Algoritmusok és adatszerkezetek

TG feladatok

Feladatsor I:

1. Hadamard mátrixnak egy olyan nxn-es H_n négyzetes mátrixot nevezünk, amelyre teljesül az alábbi összefüggés: $H_n H_n^T = nI_n$., vagyis a mátrixot a saját transzponáltjával megszorozva az nxn-es egységmátrix n-szeresét kapjuk. Nem ismert pontosan, hogy milyen n számokra létezik Hadamard mátrix, azonban van egy sejtés, hogy ha n osztható 4-gyel, akkor van ilyen mátrix. Ismert azonban egy konstrukció Hadamard mátrixra, amennyiben $n=2^m$, ahol m=1,2,3.., vagyis n=2,4,8,…. A konstrukció egy rekurzív definícióval adható meg, ami a következő:

$$H_2 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$
$$H_4 = \begin{pmatrix} H_2 & H_2 \\ -H_2 & H_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$
$$H_{2^m} = \begin{pmatrix} H_{2^{m-1}} & H_{2^{m-1}} \\ -H_{2^{m-1}} & H_{2^{m-1}} \end{pmatrix},$$

Tervezzen rekurzív algoritmust és készítsen programot, amely adott m-re előállítja és a képernyőre rajzolja valamilyen formában H_{2^m}-et és ellenőrzi, hogy teljesül-e rá a megadott tulajdonság!

2. Oldja meg a következő rekurziót: $T(1) = 8$, és minden $n \geq 2$-re, $T(n) = 3T(n-1) - 15$.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	1	0	0	0	0

Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!

4. Adott egy n pozitív egész szám. Tervezzen algoritmust és készítsen programot, ami lineáris időben generálja az 1 és n közötti bináris számokat sor adatstruktúra használatával. Például n=16 esetén az eredmény a következő: 1 10 11 100 101 110 111 1000 1001 1010 1011 1100 1101 1110 1111 10000
5. Az utazó ügynök problémája a következőképpen van definiálva. Adott egy irányított gráf. Keressük meg egy minimális összköltségű Hamilton körét, vagyis olyan kört ami a gráf összes csúcsán pontosan egyszer halad át. A mohó algoritmus az utazó ügynök problémáira az 1 -es csomópontból indul, és mindig a minimális költség mentén halad a következő csomópont fel. Ezt addig ismétli, amíg nem talál egy olyan csomópontot, amelyet korábban már meglátogatott, vagy nem tud továbbmenni.

Bizonyítsuk be, hogy a fenti mohó algoritmus nem feltétlenül ad optimális megoldást.

Feladatsor II:

1. Tervezzen algoritmust és készítsen programot, amely adott n-re véletlenszerűen előállítja az 1,2,...,n számok egy tetszőleges permutációját, majd a permutációt ciklusokra bontja, végül a kapott ciklusokat az eredeti permutációval együtt kanonikus alakban a képernyőre írja. Egy ciklikus jelölést kanonikusnak nevezünk, ha a következő tulajdonságok teljesülnek rá:

   Minden cikluson belül a legkisebb elem van a ciklus elején.
   A ciklusokat olyan sorrendben írjuk le, hogy az első elemek *csökkenő* sorozatot alkossanak.
   Például az 1,2,3,4,5,6 számok 6 2 1 5 4 2 permutációja ciklusos formában kifejezve lehet a következő: (3 1 6)(5 4)(2).

   Ugyanakkor kanonikus alakban a ciklikus jelölés a következő: (4 5)(2)(1 6 3).

   A program futása során a megadott permutációra az output a következőképpen nézzen ki:

   N=6

   Permutáció: 6 2 1 5 4 2

   Kanonikus ciklikus forma: (4 5)(2)(1 6 3).

