

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, rész megoldásért, 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 pusztá 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 dolgozathból nem derül ki, hogy a megoldó melyiket tartotta helyesnek, akkor a kevesebb pontot érő megoldáskezdeményt értékeljük (akkor is, ha ez a pontszám 0).

1. Bizonyítsuk be (alkalmas c konstans és n_0 küszöbindex megadásával) vagy cáfoljuk meg a következő állítást: az

$$f(n) = 16 \cdot n^3 \cdot \log_2 n + 8 \quad \text{és} \quad g(n) = 7 \cdot n^5 + \sqrt[3]{n}$$

függvényekre $f \in O(g)$ teljesül.

Megfelelő (azaz $c \cdot g(n)$) alakú felső becslésre történő érdemi törekvés: 2 pont

Helyes felső becslések az egyes lépésekben, azaz a $\log_2 n$ -nel és a 8-cal való elbánás, az n^5 -re váltás, és a $\sqrt[3]{n}$ behozása: 1+1+1+1 pont

Az egyes lépésekben használt felső becslésekhez a helyes küszöbindexek megállapítása: 2 pont

Megfelelő végső konstansszorzó és küszöbindex kiolvasása: 2 pont

Pusztán annak megállapítására, hogy $f \in O(g)$, vagy a nagy ordó definíciójának alkalmazás nélküli kimondására nem jár pont.

EGY LEHETSÉGES HELYES MEGOLDÁS

Mivel

$$16 \cdot n^3 \cdot \log_2 n + 8 \underset{\substack{\uparrow \\ \text{ha } n > 0, \\ \text{akkor } \log_2 n \leq n}}{\leq} 16n^3 \cdot n + 8 \underset{\substack{\uparrow \\ \text{ha } n \geq 1, \\ \text{akkor } n^4 \leq n^5 \\ \text{és } 1 \leq n^5}}{\leq} 16n^5 + 8n^5 = 24n^5 \underset{\substack{\uparrow \\ \text{ha } n \geq 0, \\ \text{akkor } 0 \leq \sqrt[3]{n}}}{\leq} \frac{24}{7} \cdot (7n^5 + \sqrt[3]{n}),$$

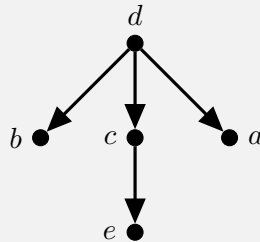
ezért a nagy ordó definíciójában például a $c = 24/7$ és az $n_0 = 1$ értékeket választva látjuk $f \in O(g)$ teljesül.

2. Legyen G egy 5-csúcű teljes (irányítatlan) gráf az a, b, c, d, e csúcsokon. Hányféleképpen lehet G minden élét valamelyik irányban úgy megirányítani, hogy a kapott G' irányított gráfnak legyen olyan mélységi bejárása, ami során a csúcsok az alábbi mélységi és befejezési számokat kapják?

$$a: (5, 4); \quad b: (2, 1); \quad c: (3, 3); \quad d: (1, 5); \quad e: (4, 2)$$

EGY LEHETSÉGES HELYES MEGOLDÁS

Ekkor a mélységi erdőnek a következőképpen kell kinéznie. (Ezt nem kell indokolni.)



(4 pont)

Vagyis a G gráf $\{a, d\}$, $\{b, d\}$ és $\{c, d\}$ és $\{c, e\}$ éleit csak a fenti ábrának megfelelően lehet irányítani.

(0 pont)

Mivel az előadáson tanultak szerint a mélységi bejárással kapott keresztélek mindig egy későbbi ágból mutatnak egy korábbiba, ezért a $\{d, e\}$ él kivételével minden más él csak egyféleképpen irányítható (a fenti ábrán jobbról balra).

