

A rendelkezésre álló idő: 90 perc.

Minden feladat egységesen 10 pontot ér. Az aláírás megszerzéséhez legalább 24 pont szükséges. A teljes pontszám eléréséhez a megoldás(oka)t indokolni kell!

Kérjük, írja fel a **nevét**, **NEPTUN-kódját** és a **gyakorlatvezető nevét** az összetűzött lapok jobb felső sarkába.

Általános alapelvek. A pontozási útmutató célja, hogy a javítók a dolgozatokat egységesen értékeljék. Ezért az útmutató minden feladat (legalább egy lehetséges) megoldásának főbb gondolatait és az ezekhez rendelt részpontszámokat közli. Az útmutatónak nem célja a feladatok teljes értékű megoldásának részletes leírása; a leírt lépések egy maximális pontszámot érő megoldás vázlatának tekinthetők. 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. Így például az anyagban szereplő ismeretek, definíciók, tételek pusztán leírása azok alkalmazása nélkül nem ér pontot (még akkor sem, ha egyébként valamelyik leírt tény a megoldásban valóban szerephez jut). 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, amelyből a dolgozatban leírt gondolatmenet alkalmas kiegészítésével a feladat hibátlan megoldása volna kapható. Ha egy megoldó egy feladatra több, egymástól lényegesen különböző megoldást is elkezd, akkor legfőképpen 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). Az útmutatóban szereplő részpontszámok szükség esetén tovább is oszthatók. Az útmutatóban leírttól eltérő jó megoldás természetesen maximális pontot ér, de bizonyítás nélkül csak az előadáson szereplő tételekre és állításokra lehet hivatkozni.

1. Tegyük fel, hogy minden $n \geq 2025$ esetén teljesül:

$$2 \cdot n \leq f(n) \leq 10 \cdot n \cdot \log_2 n,$$

és minden $n \geq 428$ esetén teljesül:

$$2 \cdot \log_2 n \leq g(n) \leq 10^5 \cdot \sqrt{n}.$$

Mutassa meg megfelelő $c > 0$ és $n_0 \in \mathbb{N}$ konstansok megadásával, hogy $f(n) + g(n) \in O(n \log_2 n)$.

Megoldás: A feltételek szerint minden $n \geq 2025$ esetén

$$f(n) \leq 10 \cdot n \cdot \log_2 n \quad \text{és} \quad g(n) \leq 10^5 \cdot \sqrt{n}.$$

2 pont

Ezért

$$f(n) + g(n) \leq 10 \cdot n \cdot \log_2 n + 10^5 \cdot \sqrt{n}.$$

1 pont

Mivel $10^5 \cdot \sqrt{n} \leq 10^5 \cdot n \cdot \log_2 n$, igaz, ha $n \geq 2$ (tehát $n \geq 2025$ esetén is), ezért

2 pont

$$f(n) + g(n) \leq 10 \cdot n \cdot \log_2 n + 10^5 \cdot n \cdot \log_2 n = (10^5 + 10) \cdot n \cdot \log_2 n.$$

2 pont

teljesül, ha $n_0 = 2025$ és $c = 10^5 + 10$, ezért

2 pont

$$f(n) + g(n) \in O(n \log_2 n).$$

1 pont

2. Tegyük fel, hogy egy rendezett tömb n különböző egész számot tartalmaz az $\{1, 2, 3, \dots, n, n+1, n+2\}$ halmazból, azaz pontosan **két** szám hiányzik. Adjon algoritmust, ami megtalálja a két hiányzó számot és legfeljebb $O(\log n)$ összehasonlítást használ.

Megoldás: Legyen $A[1 \dots n]$ a rendezett tömb, amely az $\{1, 2, \dots, n+2\}$ elemeiből két hiányzó szám ($m_1 < m_2$) mellett tartalmaz n elemet. Ekkor $i < m_1$ esetén $A[i] = i$, $m_1 \leq i < m_2 - 1$ esetén $A[i] = i + 1$, és $m_2 \leq i - 1$ esetén pedig $A[i] = i + 2$.

2 pont

Ha az m_2 határa egygyel elcsúszik.

-1 pont

Először keressük a legkisebb i indexet, amelyre $A[i] < i$.

