeljebb 5-élű út?

EGY LEHETSÉGES HELYES MEGOLDÁS

Belátjuk, hogy az X probléma P-beli. *(0 pont)*

Futtassunk G minden csúcsából szélességi bejárást; ezzel megkapjuk minden csúcsból az összes többibe vezető legkevesebb élű utaknak a hosszát. Ha ez mindig legfeljebb 5, akkor a válasz igen, különben pedig nem. *(3 pont)*

A bemenet mérete $\Theta(n^2)$ (avagy pontosabban n^2), a fenti algoritmus lépésszáma pedig $n \cdot O(n^2) = O(n^3)$, ami polinomiális. *(2 pont)*

Tehát az X probléma valóban P-beli. *(0 pont)*

Megmutatjuk, hogy nem létezik IP $\prec X$ Karp-redukció. Indirekt tegyük fel, hogy létezik. *(0 pont)*

Ekkor $X \in$ P és a Karp-redukció tulajdonságai miatt IP is P-beli. *(2 pont)*

Az előadáson tanultak szerint az IP probléma NP-nehéz, *(1 pont)*

vagyis az előadáson tanultak szerint P = NP. *(2 pont)*

Pusztán annak megállapításáért, hogy $X \in$ P, nem jár pont. Ha azonban egy megoldó megpróbálja ezt belátni, de helytelen algoritmust ad rá, úgy az ezért járó $3 + 2$ pontból legfeljebb 2-et megkaphat.

7. Mutassuk meg, hogy az alábbi eldöntési probléma NP-nehéz.

   Bemenet: $s_1, \ldots, s_m, b$ nemnegatív egész számok.

   Kérdés: el lehet-e látni előjellel az $s_1, \ldots, s_m$ számokat úgy, hogy az előjeles összegük éppen b legyen?

Értelmes NP-nehéz feladat választása a Karp-redukcióhoz:	*1 pont*
Jó irányba történik a Karp-redukció:	*1 pont*
Ebből következtetés a feladatbeli probléma NP-nehézsége:	*1 pont*
Helyes konstrukció:	*3 pont*
A konstrukció elkészítésének lépésszáma:	*1 pont*
A konstrukció helyességének belátása:	*3 pont*

EGY LEHETSÉGES HELYES MEGOLDÁS

Megmutatjuk, hogy az ismerten NP-nehéz PARTÍCIÓ probléma polinomiálisan visszavezethető a feladatbeli X problémára. Ebből az előadáson tanult tétel szerint már következni fog, hogy az X probléma NP-nehéz.

Rendeljük hozzá a PARTÍCIÓ probléma tetszőleges $I = (s_1, \ldots, s_m)$ bemenetéhez az X probléma azon $I' = (s'_1, \ldots, s'_m, b')$ bemenetét, ahol $s'_1 := s_1$, $\ldots$, $s'_m := s_m$, $b' := 0$.

Ekkor ha I mérete N, akkor I' elkészítésének lépésszáma $O(N)$, ami polinomiális.

Belátjuk, hogy a PARTÍCIÓ problémában az I bemenetre pontosan akkor lesz igenlő a válasz, ha az X problémában az I' bemenetre is igenlő a válasz.

Először tegyük fel, hogy a PARTÍCIÓ problémában az I bemenetre igenlő a válasz, azaz az $s_1, \ldots, s_m$ számok két csoportba oszthatók úgy, hogy a két csoportban a számok összege egyenlő legyen. Minden $i \in \{1, \ldots, m\}$ esetén, ha s_i az első csoportba tartozik, akkor adjunk az X problémában az s'_i számnak pozitív előjelet, ha pedig s_i a második csoportba tartozik, akkor adjunk az s'_i számnak negatív előjelet. Ekkor az $s'_1, \ldots, s'_m$ számok előjeles összege éppen $b' = 0$ lesz, vagyis az X problémában az I' bemenetre is igenlő a válasz.

Fordítva, tegyük fel, hogy az X problémában az I' bemenetre igenlő a válasz, azaz az $s'_1, \ldots, s'_m$ számokat el lehet látni előjellel úgy, hogy az előjeles összeg éppen $b' = 0$ legyen. Minden $i \in \{1, \ldots, m\}$ esetén, ha s_i pozitív előjelet kapott, akkor a PARTÍCIÓ problémában rakjuk az s_i számot az első csoportba, ha pedig s'_i negatív előjelet kapott, akkor rakjuk az s_i számot a második csoortba. Ekkor az első és a második csoportbeli számok összege egyenlő lesz, vagyis a PARTÍCIÓ problémában az I bemenetre is igenlő a válasz.

Tehát az X probléma valóban NP-nehéz.