(3 pont)

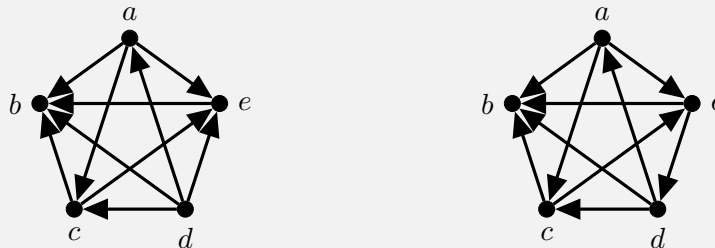
A $\{d, e\}$ élt viszont bármelyik irányba irányíthatjuk: ha d -ből e -be irányítjuk, akkor egy előreél lesz a mélységi bejárás során, az ellenkező irányba irányítva pedig egy visszaél.

(2 pont)

Vagyis a G gráfnak két megfelelő irányítása létezik;

(1 pont)

ezek alább láthatók.



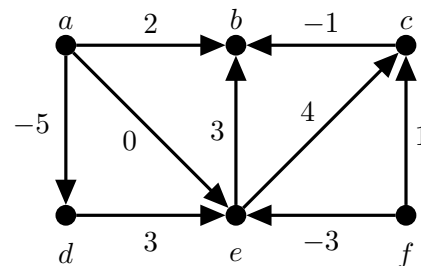
Ha egy megoldó mindenféle indoklás nélkül felrajzolja akár mindkét keresett irányítást, azért nem jár pont. Ha viszont egy megoldó a két keresett irányítás felrajzolása után a mélységi bejárás lefuttatásával megmutatja, hogy ezek az irányítások valóban lehetségesek, de arra egyáltalán nem tér ki, hogy más irányítás miért nincsen, az összesen legfeljebb 4 pontot kaphat.

Ha egy megoldó lépésenként végigveszi, hogy hogyan fut le a mélységi bejárás és ehhez G melyik élét milyen irányba kell irányítani, akkor 4 pont jár arra, ha kiderül, hogy érti, hogy hogyan megy a mélységi bejárás, 1 pont az ez alapján jól meghatározott két lehetőségért, és 5 pont az indoklásra.

(Példa egy ilyen típusú megoldás elkezdéséhez. Mivel a bejárás a d csúcsból indult és azt nem fejeztük be rögtön, ezért onnan átléptünk b -be, azaz a b és d közötti élt d -ből b -be kell irányítani. Mivel viszont b -t rögtön be is fejezzük, ezért a többi b -re illeszkedő élt is b -be kell irányítani. A b csúcs befejezése után visszalépünk d -be, és mivel őt még mindig nem fejezzük be, ezért onnan átlépünk c -be, azaz a c és d közötti élt c -be kell irányítani, stb.)

3. Tekintsük a jobbra látható élsúlyozott gráfot.

- (a) Írjuk fel a gráf egy topologikus sorrendjét. (Kivételesen nem kell elmagyarázni, hogy ez miért jó.)
- (b) Az órán tanult algoritmus segítségével határozzunk meg egy leghosszabb ab -utat és annak hosszát, és dokumentáljuk az algoritmus lépéseit. (Egyértelműen derüljön ki, hogy mikor mit csinál az algoritmus.)



- (a) (A gráfnak három topologikus sorrendje is van, de természetesen csak egyet kell ezek közül megadni.)

$$a, d, f, e, c, b \quad \text{vagy} \quad a, f, d, e, c, b \quad \text{vagy} \quad f, a, d, e, c, b \quad (2 \text{ pont})$$

- (b) (Mivel a gráf aciklikus, ezért alkalmazható az órán tanult algoritmus a leghosszabb utak meghatározására. A választott topologikus sorrend szerint haladva az alábbi számolásokat kell elvégezni.)