1 pont

Ezt a bináris kereséshez hasonlóan a tesszük. A középsőre megnézzük, hogy $A[i] = i$ teljesül-e. Ha igen, akkor nagyobb indexűek között folytatjuk, ha nem akkor a kisebb indexűeknél.

2 pont

Ezután $m_1 = i$ (ha $i = 1$ és $A[1] \neq 1$, akkor $m_1 = 1$; ha minden i esetén $A[i] = i$, akkor $m_1 = n + 1$).

1 pont

A zárójeles rész hiánya miatt nem kell levonni.

Most keressük a legkisebb $j \geq m_1$ indexet, amelyre $A[j] < j + 1$.

2 pont

Ezt az előző bináris kereséshez hasonlóan találjuk meg. Ekkor $m_2 = j + 1$ (ha nincs ilyen j , akkor $m_2 = n + 2$). 1 pont

Mivel mindkét keresés $O(\log n)$ összehasonlítást igényel, az egész algoritmus $O(\log n)$ összehasonlítást használ. 1 pont

3. Adott két tömb, mindkettőben n különböző egész szám van, de a két tömbnek lehetnek közös elemei. A tömbök nem feltétlenül rendezettek. Tervezzon olyan algoritmust, amely kiírja azokat a számokat, amik mindkét tömbben szerepelnek vagy jelzi, ha nincsen a tömböknek közös eleme. Az algoritmus legfeljebb $O(n \log n)$ összehasonlítást használjon.

Megoldás: Legyen A és B a két tömb, melyek mindegyikében n különböző egész szám szerepel. Az algoritmus a következő lépésekből áll (futásidőkkel):

1. Rendezés: Rendezzük az A tömböt $O(n \log n)$ időben (pl. összefésüléses rendezéssel).

3 pont

2. Keresés: Vegyük sorra a B tömb elemeit, és minden $x \in B$ esetén végezzünk bináris keresést a rendezett A -ban, hogy ellenőrizzük, x szerepel-e benne.

3 pont

Mivel bináris keresés $O(\log n)$ összehasonlítást használ egy elemre, a teljes keresési fázis $O(n \log n)$ lesz.

2 pont

3. Kimenet: Írjuk ki az összes olyan x -et, amelyet a keresés megtalált. Ha egyetlen közös elem sem található, jelezzük, hogy nincs közös elem.

2 pont

4. Az alábbi szomszédossági mátrix segítségével adott egy súlyozott, irányított gráf, melyben topologikus rendezés a csúcsok a, b, c, d, e, f, g sorrendje. (Ezt nem kell belátni.)

	a	b	c	d	e	f	g
a	0	∞	-2	∞	∞	∞	∞
b	∞	0	∞	-3	∞	2	∞
c	∞	∞	0	1	4	∞	∞
d	∞	∞	∞	0	∞	∞	-1
e	∞	∞	∞	∞	0	-2	∞
f	∞	∞	∞	∞	∞	0	3
g	∞	∞	∞	∞	∞	∞	0

- (a) Az órán tanult, a topologikus sorrendet használó eljárással számítsa ki a legrövidebb utak hosszát a b csúcsból az összes többi csúcsba.

- (b) Az (a) pontban használt eljárás segítségével határozzon meg egy legrövidebb utat a b pontból az f pontba.

Megoldás:

Topologikus sorrend szerint megyünk végig:

1 pont

a: a a forrás előtt van, így $D[a] = \infty$.

1 pont

b: Forrás: $D[b] = 0$.

1 pont

c: Bejövő élek:

$$a \rightarrow c : w(a,c) = -2, \quad D[a] + (-2) = \infty - 2 = \infty;$$

Tehát $D[c] = \infty$.

1 pont

d: Bejövő élek:

$$b \rightarrow d : w(b,d) = -3, \quad D[b] + (-3) = 0 - 3 = -3;$$

$$c \rightarrow d : w(c,d) = 1, \quad D[c] + 1 = \infty + 1 = \infty.$$

Ezért $D[d] = -3$ (és $\text{elozo}[d] = b$).

1 pont

e: Bejövő élek:

$$c \rightarrow e : w(c,e) = 4, \quad D[c] + 4 = \infty;$$

Így $D[e] = \infty$.

