Algoritmuselmélet (BMEVISZA213) feladatgyűjtemény

Drótos Márton

2012. május 30.

Tartalomjegyzék

1. Előszó, használati tanácsok

Jelen feladatgyűjteményben többnyire az elmúlt években a gyakorlataimon feladott feladatok szerepelnek. Sok feladat régebbi ZH-kon, vizsgákon szerepelt.

A megoldások főleg az ötletek leírására helyezik a hangsúlyt, sokszor az egyszerű technikai részletek hiányoznak. Feltételezem, hogy ezeket mindenki képes önállóan is megfogalmazni. Mechanikus feladatból kevés van (nem is szerepel minden típus), hiszen ezeket az előadásokból és a weben található animációkból sokkal könnyebb megtanulni.

A megoldásokat megtanulni, vagy a feladatokat megoldásokkal együtt nézni nem tanácsos, hiszen a számonkéréseken teljesen ismeretlen feladatok lesznek, amiket önállóan kell megoldani. A legcélravezetőbb módszer az, hogy ha valaki önállóan megpróbálja üres papíron megoldani a feladatokat, majd az itt található megoldással összeveti (különös tekintettel az indoklásra).

Minden témakörben az első néhány feladat bevezető és/vagy típusfeladat. Természetesen ez nem jelenti azt, hogy csak ilyenek szerepelnek számonkérésen. A többi feladat többé-kevésbé nehézség szerint van rendezve.

A jegyzet kinyomtatását nem javaslom, egyrészt papírtakarékossági okokból, másrészt elképzelhető, hogy az idők folyamán néhány helyen javítani fogok rajta, esetleg új feladatokkal bővül.

2. Nagyságrendek

1. Bizonyítsuk be, hogy

(a) $\log_2 f(n) = \Theta(\log_{100} f(n)) \quad (f(n) > 0)$
(b) $f(x) = a_k x^k + a_{k-1}x^{k-1} + \cdots + a_0 \quad (a_k \neq 0) \quad \Rightarrow \quad f(n) = \Theta(n^k)$
(c) $2^{n+1} = O(2^n)$, de $2^{2n} \neq O(2^n)$
(d) $\max(f(n), g(n)) = \Theta(f(n) + g(n)) \quad (f(n), g(n) > 0)$

2. [Vizsga: 2007. június 12.] Egy $\mathcal{A}$ algoritmusról azt tudjuk, hogy n hosszú bemeneteken a lépésszáma $O(n \log n)$. Lehetséges-e, hogy

(a) van olyan x bemenet, amin a lépésszáma x^3?
(b) minden x bemeneten legfeljebb $2007|x|$ lépést használ?

(Szokás szerint $|x|$ az x szó hosszát jelöli.)

3. Jelöljük $T(n)$-nel egy algoritmus legnagyobb lehetséges lépésszámát az n méretű inputokon. Tudjuk, hogy $T(n) \leq 10$, ha $n \leq 5$ és $T(n) \leq T(n-1) + n/3$, ha $n > 5$. Ekkor mit tudunk mondani $T(n) = O(n)$, $T(n) = O(n^2)$ és $T(n) = O(n^3)$ egyenlőségek helyességéről?

4. [ZH: 2011. március 28.] Egy problémára két algoritmusunk van.
Az $\mathcal{A}$ algoritmus az $n \geq 2$ méretű problémából 10 lépéssel 2 db $n-1$ méretűt készít és ezeket oldja meg rekurzívan.
A $\mathcal{B}$ az $n \geq 2$ problémából 3 lépéssel 4 db $n-1$ méretűt készít és ezeket oldja meg rekurzívan. Az $n = 1$ esetben mindkét eljárás 1 lépést használ.
Melyik algoritmus lesz nagy n értékekre gyorsabb?

---

5. Igaz-e, hogy

(a) ha $f = O(g)$ és $g = O(h)$, akkor $f = O(h)$
(b) ha $f = \Omega(g)$ és $g = \Omega(h)$, akkor $f = \Omega(h)$

6. Az alábbi függvényeket rendezzük olyan sorozatba, hogy ha f_i után közvetlenül f_j következik a sorban, akkor $f_i(n) = O(f_j(n))$ teljesüljön!

$$f_1(n) = 8n^{2.5}, \quad f_2(n) = 5\sqrt{n} + 1000n, \quad f_3(n) = 2^{\log^2 n}, \quad f_4(n) = 2008n^2 \log n$$

7. Az $\mathcal{A}$ algoritmusról azt tudjuk, hogy n hosszú bemeneteken a lépésszáma $O(n^2)$. Lehetséges-e, hogy

(a) $\forall n$ hosszú bemeneten $O(n)$ lépést használ?
(b) $\exists x$, hogy az x bemeneten az algoritmus lépésszáma $10|x|^2 \log |x| - 800$ (ahol $|x|$ az x bemenet hosszát jelöli)?

8. [Vizsga: 2007. június 19.] Az alábbi függvényeket rendezze olyan sorozatba, hogy ha f_i után közvetlenül f_j következik a sorban, akkor $f_i(n) = O(f_j(n))$ teljesüljön!

$$f_1(n) = 2^{100n} - 2^{50n}, \quad f_2(n) = 2007n^3, \quad f_3(n) = 3^{3n}$$

9. [Vizsga: 2007. június 5.] Jelölje egy algoritmus maximális lépésszámát az n hosszú bemeneteken $L(n)$. Azt tudjuk, hogy minden $n = 2k > 4$ páros számra $L(2k) \leq L(2k-2) + 1$ teljesül, és hogy $L(4) = 10$. Következik-e ebből, hogy az algoritmus lépésszáma $O(n)$?

10. [PZH: 2011. április 22.] Egy problémára két algoritmusunk van.
Az $\mathcal{A}$ algoritmus az $n \geq 2$ méretű problémából 5 lépéssel 2 db legfeljebb $n/2$ méretűt készít és ezeket oldja meg rekurzívan.
A $\mathcal{B}$ algoritmusról azt tudjuk, hogy lépésszáma az n méretű problémákon $O(n^2)$.
Ha ennyiből lehetséges, határozza meg, melyik algoritmus lesz nagy n értékekre gyorsabb! Ha ennyi információból még nem következik, hogy $\mathcal{A}$ vagy $\mathcal{B}$ lesz a gyorsabb, akkor indokolja meg, miért nem!

11. Tudjuk, hogy $f(x) = O(h(x))$ és $g(x) = O(h(x))$. Igaz-e, hogy
(a) ha $h(x) = 3x$, akkor $f(g(x)) = O(h(x))$
(b) $f(g(x)) = O(h(x))$ $\forall h$ függvényre

3. Dinamikus programozás

1. **[ZH: 2008. március 28.]** Egy $n \times n$ méretű táblázat minden eleme egy egész szám. A táblázat bal alsó sarkából akarunk eljutni a jobb felső sarkába úgy, hogy egy lépésben a táblázatban vagy felfelé vagy jobbra egyet lépünk. Azt szeretnénk, hogy a lépegetés során látott elemek növekvő sorrendben kövessék egymást. Egy ilyen út értéke a benne szereplő számok összege. Adjon $O(n^2)$ futási idejű algoritmust, ami meghatározza, hogy az adott táblázatban a szabályok szerinti utak értékei között mekkora a legnagyobb!
2. Adott egy fa, melynek csúcsaihoz súlyok vannak rendelve. Adjunk lineáris algoritmust, ami meghatározza a fában található maximális súlyú független ponthalmaz súlyát!
3. **[Vizsga: 2007. június 12.]** Egy n és egy m karakterből álló szövegben meg akarjuk találni a legnagyobb azonos darabot, azaz ha az egyik szöveg $a_1a_2 \cdots a_n$ és a másik $b_1b_2 \cdots b_m$, akkor olyan $1 \leq i \leq n$ és $1 \leq j \leq m$ indexeket keresünk, hogy

$$a_{i+1} = b_{j+1}, a_{i+2} = b_{j+2}, \ldots, a_{i+t} = b_{j+t}$$

teljesüljön a lehető legnagyobb t számra. Adjon erre a feladatra $O(mn)$ lépést használó algoritmust!