$$\text{maxút}(a) = 0 \quad \text{előző}(a) = *$$

$$\begin{aligned} \text{maxút}(d) &= \max \{ \text{maxút}(a) + w(ad) \} = \\ &= \max \{ 0 + (-5) \} = -5 \end{aligned} \quad \text{előző}(d) = a$$

$$\text{maxút}(f) = \max \emptyset = -\infty \quad \text{előző}(f) = *$$

$$\begin{aligned} \text{maxút}(e) &= \max \{ \text{maxút}(a) + w(ae), \text{maxút}(d) + w(de), \text{maxút}(f) + w(fe) \} = \\ &= \max \{ 0 + 0, (-5) + 3, (-\infty) + (-3) \} = 0 \end{aligned} \quad \text{előző}(e) = a$$

$$\begin{aligned} \text{maxút}(c) &= \max \{ \text{maxút}(e) + w(ec), \text{maxút}(f) + w(fc) \} = \\ &= \max \{ 0 + 4, (-\infty) + 1 \} = 4 \end{aligned} \quad \text{előző}(c) = e$$

$$\begin{aligned} \text{maxút}(b) &= \max \{ \text{maxút}(a) + w(ab), \text{maxút}(c) + w(cb), \text{maxút}(e) + w(eb) \} = \\ &= \max \{ 0 + 2, 4 + (-1), 0 + 3 \} = 3 \end{aligned} \quad \text{előző}(b) = c$$

(Az $\text{előző}(b)$ értéke e is lehetett volna.) (7 pont)

Egy leghosszabb ab -út (az előző értékekből visszaírva): $aecb$ (vagy aeb), (1 pont)

és ennek hossza 3. (0 pont)

Az algoritmus futtatásáért járó 7 pontból 5 pont jár az egyes csúcsokba vezető leghosszabb utak hosszainak kiszámolására, és 2 pont jár az előző értékek meghatározására. Ha semmilyen formában nem jelennek meg részszámítások, akkor a leghosszabb utak hosszainak kiszámítására járó 5 pontból legfeljebb 1 pont adható, amennyiben a megoldásból kiderül, hogy egy topologikus sorrend szerinti sorrendben kell meghatározni a leghosszabb utak hosszát. Ha egy jó rajz alapján pontosan kiderül, hogy milyen számolásokat végzett egy megoldó, akkor az is maximális pontot ér. Egy-két számolási hiba 1 pont levonást, több pedig 2 pont levonást jelent. Ha egy megoldó az f -fel kezdődő topologikus sorrenddel dolgozik, akkor az inicializálás része a $\text{maxút}(f) = -\infty$ és $\text{előző}(f) = *$ értékek beállítása.

4. Egy $n \times n$ -es táblán lépkedünk egy huszárral (vagyis egy lóval), de minden lépésben vagy kettőt felfelé és egyet oldalra (azaz jobbra vagy balra), vagy pedig egyet felfelé és kettőt oldalra haladhatunk. A tábla néhány mezőjéről tudjuk, hogy azokra nem szabad rálépnünk. Adjunk $O(n^2)$ lépésszámú algoritmust annak meghatározására, hogy a tábla bal alsó sarkában álló huszárral legkevesebb hány lépéssel tudunk így eljutni a jobb felső sarokba, vagy igazolja, hogy ilyen út nem létezik.

I. megoldás (szélességi bejárással vagy „DAG-os algoritmussal”).

Gráf definiálása:

3 pont

Legkevesebb élű utat keresünk a bal alsó sarokból a jobb felsőbe:

2 pont

Szélességi bejárást használunk (vagy „DAG-os algoritmust”, és ezt lehet):

2 (=1+1) pont

Lépésszám (becslés az élek számára, éllista elkészítése, útkeresés lépésszáma, és ebből teljes lépésszám):

1+0+2+0 pont

EGY LEHETSÉGES HELYES MEGOLDÁS

Legyenek a G gráf csúcsai a tábla mezői, és egy mezőből pontosan akkor vezessen irányított él egy másikba, ha az első mezőről átléphetünk a másikra. Egy legkevesebb élű utat keresünk a tábla bal alsó sarkának megfelelő csúcsból a jobb felsőnek megfelelőbe, amit a bal alsó sarokból indított szélességi bejárással tudunk találni. A gráfnak n^2 csúcsa van, és minden csúcsból legfeljebb 4 él vezet ki, vagyis a gráfnak legfeljebb $4n^2$ éle van. Tehát ha G -nek az éllistáját készítjük el, akkor annak a mérete $O(n^2 + 4n^2) = O(n^2)$ lesz, és ezzel a szélességi bejárás lépésszáma is $O(n^2 + 4n^2) = O(n^2)$ lesz. Vagyis a teljes algoritmus lépésszáma $O(n^2) + O(n^2) = O(n^2)$.