1 pont

f: Bejövő élek:

$$b \rightarrow f : w(b,f) = 2, \quad D[b] + 2 = 0 + 2 = 2;$$

$$e \rightarrow f : w(e,f) = -2, \quad D[e] + (-2) = \infty - 2 = \infty.$$

Tehát $D[f] = 2$ (és $\text{elozo}[f] = b$).

1 pont

g: Bejövő élek:

$$d \rightarrow g : w(d,g) = -1, \quad D[d] + (-1) = -3 - 1 = -4;$$

$$f \rightarrow g : w(f,g) = 3, \quad D[f] + 3 = 2 + 3 = 5;$$

Így $D[g] = \min\{-4, 5\} = -4$ (és $\text{elozo}[g] = d$).

1+1 pont

Eredmény: Legrövidebb utak b forrástól: *Nem kell feltétlen külön leírni, ha már ki van számolva.*

$$\begin{aligned} D[a] &= \infty \quad (a \text{ nem érhető el}); \\ D[b] &= 0; \\ D[c] &= \infty; \\ D[d] &= -3; \\ D[e] &= \infty; \\ D[f] &= 2; \\ D[g] &= -4. \end{aligned}$$

(b) Legrövidebb út $b \rightarrow f$:

Mivel $D[f] = 2$ és $\text{elozo}[f] = b$, a legrövidebb út egyszerűen: $b \rightarrow f$, súlya 2.

1 pont

5. Egy irányított, élsúlyozott gráf csúcsait és éleit a jobb oldali ábra mutatja.

Futtassa a Dijkstra-algoritmust az A pontból kiindulva.

- Határozza meg az összes többi csúcsba vezető legrövidebb utak hosszát!
- Adjon meg egy-egy legrövidebb utat A -ból a következő csúcsokhoz: C , E , G .

(Indokolni nem kell, de látszódjon, hogy lépésenként hogyan változik a D és a P tömb illetve a KÉSZ halmaz.)

Megoldás:

It.	Új csúcs KÉSZ-ben	$D(\cdot)$							$P(\cdot)$						
		A	B	C	D	E	F	G	A	B	C	D	E	F	G
0	A	0	4	∞	3	∞	∞	∞	–	A	–	A	–	–	–
1	D	0	4	∞	3	8	∞	4	–	A	–	A	D	–	D
2	B	0	4	6	3	6	∞	4	–	A	B	A	B	–	D
3	G	0	4	6	3	6	∞	4	–	A	B	A	B	–	D
4	C	0	4	6	3	6	7	4	–	A	B	A	B	C	D
5	E	0	4	6	3	6	7	4	–	A	B	A	B	C	D
6	F	0	4	6	3	6	7	4	–	A	B	A	B	C	D

6 pont

(a) Legrövidebb utak hosszai A -ból:

1 pont

$$\begin{aligned} A &: 0, \\ B &: 4, \\ C &: 6, \\ D &: 3, \\ E &: 6, \\ F &: 7, \\ G &: 4. \end{aligned}$$

(b) Egy-egy legrövidebb út A -ból:

- $\underline{A \rightarrow C}$: $P(C) = B$ és $P(B) = A \Rightarrow$ út: $A \rightarrow B \rightarrow C$ (összsúly: $4 + 2 = 6$).
- $\underline{A \rightarrow E}$: $P(E) = B$ és $P(B) = A \Rightarrow$ út: $A \rightarrow B \rightarrow E$ (összsúly: $4 + 2 = 6$).
- $\underline{A \rightarrow G}$: $P(G) = D$ és $P(D) = A \Rightarrow$ út: $A \rightarrow D \rightarrow G$ (összsúly: $3 + 1 = 4$).

1 pont

1 pont

1 pont

6. Az egyetem újonnan szerződött burkolója egy különleges feladaton töri a fejét: Egy hosszú folyosó $3 \times n$ -es padlóját kell lefedni 3×1 -es járólappal (a járólapok keresztben és hosszában is elhelyezhetők, tehát 3×1 -es vagy 1×3 -es alakban is használhatók), az alábbi feltételekkel:

- a járólapok nem fedhetik egymást,
- nem lóghatnak túl a padló szélén, és
- a teljes padlót pontosan le kell fedni.