---

4. **[Vizsga: 2007. május 29.]** Legyen $w = w_1w_2 \cdots w_n$ egy n betűből álló szó. Hívjuk részszónak w egy tetszőleges $w_iw_{i+1} \cdots w_{i+k}$ darabját ($1 \leq i \leq n-1$, $1 \leq k \leq n-i$). Adjunk algoritmust, ami $O(n)$ lépésben meghatározza az összes a-val kezdődő és b-re végződő részszó számát.
5. Egy játékban egy $n \times m$ rács bal felső sarkából kell eljutnunk a jobb alsó sarokba. Egy lépés során a rács mentén vizszintesen vagy függőlegesen tudunk a következő rácspontba lépni. Azonban van néhány kereszteződés, ahova nem szabad lépnünk. Ezek helyét az R tömb írja le, $R[i, j] = 1$, ha az (i, j) kereszteződésbe nem léphetünk, egyébként $R[i, j] = 0$. Adjunk $O(nm)$ futási idejű algoritmust annak meghatározására, hogy pontosan $n + m - 2$ lépést téve a rácson hányféleképpen tudjuk a célt elérni!
6. **[ZH: 2010. április 19.]** Egy $n \times k$ méretű táblázatban van néhány megjelölt elem. A táblázat bal alsó sarkából akarunk eljutni a jobb felső sarkába úgy, hogy minden lépésben a táblázat egy eleméről vagy a közvetlen felette vagy a tőle jobbra lévő elemre mehetünk (ha van ilyen). Adjon $O(nk)$ idejű algoritmust, amely a megjelölt elemek helyét ismerve meghatározza, hogy egy ilyen út során maximálisan hány alkalommal tudunk megjelölt elemre lépni!
7. Legyenek $a_1, a_2, \ldots, a_n$ tetszőleges egész számok és $m < n^2$ egész. Adjunk algoritmust, amely a bináris alakjukkal megadott $a_1, a_2, \ldots, a_n$ és m számokról eldönti polinom időben, hogy az $a_1, a_2, \ldots, a_n$ számok közül kiválasztható-e néhány úgy, hogy az összegük m-mel osztva egyet adjon maradékul!
8. Adjunk algoritmust, ami egy n csúcsú fában lineáris időben meghatározza a fában levő leghosszabb út hosszát!
9. **[Vizsga: 2009. június 11.]** A véges hosszú 0-1 sorozatok egy részét valamilyen szempontból jónak tekintjük, a többit rossznak. Van egy A algoritmusunk, mely adott n hosszú 0-1 sorozatról $O(\log(n!))$ lépésben megmondja, hogy a sorozat jó vagy rossz.
Adjon olyan eljárást, mely A-t felhasználva, ha adott egy m hosszú 0-1 sorozat, $y = y_1y_2 \cdots y_m$, akkor eldönti, hogy y előáll-e néhány jó sorozat egymás után fűzéséből. Az eljárás lépésszáma összesen legyen $O(m^4)$.
Segítség: $\log(n!) \approx O(n \log n) = O(n^2)$.

10. Egy n szóból álló szöveget kell sorokra tördelni. A szöveg i-edik szava ℓ_i karakterből áll, egy sor s karakter hosszú. Ha egy sor a szöveg i-edik szavával kezdődik és a j-edik szóval végződik, akkor az elválasztó szóközöket is figyelembe véve $t = s-(\ell_i+\ell_{i+1}+\cdots+\ell_j+j-i)$ üres hely marad a sor végén. Egy ilyen sor hibája legyen t^2. A tördelés hibája a nem üres sorok hibáinak összege. Adjunk $O(n^2)$ lépéses algoritmust egy legkisebb hibájú tördelés meghatározására! (A szavak sorrendje rögzített.)

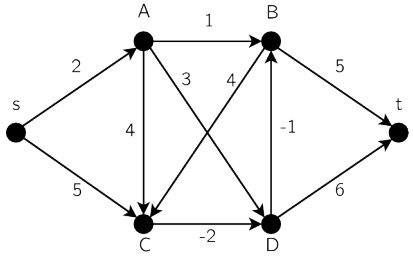
11. Az $1, 2, \ldots, n$ számoknak adott két permutációja, $x_1, \ldots, x_n$ és $y_1, \ldots, y_n$. A két sorozat egy közös részsorozata egy $1 \le i_1 < \cdots < i_k \le n$, és egy $1 \le j_1 < \ldots < j_k \le n$ indexsorozattal adható meg, ahol $x_{i_m} = y_{j_m}$ teljesül minden $1 \le m \le k$ esetén. Adjunk $O(n^2)$ lépésszámú algoritmust, ami az x és y sorozatokban meghatároz egy leghosszabb közös részsorozatot!

12. [MSc felvételi 2009. tavasz] Adott egy n és egy m hosszú 0-1 sorozat, $a_1, a_2, \ldots, a_n$, illetve $b_1, b_2, \ldots, b_m$. Ezek alapján egy T tömböt töltöttünk ki a következő módon:
Ha $0 \le i \le n$, akkor $T[i, 0] = 0$. Ha $0 \le j \le m$, akkor $T[0, j] = 0$.
Ha $1 \le i \le n$ és $1 \le j \le m$, akkor $T[i, j] = \begin{cases} T[i-1, j-1]+1 & \text{ha } a_i = b_j \\ \max(T[i, j-1], T[i-1, j]) & \text{ha } a_i \ne b_j \end{cases}$
Írja le, hogy mi a jelentése a $T[i, j]$ értéknek! A két sorozatnak milyen tulajdonságát adja meg a $T[n, m]$ érték?

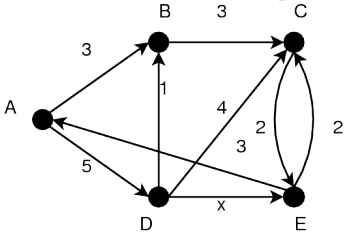
13. Adottak $M_1, M_2, \ldots, M_n$ munkák $H_1, H_2, \ldots, H_3$ határidőkkel és $P_1, P_2, \ldots, P_n$ profitokkal. Mindegyik munka elvégzése pontosan 1 napunkba kerül (egy napon csak egy munkát végezhetünk el). Adjunk $O(n^2)$ lépésszámú algoritmust, amely meghatározza, hogy mely munkákat vállaljuk el a profit maximalizálása érdekében!

4. Szélességi bejárás, legrövidebb utak, párosítások

1. Határozzuk meg a Bellmann-Ford algoritmussal a legrövidebb utat s és t között, nyomon követve az algoritmust!



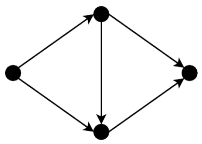
2. [ZH: 2009. április 24.] Dijsktra-algoritmussal határozza meg az alábbi gráfban az A pontból az összes többi pontba menő legrövidebb utak hosszát az x pozitív valós paraméter függvényében. Minden lépésnél írja fel a távolságokat tartalmazó D tömb állapotát és a KÉSZ halmaz elemeit.



3. Hogy néz ki egy irányítatlan teljes gráf szélességi bejárása?

4. [Vizsga: 2012. május 24.] Külföldi ösztöndíjakat szeretnénk megpályázni, ehhez ajánlólevelekre van szükségünk. Összesen n helyre adjuk be a pályázatot, minden pályázathoz két ajánlólevél szükséges. Ajánlólevelet m darab embertől tudunk kérni, de nem akarunk senkit sem túlságosan terhelni, ezért egy embertől legfeljebb egy ajánlólevelet akarunk kérni. (Az ajánlólevelek egyediek, egy levelet csak egy pályázatnál tudunk felhasználni.) Sajnos a pályázatok olyanok, hogy nem minden lehetséges ajánló személy jó minden helyre (azt tudjuk, ki hova jó). Adjon algoritmust, ami $O((n+m)nm)$ lépésben javasol egy lehetséges megoldást!

5. Éllistával adott a súlyozott élű $G(V, E)$ gráf. Tegyük fel, hogy az élek súlyai az $1, 2, 3$ számok közül valók. Javasoljunk $O(n+e)$ költségű algoritmust az $s \in V$ pontból az összes további $v \in V$ pontokba vivő legrövidebb utak hosszának meghatározására!
6. Legfeljebb hány komponensből állhat egy irányított gráf szélességi bejárása során keletkező erdő?
7. Határozzuk meg a következő gráfban az élsúlyokat úgy, hogy a Dijkstra algoritmus rossz eredményt adjon!