EGY MÁSIK LEHETSÉGES HELYES MEGOLDÁS

A fenti G gráfban kereshetünk a „DAG-os algoritmussal” is legrövidebb utat úgy, hogy minden él súlyát 1-nek vesszük. Mivel minden lépéssel feljebb kerülünk a táblán, ezért a gráfban nincs irányított kör, tehát használhatjuk az algoritmust. A lépésszám, pontosan ugyanúgy mint a szélességi bejárásnál, $O(n^2)$ lesz.

II. megoldás (dinamikus programozással).

EGY LEHETSÉGES HELYES MEGOLDÁS

Részfeladatok megfogalmazása:

(2 pont)

Számozzuk a táblázat sorait lentől felfelé, az oszlopait pedig balról jobbra, 1-től n -ig. Jelölje $T[i, j]$ azt, hogy az $(1, 1)$ -edik mezőből legkevesebb hány lépéssel tudunk eljutni az (i, j) -edik mezőbe, ahol $i, j \in \{1, \dots, n\}$.

Inicializálás:

(1 pont)

Vegyünk fel még egy (-1) -edik, 0 ., $(n+1)$ -edik és $(n+2)$ -edik oszlopot, illetve egy (-1) -edik, 0 . sort a T táblázathoz, és ezeknek minden eleme legyen ∞ . Továbbá legyen $T[1, 1] = 0$.

Kitöltési szabály:

(3 pont)

$$T[i, j] = \begin{cases} \min\{T[i-1, j-2], T[i-1, j+2], T[i-2, j-1], T[i-2, j+1]\} + 1, & \text{ha az } (i, j)\text{-edik mezőre ráléphetünk,} \\ \infty, & \text{különben.} \end{cases}$$

Kitöltési sorrend:

(1 pont)

Alulról felfelé haladva, soronként balról jobbra.

Válasz kiolvasása:

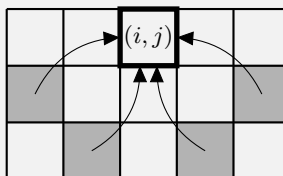
(1 pont)

A keresett érték $T[n, n]$. (Ha ez végtelen, akkor az azt jelenti, hogy ebbe a mezőbe nem lehet eljutni.)

Helyesség bizonyítása:

(1 pont)

Minden mezőbe (legfeljebb) négy másikból léphetünk be, ezért az oda vezető legrövidebb út hossza éppen eggyel több, mint az említett négy másik mezőbe vezető utak közül a legrövidebbé. A kérdés pedig az volt, hogy az (n, n) -edik mezőbe legkevesebb hány lépéssel tudunk eljutni.



Lépésszám:

(1 pont)

Egy $(n+2) \times (n+2)$ -es táblázatot töltünk ki, és minden mező kitöltéséhez $O(1)$ lépés kell. A válasz kiolvasása $O(1)$ lépés. Ez összesen $O((n+2)^2) \cdot O(1) + O(1) = O(n^2)$ lépés.

Természetesen az is jó, ha egy megoldó az inicializálásban az extra sorok és oszlopok felvétele helyett azt mondja a kitöltési szabályban, hogy ha a képletben szereplő tagok indexei nem 1 és n közöttiek, akkor azt a tagot nem vesszük figyelembe.

Ha egy megoldó érdemi kísérletet tesz a kitöltési szabály elmagyarázására, akkor a helyesség bizonyítására járó 1 pontot kapja meg.

5. Szomszédossági mátrixával adott egy n -csúcú, irányított G gráf. A gráf néhány éle piros, a többi nem. Adott még a gráfban egy s és egy t csúcs. Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben akarunk meghatározni egy olyan st -utat, ami a lehető legkevesebb piros színű élt tartalmazza?