Adjon meg egy $O(n)$ lépésben működő algoritmust, amely kiszámolja, hogy hányféleképpen lehet szabályosan lefedni a padlót ezekkel a feltételekkel.

Megoldás: $T(i)$ jelentése: megadja annak a számát, hányféleképpen lehet lefedni egy $3 \times i$ -es padlót 3×1 és 1×3 járólapokkal a feladatban adott feltételekkel. **1 pont**

Sorrend: Az értékeket az $i = 0$ -tól $i = n$ -ig kell kiszámítani, tehát az értékek növekvő sorrendben kerülnek kiszámításra. **1 pont**

Kezdő lépések: **2 pont**

$$\begin{aligned} T(0) &= 1 \quad (\text{üres padló esetén 1 triviális lefedés}), \\ T(1) &= 1 \quad (\text{egy } 3 \times 1 \text{ padló egyetlen módon fedhető le}), \\ T(2) &= 1 \quad (\text{egy } 3 \times 2 \text{ padlót csak kizárólag } 3 \times 1\text{-es járólapokkal lehet lefedni}). \end{aligned}$$

Vége: Miután elértük $i = n$ -t, a padló lefedési lehetőségeinek száma $T(n)$ lesz, ezt adjuk vissza az algoritmus eredményeként. **1 pont**

Tovább lépés: Minden i -re, ahol $3 \leq i \leq n$ esetén:

$$T(i) = T(i-1) + T(i-3),$$

attól függően, hogy az i -edik 3×1 -es vagy 1×3 -as volt. **2 pont**

Helyesség: Az i -edik „oszlopban” lévő 3 négyzetet vagy egy 3×1 -es lappal fedjük le, vagy pedig 3 db 1×3 -as fedi le őket. Más lehetőség nincs. **2 pont**

Futásidő: Mivel a kezdőértékek beállítása $O(1)$ és a tovább lépési ciklus $i = 3$ -tól $i = n$ -ig szintén minden i -re $O(1)$ idejű (hiszen csak 2 előző értéket kell kiolvasni és összeadni a tömbben), az algoritmus futási ideje: $O(n)$. **1 pont**

7. Algoritmisztán térképe egy szomszédsági mátrixszal adott irányított gráf, melynek csúcsai a városok, irányított élei pedig a városok között vezető közvetlen utak. Az utak használatáért általában fizetni kell, 2025 útszakasz kivételével ez az ár pozitív, de erre a 2025 útra nulla. Az algoritmisztáni árhivatal azt a szabályt hozta, hogy egy utazó egy út során legfeljebb három ingyenes útszakaszt használhat, egyébként összedől az ország költségvetése. Adjon $O(n^2)$ lépésszámú algoritmust, ami meghatározza a legolcsóbb olyan utat egy adott A városból egy adott B városba, ami legfeljebb három ingyenes utat használ.

Megoldás: Legyen $G = (V, E)$ az eredeti, szomszédsági mátrixos irányított gráf (a városokat csúcsok, az utak élek jelölik), és legyen $F \subseteq E$ az ingyenes útszakaszok halmaza. A szabály azt írja elő, hogy egy utazó legfeljebb 3 ingyenes útszakaszt használhat egy út során.

Algoritmus:

Minden olyan S részhalmazon iterálunk, melyre $S \subseteq F$ és $|S| = 3$, **2 pont**
ezek száma $\binom{2025}{3}$, ami egy konstans. **1 pont**

- (a) Egy adott S -re módosítjuk a gráfot úgy, hogy az új élhalmaz $E_S = \{e \in E \mid e \notin F\} \cup S$ tartalmazza az összes fizetős élt, valamint az S -ben kiválasztott ingyenes éleket. **2 pont**
- (b) Minden módosított gráfra $G_S = (V, E_S)$ futtassuk a Dijkstra algoritmust A városból B városba, és jelöljük $C(S)$ -sel a minimális útköltséget. **2 pont**
- (c) Válasszuk ki a legkisebb költséget: $C^* = \min\{C(S) \mid S \subseteq F; |S| = 3\}$. **1 pont**

Futásidő: Mivel az ingyenes élekből választott részhalmazok száma konstans, és egy Dijkstra algoritmus a szomszédsági mátrix miatt $O(n^2)$ lépést igényel, a teljes algoritmus futása $O(n^2)$. **2 pont**