8. Adjuk meg az összes olyan minimális élszámú irányított gráfot (élsúlyokkal együtt), amely(ek)re az alábbi táblázat a Dijkstra algoritmusban szereplő $D[\,]$ tömb változásait mutathatja. Adjuk meg a legrövidebb utakat tartalmazó $P[\,]$ tömb állapotait is!

v_1	v_2	v_3	v_4	v_5	v_6
0	2	6	∞	∞	7
0	2	5	9	∞	6
0	2	5	6	9	6
0	2	5	6	8	6
0	2	5	6	7	6

9. **[ZH: 2009. április 24.]** Egy kezdő autóvezető a városban való közlekedése során szeretne gyakorlatának megfelelő útvonalat választani. Az úthálózat egy irányítatlan gráfként van megadva, a csúcsok a kereszteződések, az élek az utak, a csúcsoknál adott, hogy nehéz-e számára a kereszteződés. (Az hogy nehéz, a kereszteződés tulajdonsága, nem azon múlik, merről érkezik oda és és merre akar rajta áthaladni.) Írjon le egy algoritmust, amivel meg lehet határozni, hogy az autós az egyik adott csúcsnál levő otthonából mely csúcsokba tud autóval úgy eljutni, hogy útja során két nehéz csúcs soha nem jön közvetlenül egymás után. Az algoritmus lépésszáma éllistás megadás esetén legyen $O(n+e)$, ahol n a csúcsok és e az élek száma.
10. Egy G gráfban pontosan egy él súlya negatív, és nincs a gráfban negatív összsúlyú irányított kör. Adjunk $O(n^2)$ lépésszámú algoritmust az $s \in V(G)$ pontból az összes többi pontba vezető legrövidebb utak meghatározására!
11. **[ZH: 2007. április 27.]** Kutyasétáltatáskor egy parkban egy gazda rögzített, egyenes szakaszokból álló útvonalon halad, aminek töréspontjai $t_1, \ldots, t_n$, a bejáratot jelölje t_0, a kijáratot t_{n+1}. A kutyája szabadon szaladgál, de a t_i pontokban találkozik a gazdájával. A t_i és t_{i+1} pontokban való találkozás között a kutya szeretne egy fát is meglátogatni (minden $i = 0, 1, \ldots, n$ esetén legfeljebb egyet-egyet). Legyenek adottak az $s(t_i, t_{i+1})$ távolságok ($0 \leq i \leq n$), valamint minden fának az összes t_i ponttól vett távolsága. Tegyük fel, hogy két találkozás között a kutya legfeljebb kétszer akkora távolságot tud megtenni, mint a gazda. Adjon algoritmust, ami segít a kutyának eldönteni, hogy mikor melyik fát látogassa meg ha a kutya célja, hogy minél több fánál járjon. Az algoritmus lépésszáma legyen $O(n^2 f + n f^2)$, ahol f a parkban levő fák számát jelöli.
12. Legyen $G = (V, E)$ mátrixszal adott n pontú, súlyozott élű irányított gráf! Tegyük fel, hogy G nem tartalmaz negatív összhosszúságú irányított kört, továbbá azt, hogy a G-beli egyszerű irányított utak legfeljebb 25 élből állnak. Javasoljunk $O(n^2)$ költségű módszert az $1 \in V$ pontból az összes további $v \in V$ pontokba vivő legrövidebb utak hosszának a meghatározására!
13. Adott éllistával egy n pontú, e élű G összefüggő irányítatlan gráf. Adjunk $O(e)$ költségű algoritmust olyan $X \subset V(G)$ központi ponthalmaz keresésére, melyre $|X| \leq n/2$ teljesül! Az $X \subseteq V(G)$ egy *központi ponthalmaz*, ha G minden pontja vagy X-beli, vagy egyetlen éllel elérhető valamelyik X-beli pontból.

5. Kupacok

1. **[PZH: 2008. május 9.]** A 10 elemű A tömb első 8 elemére legyen $A[i] = 2i (1 \leq i \leq 8)$, és tekintsük ezt, mint egy 8 elemű kupacot. Rajzolja le az ehhez tartozó fát! Hajtsa végre rajta a BESZÚR(3), BESZÚR(1), MINTÖR műveletsort, rajzolja le az egyes műveletek után a kupacot (és jelezze a közben szükséges részlépéseket is)!
2. **[ZH: 2008. március 28.]** Egy orvosi rendelőben a regisztrációnál kell bejelentkezni, ahol az ott dolgozók eldöntik, hogy a beteg az épp rendelő két orvos közül A-hoz vagy B-hez kell kerüljön, vagy bármelyikükhöz kerülhet. Ezen kívül, a beutaló ismeretében, a beteghez egy, a sürgősséget kifejező, számot is rendelnek. Amikor valamelyik orvos végzett egy beteggel, akkor azon betegek közül, akiket nem csak a másik orvos láthat el behívja a legnagyobb sürgősségi számút. Tegyük fel, hogy a kiosztott sürgősségi számok egymástól különbözőek. Írjon le egy olyan adatszerkezetet, ami abban az esetben ha n beteg várakozik, akkor a regisztráción az új beteg beillesztését, illetve az orvosoknak a következő beteg kiválasztását $O(\log n)$ lépésben lehetővé teszi.

---

3. **[Vizsga: 2008. május 27.]** Egy kupacba beraktunk egy új x elemet, majd végrehajtottunk egy MINTÖR műveletet. Mikor fordul elő, hogy végül az eredeti kupacot kapjuk vissza?
4. Adjunk hatékony (*hehe :)*) algoritmust egy kupac tizedik legkisebb elemének meghatározására!
5. Adott egy n elemet tartalmazó kupac és egy k kulcs. Keressük meg a kupac k-nál kisebb elemeit! Ha m ilyen elem van, akkor az algoritmus $O(m)$ elemi lépést használhat.
6. Adjunk konstans szorzó erejéig optimális költségű algoritmust az alábbi problémára!
   INPUT: Egy $A[1:n]$ tömb, amely eredetileg az $1, \ldots, n$ számokat tartalmazta kupacba rendezve, de öt elem megsérült, és a helyére $*$ került.
   FELADAT: Találjuk meg a tömb összes olyan kitöltését, ami lehetett az eredeti!
7. Tervezzünk olyan adatszerkezetet, ami egy rendezett halmaz elemeinek tárolására szolgál. A megvalósítandó műveletek: *Felépít*(n): n elemből felépíti a struktúrát; *Mintör*, *Maxtör*: a min. illetve max. elem törlése; *Beszúr*(x): az x elemet a struktúrába illeszti. Az egyes műveletek költsége ne legyen több, mint *Felépít*: $O(n)$; *Mintör, Maxtör, Beszúr*: $O(\log n)$, ahol n a tárolt elemek száma.
8. Igazoljuk, hogy egy n elemből álló kupac felépítése $\Omega(n)$ összehasonlítást igényel!

6. Keresés, rendezés

1. Legalább hány összehasonlítás kell ahhoz, hogy egy n elemű tömbből egy olyan tagot találjunk meg, ami a tömb 10 legkisebb eleme közé tartozik?
2. **[ZH: 2004. április 8.]** Az $A[1:n]$ tömbben levő elemekről tudjuk, hogy $A[1] \neq A[n]$. Adjon $O(\log n)$ összehasonlítást használó algoritmust, amely talál egy olyan i indexet, hogy $A[i] \neq A[i+1]$!
3. **[ZH: 2004. március 29.]** Az $A[1 \ldots n]$ tömbben egész számokat tárolunk, ugyanaz a szám többször is szerepelhet. Határozzuk meg $O(n \log n)$ lépésben az összes olyan számot, amelyik egynél többször fordul elő a tömbben.
4. **[Vizsga: 2007. június 19.]** Adott a síkon n pont, melyek koordinátái $(a_1, b_1) \ldots (a_n, b_n)$. Olyan $P = (x, y)$ pontot keresünk a síkon, amire az alábbi összeg minimális.

$$\sum_{i=1}^{n} (|a_i - x| + |b_i - y|)$$

   Adjon algoritmust, ami $O(n \log n)$ lépésben meghatároz egy ilyen P pontot.
5. Vázoljunk egy $O(n)$ időigényű algoritmust (az időkorlát bizonyításával együtt) n olyan egész számból álló sorozat rendezésére, melynek elemei az
   (a) $\{1, \ldots, 3n\}$ tartományba esnek!
   (b) $\{1, \ldots, n^7 - 1\}$ tartományba esnek!