Megfelelő élsúlyozás választása:	2 pont
Legrövidebb st -utat keresünk:	2 pont
Megfelelő algoritmus kiválasztása:	2 pont
Algoritmus használhatóságának megindoklása:	1 pont
Lépésszám (élsúlyozott szomszédossági mátrix elkészítése, algoritmus futtatása, összegzés):	1+2+0 pont

EGY LEHETSÉGES HELYES MEGOLDÁS

Legyen a piros élek súlya 1, a többié pedig 0. Erre az élsúlyozásra nézve kell egy legrövidebb st -utat keresnünk. Mivel minden él súlya nemnegatív, ezért használhatjuk erre az (s csúcsból indított) Dijkstra-algoritmust. Az élsúlyozott gráf szomszédossági mátrixának elkészítése $O(n^2)$ lépés (az eredeti gráf szomszédossági mátrixában a neméleket jelentő 0-kat át kell írni ∞ -re, a piros éleknek megfelelő 1-eseket megtartjuk, a nempiros éleknek megfelelő 1-eseket pedig 0-ra cseréljük). A Dijkstra-algoritmus futtatása $O(n^2)$ lépés (hiszen az élsúlyozás után a gráfnak továbbra is n csúcsa lesz). Ez összesen $O(n^2) + O(n^2) = O(n^2)$ lépés.

6. Egy zárt rendszerben egy részecske n különböző állapotba hozható. Egy $n \times n$ -es mátrixban adott, hogy egy állapotból egy másikba való átmenethez mennyi energia szükséges vagy mennyi energia szabadul fel, ha létezik ilyen közvetlen átmenet. Tudjuk, hogy a részecskét nem lehet úgy visszajuttatni egy korábbi állapotába, hogy ezen folyamat során összességében energia szabaduljon fel. Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^3)$ lépésben akarjuk eldönteni, hogy egy adott A állapotból eljuttatható-e a részecske egy másik adott B állapotba úgy, hogy a folyamat során összességében energia szabaduljon fel?

Gráf definiálása (csúcsok, élek):	2 pont
Megfelelő élsúlyozás definiálása:	1 pont
A legrövidebb AB -út hossza negatív-e:	2 pont
Megfelelő algoritmus kiválasztása:	2 pont
Algoritmus használhatóságának megindoklása:	2 pont
Lépésszám:	1 pont

EGY LEHETSÉGES HELYES MEGOLDÁS

Legyenek a G gráf csúcsai a lehetséges állapotok, és pontosan akkor vezessen él egy csúcsból egy másikba, ha az elsőnek megfelelő állapotból létezik közvetlen átmenet a másikba. Ha egy közvetlen átmenethez energia szükséges, akkor a megfelelő él súlya legyen pozitív előjellel a szükséges energia mennyisége, ha viszont a közvetlen átmenet során energia szabadul fel, akkor az él súlya legyen a felszabaduló energia mennyisége negatív előjellel. Az a kérdés, hogy vezet-e A -ból B -be olyan irányított út, aminek a hossza negatív. Mivel a feladat szövege szerint ebben a gráfban nincs negatív összhosszúságú irányított kör, ezért használhatjuk a Bellman–Ford-algoritmust egy legrövidebb AB -út meghatározására. Ha egy legrövidebb AB -út hossza negatív, akkor el lehet juttatni A -ból B -be a részecskét úgy, hogy ezzel összességében energia szabaduljon fel, különben viszont nem. Mivel a gráf szomszédossági mátrixszal adott, ezért az algoritmus lépésszáma $O(n^3)$.

7. Adott egy n és egy m darab karakterből álló szó, $s = s_1 \dots s_n$ és $t = t_1 \dots t_m$. Ezekben szeretnénk meghatározni egy leghosszabb majdnem közös részszó hosszát: azaz a legnagyobb olyan k számot, amihez létezik olyan $s_i s_{i+1} \dots s_{i+k-1}$ és $t_j t_{j+1} \dots t_{j+k-1}$, amelyek legfeljebb egy karakterben térnek el egymástól, azaz az $s_i = t_j$, $s_{i+1} = t_{j+1}$, \dots , $s_{i+k-1} = t_{j+k-1}$ feltételek közül legfeljebb az egyik nem teljesül. Adjunk erre a feladatra dinamikus programozást használó $O(nm)$ lépésszámú algoritmust.