---

6. **[ZH: 2007. április 27.]** A valós számokból álló $a_1^2 \dots a_n^2$ sorozat egy darabig nő, utána csökken. Adjon $O(n)$ összehasonlítást használó algoritmust, ami rendezi az $a_1, \dots a_n$ sorozat!
7. Az $A[1:n]$ tömbben egy rendezett univerzum n különböző eleme volt, nagyság szerint növekvő sorrendben. Valaki időközben megkeverte a tömb elemeit, de csak annyira, hogy minden egyes elem új helye az eredetitől legfeljebb 5 távolságra esik. Adjunk $O(n)$ futásidejű algoritmust az eredeti állapot helyreállítására!
8. **[ZH: 2010. április 19.]** Az A tömbben n különböző számot tárolunk. Tudjuk, hogy $A[1] > A[2]$ és $A[n-1] < A[n]$. Adjon algoritmust, mely $O(\log n)$ összehasonlítással megtalál a tömbben egy lokális minimumot (ha van), azaz egy olyan $1 \leq i \leq n$ indexet, hogy $A[i]$ tömbbeli szomszédai nagyobbak, mint $A[i]$.
9. A (növekvően) rendezett $A[1:n]$ tömb k elemét valaki megváltoztatta. A változtatások helyeit nem ismerjük. Javasoljunk $O(n + k \log k)$ költségű algoritmust az így módosított tömb rendezésére!
10. Legyen adott egy egészekből álló $A[1:n]$ tömb valamint egy b egész szám. Szeretnénk hatékonyan eldönteni, hogy van-e két olyan $i, j \in \{1, \dots, n\}$ index, melyekre $A[i] + A[j] = b$. Oldjuk meg ezt a feladatot $O(n \log n)$ időben!
11. **[Vizsga: 2009. június 11.]** Adott a számegyenesen n intervallum, $[a_1, b_1], \dots, [a_n, b_n]$. Azt akarjuk tudni, hogy együtt milyen hosszú részt fednek le a számegyenesből (azaz, hogy mennyi az $\cup_{i=1}^{n}[a_i, b_i]$ összhossza). Adjon $O(n \log n)$ lépéses algoritmust ennek a hossznak a meghatározására!
12. Adottak a sík egész koordinátájú $P_1 = (x_1, y_1), \dots, P_n = (x_n, y_n)$ koordinátájú pontjai. Javasoljunk $O(n)$ költségű módszert olyan $P_i \neq P_j$ pontok kiválasztására, amelyeken átmenő egyenes által meghatározott félsíkok közül az egyik tartalmazza az összes pontot!
13. **[Vizsga: 2004. június 10.]** Az n méretű (nem feltétlenül rendezett) A tömb elemei különböző pozitív egész számok. Adjon algoritmust, amely meghatároz egy $1 \leq k \leq n$ számot és kiválaszt k különböző elemet az A tömbből úgy, hogy a kiválasztott elemek összege nem több, mint k^3. Ha nincs ilyen k, akkor az algoritmus jelezze ezt a tényt! Az algoritmus lépésszáma legyen $O(n \log n)$! (Két szám összehasonlítása, összadása vagy szorzása egy lépésnek számít.)
14. **[Vizsga: 2004. június 3.]** A $2^k - 1$ elemű A tömb elemei mind különbözőek és növekvő sorrendben vannak. Minden elemet egy k hosszú bitsorozat ír le, tehát tekinthetjük úgy, hogy a $0, 1, 2, \dots, 2^k - 1$ számokat tároljuk egy kivételével. A feladat ennek a hiányzó számnak a megkeresése. Ehhez egy lépésben valamelyik elem egy bitjére kérdezhetünk rá: a $BIT(i, j)$ eljárás az $A[i]$ elem j-edik bitjét mondja meg. Adjon olyan algoritmust, amely a BIT eljárás $O(k)$-szori hívásával megtalálja a hiányzó számot (bitsorozatot).
15. Adott egy dobozban n különböző méretű anyacsavar, valamint egy másik dobozban a hozzájuk illő apacsavarok. Kizárólag a következő összehasonlítási lehetőségünk van: egy apacsavarhoz hozzápróbálunk egy anyacsavart. A próbának háromféle kimenete lehet: apa<anya, apa=anya, apa>anya, annak megfelelően, hogy az apacsavar külső átmérője hogyan viszonyul az anyacsavar belső átmérőjéhez. Szeretnénk minden anyacsavarhoz megtalálni a megfelelő apacsavart. Adjunk erre a feladatra *átlagosan* $O(n \log n)$ összehasonlítást felhasználó módszert!
16. A 4 elemű I abc felett adott két szó: $x = x_1 x_2 \dots x_n$ és $y = y_1 y_2 \dots y_k$, ahol $1 \leq k \leq n$ és $\forall i, j : x_i, y_j \in I$. Keressük az x szóban az olyan részszavakat, amelyek anagrammái y-nak, azaz az olyan i indexeket, hogy az $x_i, x_{i+1}, \dots, x_{i+k-1}$ betűk megfelelő sorrendbe rakva az y szót adják. Adjunk algoritmust, ami x-ben az összes ilyen i helyet $O(n)$ lépésben meghatározza!
17. **[Vizsga: 2003. május 30.]** Adott összesen $2n$ különböző szám két n elemű halmazban, $A = \{a_1, \dots, a_n\}$ és $B = \{b_1, \dots, b_n\}$. Azt szeretnénk eldönteni minimális számú összehasonlításssal, hogy a $2n$ szám közül a legnagyobb az A vagy a B halmazban van-e. (Azaz nem kell feltétlenül meghatározni, melyik elem a legnagyobb, csak azt, hogy melyik halmazba tartozik.) Mutassa meg, hogy ehhez a feladathoz is legalább annyi összehasonlítás kell, amennyi $2n$ elem közül a maximális meghatározásához szükséges.
18. **[ZH: 2002. április 8.]** Adottak a $c_1, c_2, \dots, c_n$ különböző egész számok. Ezeket szeretnénk nagyság szerint rendezni növekvő, vagy csökkenő sorrendbe úgy, hogy a szokásos összehasonlítás helyett, most a következő kérdéseket lehet feltenni: *Három kiválasztott elem közül melyik a középső?* Bizonyítsuk be, hogy a leghatékonyabb algoritmus $\Theta(n \log_2 n)$ összehasonlítást használ!

7. Keresőfák, p-f fák, 2-3 fák

1. **[Vizsga: 2007. június 5.]** Adott egy n csúcsú és egy k csúcsú piros-fekete fa. A két fában tárolt összes elemből $O(n+k)$ lépésben készítsen rendezett tömböt.
2. **[ZH: 2004. március 29.]** Egy bináris keresőfában csupa különböző egész számot tárolunk. Lehetséges-e, hogy egy $KERES(x)$ hívás során a keresési út mentén a $20, 18, 3, 15, 5, 8, 9$ kulcsokat látjuk ebben a sorrendben? Ha nem lehetséges, indokolja meg miért nem, ha pedig lehetséges, határozza meg az összes olyan x egész számot, amire ez megtörténhet.
3. **[ZH: 2009. április 24.]** Egy bináris fa csúcsai 0 és 9 közötti egész számokkal vannak megcímkézve. Az inorder bejárás során a címkék sorrendje: 9, 3, 1, 0, 4, 2, 7, 6, 8, 5, a postorder bejárásnál pedig 9, 1, 4, 0, 3, x, 7, 5, y, 2. Mi lehet az x és mi az y?
4. **[ZH: 2009. április 24.]** Egy 2-3 fa gyökerének három fia van, a benne szereplő két érték 40 és 50. Mennyi lehet a tárolt elemek minimális, illetve maximális száma, ha tudjuk, hogy csak pozitív egész számokat tárol a fa?
5. **[Vizsga: 2009. június 17.]** Az MSc-re jelentkezőknek a felvételit alkotó 3 témakör mindegyikéből lesz egy írásbeli pontszámuk (P_1, P_2, P_3), és keletkezik egy felvélteli pontszámuk is (FP). Tegyük fel, hogy a P_i-k 1 és 30 közötti egészek, míg az FP tetszőleges pozitív egész szám lehet. Adjon meg egy olyan adatszerkezetet, amivel a következő műveletek az adott időben végrehajthatóak (n a jelentkezők számát jelöli)!
   BESZÚR(P_1, P_2, P_3, FP): az adott pontszámok beillesztése – $O(\log n)$
   KERES(p): a pontosan p felvételi ponttal ($FP = p$) rendelkező jelentkezők számát határozza meg – $O(\log n)$
   KORLÁT(i, q): az írásbelin az i-edik témakörből legalább q pontot elért jelentkezők számát határozza meg – $O(1)$
6. Az $[1, 178]$ intervallum összes egészei egy 2-3 fában helyezkednek el. Tudjuk, hogy a gyökérben két kulcs van, és ezek közül az első 17. Mi lehet a második? Miért?

7.
   (a) Lehet-e tetszőleges (adott) kulcshalmaz esetén olyan piros-fekete fát építeni, hogy az azonos szinten lévő elemek azonos színűek legyenek?
   (b) Van-e olyan piros-fekete fa, ami nem így néz ki?
8. **[ZH: 2007. április 27.]** Egy piros-fekete fában lehetséges-e, hogy a piros-fekete tulajdonság megsértése nélkül
   (a) néhány fekete csúcsot átváltoztathatunk pirosra?
   (b) valamelyik (csak egy) fekete csúcsot átváltoztathatjuk pirosra?
   (Mást nem változtatunk a fán.)
9. Az S_1 és S_2 kulcshalmazokat kiegészített 2-3 fákban tároljuk. Ezek az eredeti 2-3 fától annyiban különböznek csak, hogy minden csúcsban nyilván van tartva az onnan induló részfa magassága. Tudjuk továbbá, hogy az S_1-beli kulcsok mind kisebbek, mint az S_2-beliek. Javasoljunk hatékony algoritmust a két fa egyesítésére!
10. **[ZH: 2009. június 4.]** Mutassa meg, hogyan kell a 2-3 fa BESZÚR eljárását módosítani, ha a fa minden v csúcsában a szokásos dolgokon kívül azt is nyilvántartjuk, hogy hány levél van a v gyökerű részfában!
11. **[PZH: 2008. május 9.]** Vázolja a 2-3 fának (és műveleteinek) egy olyan módosítását, amiben továbbra is van KERES, BESZÚR, TÖRÖL, MIN, MAX művelet, és ezeken kívül van még RANG és K-ADIK művelet is, ahol RANG(x) azt adja vissza, hogy a tárolt elemek között az x a rendezés szerint hányadik elem, a K-ADIK(i) pedig, hogy a rendezés szerint a tárolt elemek közül melyik az i-edik. A módosítás során a felsorolt szokásos műveletek lépésszámának nagyságrendje ne változzon, és mindkét új művelet lépésszáma legyen $O(\log n)$, ahol n a tárolt elemek száma.

12. **[Vizsga: 2003. március 31.]** Tervezzen adatstruktúrát a következő feltételekkel. Természetes számokat kell tárolni, egy szám többször is szerepelhet. A szükséges műveletek:
BESZÚR(i): i egy újabb példányát tároljuk
TÖRÖL(i): i egy példányát töröljük
MINDTÖRÖL(i): i összes példányát töröljük
DARAB(i): visszaadja, hogy hány példány van i-ből
ELEM(K): megmondja, a nagyság szerinti rendezésben a K-adik elem értékét.
Az adatstruktúra legyen olyan, hogy ha m-*féle* elemet tárolunk, akkor mindegyik művelet lépésigénye $O(\log m)$.
(Például ha a tárolt elemek $1, 1, 3, 3, 3, 8$, akkor DARAB$(1) = 2$, ELEM$(4) = 3$ és $m = 3$.)
13. **[ZH: 2011. április 19.]** Adott $2^k - 1$ különböző szám, mindegyik az $\{1, 2, \ldots, n\}$ halmazból, ezekből kell egy $O(k)$ mélységű bináris keresőfát készíteni. Adjon olyan algoritmust, amely ezt $O(n)$ lépésben megcsinálja!
14. **[Vizsga: 2009. május 28.]** Adott egy n csúcsú bináris keresőfa. Ennek minden v csúcsára meg akarjuk határozni, hogy a v gyökerű részfában hány darab v-nél kisebb elem van tárolva. Adjon algoritmust, ami ezt a feladatot $O(n)$ lépésben megoldja!
15. **[ZH: 2009. április 24.]** Egy piros-feket fában jelölje x és y a gyökér két fiát. Tudjuk, hogy $fm(x) = fm(y)$, de az x csúcs két gyerekének különbözik a fekete magassága. Milyen színű lehet az y csúcs?
16. **[ZH: 2003. március 31.]** Egy 2-3 fába egymás után 1000 új elemet illesztettünk be. Mutassa meg, hogy ha ennek során egyszer sem kellett csúcsot szétvágni, akkor a beillesztések sorozata előtt már legalább 2000 elemet tároltunk a fában.
17. Egy bináris keresőfa csúcsait egy, a gyökértől egy levélig menő út szerint három osztályba soroljuk: B az úttól balra levő, U az útra eső, J pedig az úttól jobbra levő csúcsok halmazát jelöli. Igaz-e mindig, hogy minden B-beli csúcs kulcsa kisebb tetszőleges U-beli csúcs kulcsánál, és minden U-belis csúcs kulcsa kisebb tetszőleges J-beli csúcs kulcsánál?
18. Adott n pont a síkon, melyek páronként mindkét koordinátájukban különböznek. Bizonyítsuk be, hogy pontosan egy bináris fa létezik, melynek csúcsai az adott n pont, és az első koordináta szerint a keresőfa tulajdonsággal, a második szerint a kupac tulajdonsággal rendelkezik! (A kupac tulajdonságba most nem értjük bele, hogy a fa teljes bináris fa legyen.)
19. **[ZH: 2004. március 29.]** Egy kezdetben üres 2-3 fába az $1, 2, \ldots, n$ számokat szúrtuk be ebben a sorrendben. Bizonyítsa be, hogy a keletkezett fában a harmadfokú csúcsok száma $O(\log n)$.
20. Adott egy $n = 2^k - 1$ pontú teljes bináris keresőfa. A fában tárolt elemek egészek az $I = [1, 2^k]$ intervallumból, és egy szám legfeljebb egyszer fordul elő a fában. Utóbbi feltétel szerint pontosan egy olyan i egész szám van 1 és 2^k között, amely nincs a fában. Adjunk $O(\log n)$ lépésszámú algoritmust i meghatározására!
21. Egy fában az x csúcs *súlya* legyen x leszármazottainak száma. Egy bináris fát szigorúan binárisnak mondunk, ha a levelek kivételével minden csúcsnak pontosan 2 fia van. Tegyük fel, hogy egy szigorúan bináris fa minden x csúcsára fennáll, hogy

$$\frac{1}{2} < \frac{súly(bal(x))}{súly(jobb(x))} < 2.$$

Bizonyítsuk be, hogy ez csakis egy teljes fa lehet, azaz ha k szintje van, akkor a csúcsok száma $2^k - 1$.
(Ez nem kifejezetten keresőfázós feladat, de úgy általában érdekes.)

8. Hash

1. Nyitott címzéssel hash-eltünk egy 11 elemű táblába a $h(k) = k \pmod{11}$ hash-függvény segítségével. A következő kulcsok érkeztek (a megadott sorrendben): $10, 22, 31, 4, 15, 28, 17, 88, 59$. Adjuk meg a tábla végső állapotát a következő két próbamódszerre:
   (a) lineáris próba;
   (b) kvadratikus maradék próba!

2. **[Vizsga: 2010. január 21.]** Kettős hashelést használva szúrja be egy kezdetben üres, $M = 11$ méretű táblába a következő kulcsokat (ebben a sorrendben): 26, 3, 48, 14, 15, 7. A használt hash függvény legyen

$$h(k) = (k \pmod M),$$

a próbasorozat hash függvénye pedig

$$h'(k) = 1 + (k \pmod{M-5}).$$

Minden beszúrás után rajzolja le a tábla pillanatnyi állapotát!

3. **[ZH: 2010. április 19.]** Egy M méretű hash-táblába $n < M$ elemet raktunk be nyitott címzéssel, kvadratikus próbával, a $h(x)$ hash-függvényt használva. Ennek során t_1 ütközés történt (ennyiszer kellett tovább próbálkoznunk, egy elem beszúrása során több ütközés is lehetett). Ugyanezt az n elemet ugyanabban a sorrendben beszúrtuk egy M^2 méretű hash-táblába is, de most lineáris próbával, $M \cdot h(x) + 1$ hash-függvénnyel, ekkor t_2 ütközés történt. Igazolja, hogy $t_2 \leq t_1$.

---

4. **[Vizsga: 2011. június 9.]** Az alábbi hash-táblát az üresből kiindulva beszúrások sorozatával kaptuk. Határozzuk meg a beszúrások összes lehetséges sorrendjét, ha a hash-függvény a $h(x) = 3x \pmod{10}$ volt és a nyitott címzésű hash-elést lineáris próbával alkalmaztuk!

0	1	2	3	4	5	6	7	8	9
					5	19	3	33	23

5. A $T[0 : M-1]$ táblában rekordokat tárolunk nyitott címzésű hashelt szervezéssel. Az ütközések feloldására lineáris próbát alkalmazunk. Tegyük fel, hogy a tábla használata során egy hibás törlés történt: egy cellából kitöröltünk egy rekordot a törlés-bit beállítása nélkül.
   (a) Igaz-e, hogy a hibás törlés helye mindig megtalálható?
   (b) Adjunk hatékony (lineáris) algoritmust a tábla megjavítására! (Módosítsuk a táblát úgy, hogy megszűnjenek a hibás törlés negatív következményei!)

6. A $T[0 : M-1]$ táblában $2n$ elemet helyeztünk el az első $3n$ helyen egy ismeretlen hash-függvény segítségével. A táblában minden $3i$ indexű hely üresen maradt ($0 \leq i \leq n$). Legfeljebb hány ütközés lehetett, ha az ütközések feloldására
   (a) lineáris próbát használtunk?
   (b) kvadratikus próbát használtunk?

7. **[ZH: 2005. április 8.]** Egy m méretű hash-táblában már van néhány elem. Adjon $O(m)$ lépésszámú algoritmust, amely meghatározza, hogy egy újabb elem lineáris próbával történő beszúrásakor maximum hány ütközés történhet.

8. Mi a baja a $h(k) = k^2 \pmod 7$ hash-függvénynek, ha a tábla 7 méretű?

9. Mutassuk meg, hogy nyitott címzéses hashelés és lineáris próba esetén már két kulcshoz tartozó hash-függvényérték megegyezése is okozhat tetszőlegesen nagy méretű csomósodást!

10. **[Vizsga: 2008. június 3.]** Az 1 és 91 közötti összes 3-mal osztható egész számot valamilyen sorrendben egy M méretű hash-táblába raktuk a $h(x) = x \pmod M$ hash-függvény segítségével, lineáris próbával. Ennek során hány ütközés fordulhatott elő, ha $M = 35$, illetve ha $M = 36$?