Részfeladatok megfogalmazása: (2 pont)

Tetszőleges $i, j \in \{1, \dots, n\}$ esetén legyen $A[i, j]$ az a legnagyobb ℓ szám, amire az $s_1 \dots s_i$ és a $t_1 \dots t_j$ karaktorsorozatok utolsó ℓ karaktere megegyezik. Legyen továbbá $B[i, j]$ az a legnagyobb ℓ' szám, amire ugyanezen karaktorsorozatok utolsó ℓ' karaktere pontosan egy kivételével megegyezik.

Inicializálás: (1 pont)

Legyen tetszőleges $i, j \in \{1, \dots, n\}$ esetén $A[0, j] = A[i, 0] = 0$ és $B[0, j] = B[i, 0] = -\infty$.

Kitöltési szabály: (3 pont)

Tetszőleges $i, j \in \{1, \dots, n\}$ esetén

$$A[i, j] = \begin{cases} A[i-1, j-1] + 1, & \text{ha } s_i = t_j, \\ 0, & \text{különben,} \end{cases} \quad \text{és} \quad B[i, j] = \begin{cases} B[i-1, j-1] + 1, & \text{ha } s_i = t_j, \\ A[i-1, j-1] + 1, & \text{különben.} \end{cases}$$

Kitöltési sorrend: (1 pont)

Először az A táblázatot, majd a B -t, mindkettőt „fentről lefelé”, soronként „balról jobbra”, azaz i megy 1-től n -ig, és minden i esetén j is megy 1-től n -ig.

Válasz kiolvasása: (1 pont)

A keresett érték $\max \{A[i, j], B[i, j] \mid 1 \leq i, j \leq n\}$.

Helyesség bizonyítása: (1 pont)

Tekintsük az $s_1 \dots s_i$ és $t_1 \dots t_j$ karaktorsorozatok leghosszabb közös végződésének, illetve a pontosan egy karakterben eltérő leghosszabb végződésének hosszát.

Ha $s_i = t_j$, akkor ezek éppen eggyel lesznek többek, mint amennyik ezek az $s_1 \dots s_{i-1}$ és $t_1 \dots t_{j-1}$ karaktorsorozatokban voltak.

Ha viszont $s_i \neq t_j$, akkor a két karaktorsorozatnak nincs közös végződése, a pontosan egy karakterben eltérő leghosszabb végződésük pedig éppen eggyel hosszabb, mint az $s_1 \dots s_{i-1}$ és $t_1 \dots t_{j-1}$ karaktorsorozatok leghosszabb közös végződése volt.

Ezen leghosszabb majdnem közös végzödések közül a leghosszabb lesz az $s_1 \dots s_n$ és $t_1 \dots t_m$ karaktorsorozatokban a leghosszabb közös részszó hossza.

Az inicializálás jelentése, hogy egy „üres” és egy „nemüres” karaktorsorozatnak 0-hosszú minden közös végződése, és nincsen nekik pontosan egy karakterben eltérő végződésük.

Lépésszám: (1 pont)

Két $(n+1) \times (n+1)$ -es táblázatot töltünk ki, és minden mező kitöltéséhez $O(1)$ lépés kell. A válasz kiolvasása $O(2n^2)$ lépés. Ez összesen $O((n+1)^2) \cdot O(1) + O(2n^2) = O(n^2)$ lépés.

Ha egy megoldó érdemi kísérletet tesz a kitöltési szabály elmagyarázására, akkor a helyesség bizonyítására járó 1 pontot kapja meg.

A $B[i, j]$ részfeladatokat úgy is definiálhattuk volna, hogy pontosan 1 eltérés helyett legfeljebb 1 eltéréssel dolgozunk (és ekkor értelemszerűen más képletekkel kell dolgozni).