11. **[Vizsga: 2005. május 26.]** A kezdetben üres M méretű hash-táblába sorban beraktuk a $k_1, k_2, \ldots, k_n$ kulcsokat a $h(x) \equiv x \pmod M$ hash-függvénnyel, lineáris próbával. Jelölje t_1 a keletkezett táblában az egymás melletti foglalt mezők maximális számát. Amikor ugyanezt a $k_1, k_2, \ldots, k_n$ sorozatot ugyanabban a sorrendben egy üres $2M$ méretű táblába rakjuk be a $h(x) \equiv x \pmod{2M}$ hash-függvénnyel, lineáris próbával, akkor a kapott táblálban legyen t_2 az egymás melletti foglalt mezők maximális száma.
   (a) Igazolja, hogy $t_2 \leq t_1$
   (b) Igaz-e, hogy $t_1 \leq 2t_2$?

9. Mélységi bejárás, DAG, valamint alkalmazásaik

1. A 6 pontú irányított G gráf csúcsait jelölje x, y, z, u, v, w. A gráf egy mélységi bejárásánál a mélységi, ill. a befejezési számok a következők: $x : 1, 6;\ y : 2, 4;\ z : 6, 5;\ u : 3, 3;\ v : 4, 1;\ w : 5, 2$. Adjuk meg a bejáráshoz tartozó mélységi feszítőfa éleit! Rekonstruálható-e G az előző számok ismeretében?
2. **[ZH: 2008. március 28.]** Egy $n \times n$ méretű táblázat minden eleme egy egész szám. A táblázat bal alsó sarkából akarunk eljutni a jobb felső sarkába úgy, hogy egy lépésben a táblázatban vagy felfelé vagy jobbra egyet lépünk. Azt szeretnénk, hogy a lépegetés során látott elemek növekvő sorrendben kövessék egymást. Egy ilyen út értéke a benne szereplő számok összege. Adjon $O(n^2)$ futási idejű algoritmust, ami meghatározza, hogy az adott táblázatban a szabályok szerinti utak értékei között mekkora a legnagyobb! *(Persze, dinprog is kézenfekvő, de most DAG-gal!)*
3. **[ZH: 2011. március 28.]** Van b darab borítékunk, az i-ediknek a hossza h_i, a magassága m_i. Az i-edik borítékba akkor tudjuk berakni a j-edik borítékot, ha $h_j < h_i$ és $m_j < m_i$ is teljesül (nem forgatjuk és nem is hajtogatjuk a borítékokat). Célunk, hogy minél hosszabb olyan láncot alakítsunk ki, hogy az i-edikben benne van a j-edik, abban a k-adik, stb.
Legyen adott egy $L > 0$ egész és a h_i és m_i számok. Hogyan lehet $O(b^2)$ lépésben eldönteni, hogy kialakítható-e a borítékokból egy L hosszú lánc?

---

4. **[Vizsga: 2008. május 27.]** Éllistával adott egy n pontú e élű irányított gráf. Azt szeretnénk tudni, hogy van-e benne olyan minden pontot tartalmazó részgráf, ami egy, a gyökerétől a levelek felé irányított fa. Adjon $O(ne + n^2)$ lépésszámú algoritmust, ami ha van, talál egy ilyen részgráfot.
5. **[ZH: 2007. április 27.]** Az $n \times n$ méretű tábla minden mezőjére egy pozitív egész szám van írva, az i-edik sorának j-edik elemére $A[i, j]$, ahol $0 \leq i, j < n$. Feladat, hogy az első oszlopból eljussunk az utolsó oszlopba úgy, hogy egy lépésben mindig a következő oszlopba lépünk, és azon belül, ha az i-edik sorban voltunk, akkor a következő lépésben vagy az $(i-1) \pmod n$, vagy az i, vagy az $(i+1) \pmod n$ számú sorba kerülhetünk. Adjon $O(n^2)$ lépésszámú algoritmust, ami meghatározza, hogy az első oszlop melyik eleméből induljunk, ha azt akarjuk, hogy a bejárt mezőkön lévő számok összege minimális legyen (az utolsó oszlop bármelyik mezője lehet az utolsó olyan mező, amire rálépünk).
6. **[Vizsga: 2010. június 3.]** Egy falutörténet írója n korábbi lakosról gyűjtött információkat. A kérdésekre kapott válaszok a következő típusúak voltak:
   - S_i személy meghalt S_j születése előtt;
   - S_i személy élete során született S_j;
   - S_i személy korábban született, mint S_j;
   - S_i korábban halt meg, mint S_j.

   Egy S_i, S_j párra nem biztos, hogy szerepel minden választípus, és olyan pár is lehet, amely egyetlen válaszban sem szerepel együtt. Mivel az emberek időnként rosszul emlékeznek, nem biztos, hogy minden kapott információ helyes. Adjon algoritmust, amivel k db fenti típusú válaszról $O(n + k)$ lépésben eldönthető, hogy van-e közöttük ellentmondás.
7. **[ZH: 2005. április 8.]** Cirkuszi akrobaták egymás vállára állva minél nagyobb tornyot szeretnének létrehozni (a toronyban minden szinten csak egy akrobata lesz). Esztétikai és gyakorlati szempontok miatt egy ember vállára csak egy olyan állhat, aki nála alacsonyabb és könnyebb is. A cirkuszban n akrobata van, adott mindegyikük magassága és súlya. Adjon algoritmust, amely $O(n^2)$ lépésben megadja a lehetséges legtöbb emberből álló torony összeállítását.
8. **[ZH: 2007. április 27.]** Tekintsük az olyan G irányított gráfokat, amelyekben ha eltekintünk az élek irányításától, akkor a kapott irányítatlan G' gráf összefüggő. A G gráf egy mélységi bejárásánál maximálisan hány olyan csúcs lehet, amelyre a mélységi és a befejezési szám megegyezik?
9. Adjunk algoritmust, mely egy éllistával megadott irányítatlan gráfban vagy talál egy kört, vagy igazolja a gráf körmentességét $O(|V|)$ időben (függetlenül attól, hogy $|E|$ akár sokkal nagyobb is lehet, mint $|V|$)!
10. **[Vizsga: 2007. június 12.]** Egy számítógéphálózatban n számítógép van. Minden olyan eseményt, hogy az i-edik gép üzenetet küld a j-ediknek (i, j, t) formában feljegyezünk, ahol a t egész szám az üzenet küldésének időpontját jelöli. Ugyanabban a t időpontban egy gép több gépnek is küldhet üzenetet. Ha a t időpontban az i-edik gép vírusos volt, akkor egy (i, j, t) üzenet hatására a j-edik gép mefertőződhet, ami azt jelenti, hogy a $t + 1$ időponttól kezdve már a j-edik gép is vírusos lehet.

Legyen adott az (i, j, t) hármasoknak egy m hosszú listája, valamint x, y és $t_0 < t_1$ egész számok. Azt kell eldöntenünk, hogy ha az x-edik gép a t_0 időpontban vírusos volt, akkor lehet-e emiatt az y-adik gép a t_1 időpontban vírusos. Adjon algoritmust, ami ezt a kérdést $O((t_1 - t_0)n + m)$ lépés után megválaszolja.

11. **[Vizsga: 2007. június 19.]** Egy előre rögzített útvonalon úgy indulunk el, hogy az autó L literes tankja tele van. Úticélunkhoz úgy akarunk eljutni, hogy legalább egy fél tanknyi benzin maradjon az autóban. Tudjuk, hogy az utunkba eső n benzinkút közül melyikben mennyibe kerül a benzin, továbbá, hogy két szomszédos benzinkút között, valamint a kiindulóponttól az első benzinkútig, illetve az utolsó benzinkúttól a célunkig mennyi benzint fogyaszt az autó. Az egyszerűség kedvéért ha megállunk egy benzinkútnál, akkor mindig tele tankolunk. Adjon algoritmust, ami $O(Ln^2)$ lépésben megmondja, hogy hol álljunk meg tankolni ha azt akarjuk, hogy utunk során a benzinköltség minimális legyen. *(Kiegészítés: feltesszük, hogy L egész, továbbá a fogyasztás mindig egész liter.*

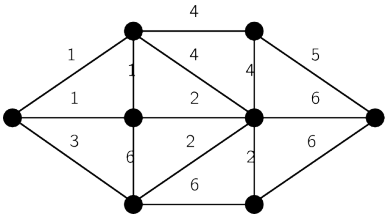
12. A $G(V, E)$ összefüggő, irányított gráf minden éle az $1, 2, \ldots k$ számok valmelyikével van súlyozva. Egy út értéke legyen az úton található élek súlyainak maximuma. Adjunk $O(|E| \log k)$ futásidejű algoritmust az adott $x, y \in V$ csúcsok közti legkisebb súlyú út értékének meghatározására!

13. **[Vizsga: 2003. május 30.]** Éllistával adott egy G gráf, melynek n csúcsa és e éle van. A gráf minden csúcsához hozzá van rendelve egy 1 és k közötti egész szám (címke). Találjunk (ha létezik) olyan *tarka* utat a gráfban, amelyben minden $1 \leq i \leq k$ címke pontosan egyszer fordul elő. Az algoritmus lépésszáma legyen $O(k!(e + n))$.

14. Bizonyítsuk be, hogy minden $G = (V, E)$ irányított gráf felbontható két DAG-ra; pontosabban az élhalmazának van olyan E_1, E_2 partíciója ($E = E_1 \cup E_2$ és $E_1 \cap E_2 = \emptyset$), hogy a $G_1 = (V, E_1)$ és a $G_2 = (V, E_2)$ gráfok DAG-ok!

10. Minimális költségű feszítőfák

1. Mennyi az alábbi gráfban a minimális feszítőfa súlya? (Gyakorlásképpen mindhárom módszerrel – Prim, Kruskal, Borůvka – csináljuk meg a feszítőfa-keresést!)



2. Egy teljes gráf ponthalmaza $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_l$. Az (x_i, x_j) élek költsége (súlya) 1, az (y_i, y_j) éleké 2, az (x_i, y_j) éleké 3. Mennyibe kerül a legolcsóbb feszítőfa?

3. **[Vizsga: 2005. június 23.]** Irányítatlan gráf tárolására adjon meg egy adatszerkezetet az alábbi műveletekkel:
ÚJCSÚCS(v): a gráfhoz hozzáad egy új csúcsot;
ÚJÉL(u, v): a már létező u és v csúcsok közé felvesz egy élet;
VANÚT(u, v): *igen* értéket ad vissza, ha vezet az u és v csúcsok között út, egyébként pedig *nem* értéket.
Ha a tárolt gráfnak n csúcsa van, akkor mindhárom művelet lépésszáma legyen $O(\log n)$.

4. **[Vizsga: 2010. május 27.]** Adott egy $G = (V, E)$ irányítatlan, összefüggő, súlyozott gráf az éllistájával valamint egy $f \in E$ él. Tegyük fel, hogy a gráfban minden él súlya különböző. Adjon $O(|V| + |E|)$ lépésszámú algoritmust annak eldöntésére, hogy van-e olyan minimális feszítőfa G-ben, amely tartalmazza az f élet!

5. **[Vizsga: 2008. június 3.]** Éllistával adott a $G = (V, E)$ egyszerű, összefüggő gráf. A gráf élei súlyozottak, a súlyfüggvény $c : E \to \{-1, 1\}$. Adjon algoritmust, ami G-ben $O(|V| + |E|)$ lépésben meghatározza, hogy mennyi a minimális súlya egy olyan részgráfnak, ami G minden pontját tartalmazza és összefüggő.
6. **[Vizsga: 2007. június 5.]** Útépítéskor a környéken sok helyen felszedték a járdát. Az építők 1-től n-ig megszámozták a fontos pontokat (kapualj, útkereszteződés, stb.). A környék állapotát két $n \times n$ táblázat írja le. A J táblázatban $J[i, j] = 1$, ha az i és j pontok az utcán szomszédosak és megmaradt az ezeket összekötő részen a járda, egyébként az érték 0. A P táblázat az ideiglenesen elhelyezhető pallókat írja le: ha az i és j pontok összeköthetőek egy pallóval, akkor $P[i, j]$ ennek a pallónak a költsége. Amennyiben a két pont nem köthető össze egy pallóval, akkor a táblázatban $*$ szerepel. (Minden palló pontosan két pontot érint.) Szeretnénk biztosítani, hogy mindenhonnan mindenhova el tudjunk jutni (hol járdán hol pallón haladva). Az építők célja, hogy úgy válasszák meg a pallók helyét, hogy minél kevesebb pallót kelljen használniuk, és ezen belül a pallók értékeinek összege minimális legyen. Írjon le egy algoritmust, ami $O(n^2)$ lépésben javasol egy ilyen elhelyezést. (Egy pontra tetszőlegesen sok palló illeszkedhet, és a gyalogosok az egy pontra illeszkedő pallók bármelyikéről bármelyikére át tudnak lépni.)
7. Mátrixával adott egy G irányítatlan súlyozott gráf. Adott még a G-nek egy F minimális súlyú feszítőfája, és az F-nek egy f éle. Adjon $O(n^2)$ lépésszámú algoritmust, ami meghatározza, hogy az f él súlyát meddig lehet úgy felemelni, hogy az F a gráf minimális feszítőfája maradjon.
8. **[Vizsga: 2008. május 27.]** Éllistával adott egy egyszerű, összefüggő, súlyozott irányítatlan $G = (V, E)$ gráf amiben nincs két egyforma súlyú él. Adjon $O(|V| \cdot |E|)$ lépésszámú algoritmust, ami megadja a gráfban a második legkisebb súlyú feszítőfát.
9. Legyen adva egy (egyszerű, irányítatlan, összefüggő) n csúcsú G gráf éllistával, az élek súlyozásával együtt. Tegyük fel, hogy a G-ből a v_1 csúcs, valamint a v_1-re illeszkedő élek elhagyásával keletkező G' gráf még mindig összefüggő, és adott a G' egy minimális költségű feszítőfája. Adjunk $O(n \log n)$ futási idejű algoritmust a G gráf egy minimális költségű feszítőfájának elkészítésére!
10. Hány éle van az n pontú egyszerű összefüggő gráfnak, ha pontosan 3 különböző feszítőfája van?
11. Éllistával adott egy összefüggő, egyszerű, irányítatlan, n csúcsú, e élű G gráf csupa különböző élsúllyal. Adjunk egy olyan $O(e)$ költségű algoritmust, ami a G gráf egy minimális feszítőfájának legalább $\frac{2}{3}n$ élét előállítja! (Azaz egy olyan élhalmazt keresünk, ami biztosan része egy minimális költségű feszítőfának.)
12. Bizonyítsuk be, hogy a piros-kék algoritmus akkor is helyesen működik, ha egy feszítőfa költségét az élsúlyok összege helyett az élsúlyok maximumával definiáljuk!

11. Bonyolultságelmélet

1. Lássuk be, hogy az alábbi problémák NP-beliek! Melyekről tudjuk megmondani, hogy coNP-ben vannak? Melyekről, hogy P-ben?
   (a) $\pi_1 = \{(G, k) \mid G$ gráf kiszínezhető k színnel$\}$
   (b) $\pi_2 = \{G \mid G$ irányítatlan gráfban van Euler-kör$\}$
   (c) $\pi_3 = \{(G, k) \mid G$ irányítatlan gráfban van k független pont$\}$
2. Bizonyítsuk be, hogy a következő probléma P-beli:

$$\pi = \left\{ G \;\middle|\; \begin{array}{l} G \text{ gráf kiszínezhető a piros, zöld, sárga, kék színekkel úgy,} \\ \text{hogy pontosan egy csúcs legyen piros és pontosan két csúcs legyen kék} \end{array} \right\}$$

3. Tegyük fel, hogy van egy olyan F eljárásunk, ami egy input G gráfra és k számra 1 lépés alatt megmondja, hogy van-e G-ben legalább k méretű független ponthalmaz. Tervezzünk olyan algoritmust, ami polinomidőben
   (a) meghatározza $\alpha(G)$-t!
   (b) talál egy $\alpha(G)$ méretű független ponthalmazt!

4. Adjunk Karp-redukciót a 3-SZÍN nyelvről a 4-SZÍN nyelvre!
5. **[Vizsga: 2007. május 29.]** A G irányítatlan gráf minden x pontjához tartozik egy $s(x)$ súly. Célunk, hogy olyan feszítőfát találjunk a gráfban, amiben a levelekhez tartozó súlyok összege minimális. Fogalmazza meg a feladathoz tartozó nyelvet és vagy lássa be róla, hogy P-ben van vagy azt, hogy NP-teljes.
6. Bizonyítsuk be a következő problémáról, hogy NP-teljes, vagy azt, hogy P-beli!

$$L = \{G \mid G \text{ irányítatlan teljes gráf, amiben van Hamilton-kör}\}$$

7. **[Vizsga: 2008. június 10.]** Tegyük fel, hogy $P \neq NP$. Az alábbi feltételek közül melyikből következik és melyikből nem következik hogy az X eldöntési probléma nem P-beli?
   (a) Egy NP-teljes Y problémára X Karp-redukálható.
   (b) Egy NP-teljes Y probléma Karp-redukálható X-re.
   (c) az X probléma NP-beli.

---

8. Lássuk be, hogy az alábbi problémák NP-beliek! Melyekről tudjuk megmondani, hogy coNP-ben vannak? Melyekről, hogy P-ben?
   (a) $\pi_1 = \{(G,k) \mid G$ páros gráfban van teljes párosítás$\}$
   (b) $\pi_2 = \{(G,k) \mid G$ páros gráfban van k élből álló párosítás$\}$
   (c) $\pi_3 = \{G \mid G$ irányítatlan gráf, van benne pontosan 100 élből álló kör$\}$
   (d) $\pi_4 = \{(G,k) \mid G$ irányítatlan gráf, van benne legalább k élből álló kör$\}$
   (e) $\pi_5 = \left\{(s_1, s_2, \ldots, s_n, b) \mid \forall i\ s_i, b \in \mathbb{Z}^+; \exists j_1, \ldots, j_k\ (1 \leq k \leq n) : \sum_{l=1}^{k} s_{j_l} = b\right\}$
9. A G irányítatlan gráf minden x pontjához tartozik egy $s(x)$ súly. Célunk, hogy olyan feszítőfát találjunk a gráfban, amiben a levelekhez tartozó súlyok összege minimális. Fogalmazzuk meg a feladathoz tartozó eldöntési problémát, és bizonyítsuk be, hogy NP-beli!
10. Tegyük fel, hogy van egy algoritmusunk, ami polinom időben megmondja, hogy adott G gráf kiszínezhető-e legfeljebb k db színnel! (Vagyis input: G és k; output: igen/nem). Hogy tudnánk ennek segítségével polinom időben meghatározni $\chi(G)$-t?
11. Tegyük fel, hogy van egy algoritmusunk, ami polinom időben megmondja, hogy adott G gráf kiszínezhető-e legfeljebb k db színnel! A fentiek értelmében azt is megtudhatjuk polinom időben, hogy mennyi $\chi(G)$. Hogyan tudnánk kiszínezni polinom időben a gráfot $\chi(G)$ színnel?
12. Legyen a π döntési probléma inputja egy G gráf, az output pedig pontosan akkor „igen”, ha G síkbarajzolható. Mutassuk meg, hogy $\pi \in NP \cap coNP$.
13. Mi az alábbi problémák bonyolultsága, ha az input egy $G(V,E)$ gráf ($|V| = n, |E| = e$)? Természetesen bizonyítsuk is be!
   (a) Van-e G-ben egy legalább 15 pontú teljes részgráf?
   (b) Van-e G-ben legalább $n/100$ hosszúságú kör?
   (c) Van-e G-ben olyan feszítőfa, amelyben a maximális fokszám legfeljebb 2?
   (d) Van-e G-ben olyan feszítőfa, amelyben a maximális fokszám legfeljebb 3?
14. Bizonyítsuk be, hogy a leghosszabb út meghatározása egy irányított, élsúlyozott G gráfban NP-teljes! (Pontosabban az ehhez tartozó eldöntési probléma.) Másképpen: bizonyítsuk be, hogy

$$L = \{(G,k) \mid G \text{ irányított, élsúlyozott gráf, van benne legalább } k \text{ súlyú út}\}$$

   NP-teljes!
15. **[Vizsga: 2007. június 12.]** Tudjuk, hogy a síkgráfokból álló nyelv P-ben van. Legyen a SÍK-MAXKLIKK nyelv a következő:

$$\{(G,k) \mid G \text{ egy síkgráf, amiben van } k \text{ pontú klikk}\}$$

   Mutassa meg, hogy ez a nyelv NP-teljes, vagy mutassa meg, hogy a nyelv P-ben van.

16. **[Vizsga: 2009. május 28.]** P-beli vagy NP-teljes az alábbi probléma? Adott egy $G = (V, E)$ irányítatlan gráf és az a kérdés, hogy van-e olyan C kör a gráfban, melyhez minden $v \notin C$ csúcsból vezet él.
17. **[Vizsga: 2010. május 27.]** Az árvíz több helyen fenyegeti a gátakat, tudjuk, hogy n kritikus hely van. Ezek közül az i-ediknél a gát megfelelő megerősítéséhez h_i darab homokzsák kell. Ha az erősítés nem történik meg (vagy csak kevesebb homokzsákkal), akkor az i-edik helyen k_i kárt okoz a folyó. Adottak a h_i és k_i pozitív számok, továbbá a gátak megerősítéséhez összesen rendelkezésre álló homokzsákok Z száma ($Z > 0$ egész). Azt szeretnénk meghatározni, hogy ennyi homokzsákkal hogyan tudjuk a kárt minimalizálni, ha feltesszük, hogy a meg nem erősített pontokon keletkező károk összeadódnak.
Fogalmazza meg a feladatot eldöntési problémaként és vagy adjon rá polinomiális algoritmust vagy igazolja, hogy a probléma NP-teljes!
18. **[Vizsga: 2010. május 27.]** Az X probléma bemenete egy binárisan felírt $N > 0$ egész szám, és akkor lesz a válasz `igen`, ha N nem 2-hatvány. Az Y probléma bemenete egy G egyszerű gráf, és akkor lesz a válasz `igen`, ha G csúcsainak színezéséhez 3-nál több szín kell. Ha feltesszük, hogy $P \neq NP$, akkor van-e $X \prec Y$, illetve $Y \prec X$ Karp-redukció?
19. **[Vizsga: 2010. június 17.]** Az X problémában adott egy G dag és egy k pozitív egész szám, a kérdés, hogy van-e G-ben legalább k élű út. Igaz-e, hogy $X \prec$ 3SZÍN, illetve, hogy 3SZÍN $\prec X$?
20. **[Vizsga: 2007. június 12.]** Igazolja, hogy ha a 3SZÍN nyelv benne van $coNP$-ben, akkor $NP = coNP$.

12. Egészértékű programozás

1. **[Vizsga: 2009. június 11.]** Egy adott egyszerű, irányítatlan gráfban maximális méretű párosítást akarunk találni. Írja le ezt a problémát egy egész értékű programozási feladatként! (A kapott egész értékű programozási feladatot nem kell megoldani.)

---

2. **[Vizsga: 2009. június 4.]** Egy adott egyszerű, irányítatlan gráfban maximális méretű teljes részgráfot akarunk találni. Írja le ezt a problémát egy egész értékű programozási feladatként! (A kapott egész értékű programozási feladatot nem kell megoldani.)
3. **[Vizsga: 2010. június 3.]** Fogalmazza meg egész értékű programozási feladatként az alábbi problémát! Egy adott $G = (V, E)$ irányítatlan egyszerű gráfban keresünk olyan maximális méretű $D \subseteq V$ csúcshalmazt, melyre teljesül, hogy minden $x \in V$ csúcsnak legfeljebb 2 szomszédja van a D halmazban!
4. **[Vizsga: 2010. június 17.]** Adott egy $G = (V, E)$ egyszerű, irányítatlan gráf. Egy olyan $W \subseteq V$ halmazt keresünk, amely a lehető legtöbb csúcsból áll és teljesül rá, hogy a gráfban bármely 2 független él 4 végpontjából W legfeljebb 2 pontot tartalmaz.
Hogyan lehet ezt a problémát egészértékű programozási feladatként felírni? (A kapott EP feladatot nem kell megoldani!)

13. Közelítő algoritmusok

1. Bizonyítsuk be, hogy a következő algoritmus legfeljebb $2\tau(G)$ ponttal lefog egy tetszőleges G gráfot: Keresünk egy tovább nem bővíthető független élhalmazt, és kiválasztjuk ezen élek végpontjait. Éles ez a korlát?

---

2. **[Vizsga: 2009. június 17.]** A Ládapakolás problémának tekintsük azt a speciális esetét, amikor az n tárgy mindegyike vagy a vagy b méretű, ahol $0 < a < b < 1$ és $a + b = 1$. Adjon ennek megoldására $O(n)$ lépésszámú algoritmust!
3. **[Vizsga: 2009. május 28.]** Egy csomagküldő szolgálatnál csupa egyforma dobozokba pakolják az elküldendő árut. Céljuk az, hogy a ládákban maradó üres helyet kitöltő anyagból minél kevesebbre legyen szükség. Igaz-e, hogy a Ládapakolásra megismert First Fit eljárás erre a problémára is egy 2-közelítő algoritmus?
4. **[Vizsga: 2006. június 12.]** Van n fájlunk, az i-edik fájl hosszát jelölje h_i. Tegyük fel, hogy a fájlok hosszuk szerint nem csökkenő sorrendben követik egymást, azaz $0 < h_1 \leq h_2 \leq \cdots \leq h_n$. Mentéskor két egyforma méretű lemez áll rendelkezésünkre. A mentésnek sorban kell történnie, előbb az első fájlról kell megmondani, melyik lemezre kerüljön, azután a másodikról, stb. (Fájlokat szétvágni nem szabad, minden fájl teljes egészében kerül az egyik vagy a másik lemezre.) Amikor a soron következő fájl már egyik lemezre sem fér rá, akkor abbahagyjuk az eljárást. Egy ilyen eljárás optimális, ha a lehető legtöbb fájlt lehet segítségével kimenteni.
   Mutassa meg, hogy az a mohó eljárás, amikor a következő fájlt oda tesszük, ahol több hely van, nem feltétlenül optimális. Legfeljebb hány fájllal fogunk kevesebbet kimenteni ezzel a mohó eljárással az optimális (szintén sorrendben mentő) megoldáshoz képest?