2. Oldja meg a következő rekurziót: $T(1) = 2$, és minden $n \geq 2$-re, $T(n) = T(n-1) + n - 1$.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	1	0	0	1
2	0	0	1	0	1	0
3	0	1	0	1	0	0
4	0	1	0	0	1	0
5	0	0	0	1	0	1
6	0	0	1	0	1	0

   Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!

4. Adott egy bináris minta, amely néhány helyen ? helyettesítő karaktert tartalmaz. Tervezzen algoritmust és készítsen programot, ami megkeresi és kiírja az összes olyan bináris karakterláncot, amelyet a helyettesítő karakter 0 vagy 1 helyettesítésével lehet létrehozni. Például a `1?11?00?1?` minta esetén az eredmény:

   1011000010
   1011000011
   1011000110
   1011000111

1011100010
1011100011
1011100110
1011100111
1111000010
1111000011
1111000110
1111000111
1111100010
1111100011
1111100110
1111100111

5. Mutassa meg, hogy Prim és Kruskal algoritmusa mindig ugyanazt a minimális feszítő fát konstruálja azokon a gráfokon, amelyeknél az élek költségei mind különböznek.

Feladatsor III.

1. Négy lány és négy fiú mindegyike páronként egy 0 és 10 közötti pontszám megadásával megmondja, hogy mennyire szívesen házasodna össze egymással. Az eredményeket egy 4x4-es „boldogság mátrixban” tároljuk. Tervezzen algoritmust és készítsen olyan programot, amely megadja az optimális párosítást, vagyis meghatározza, hogy melyik lány melyik fiúval házasodjon össze ahhoz, hogy az „összboldogság” a legnagyobb legyen. Az összboldogság, az egyes párok pontszámainak összege. A program kérje be a 4x4-es mátrixot, majd írja ki az optimális párosítást és az összpontszámot a képernyőre.

   Teszteljük programunkat az alábbi mátrixon:

$$\begin{pmatrix} 7 & 5 & 8 & 2 \\ 7 & 8 & 9 & 2 \\ 3 & 5 & 7 & 9 \\ 3 & 5 & 6 & 7 \end{pmatrix}$$

2. Oldja meg a következő rekurziót: $T(1) = 1$, és minden $n \geq 2$-re, $T(n) = 2T(n-1) + n - 1$.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	**2**	**3**	**4**	**5**	**6**
1	0	0	1	0	0	1
2	0	0	1	0	1	0
3	0	1	0	1	0	0
4	0	1	0	1	1	0
5	0	0	0	1	0	1
6	0	0	1	0	0	0

   Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!

4. Adott egy n hosszúságú rúd és az i hosszúságú rudak árainak listája, ahol 1 <= i<= n. Tervezzen algoritmust és készítsen programot, ami megadja hogyan lehet optimális módon az n hosszú rudat kisebb rudakra vágni a profit maximalizálása érdekében.

   Példa:

   Input:

   hossz[] = [1, 2, 3, 4, 5, 6, 7, 8]

   ar[] = [1, 5, 8, 9, 10, 17, 17, 20]

   Rúd hossza: 4

   A legjobb vágás 2db 2 hosszú részre történik, a költség: 5 + 5 = 10

Vágás	Profit
4	9
1, 3	(1 + 8) = 9
2, 2	(5 + 5) = 10
3, 1	(8 + 1) = 9
1, 1, 2	(1 + 1 + 5) = 7
1, 2, 1	(1 + 5 + 1) = 7
2, 1, 1	(5 + 1 + 1) = 7
1, 1, 1, 1	(1 + 1 + 1 + 1) = 4

5. Tegyük fel, hogy eseményeket kell ütemezni nagyszámú előadóterembe. Az összes esemény egy olyan ütemezését keressük, amely a lehető legkevesebb termet igényli. Készítsen olyan hatékony mohó algoritmust, amely meghatározza, hogy az egyes eseményeket melyik terembe kell beosztani. Az eseményekhez időintervallum tartozik és természetesen egyszerre nem lehet egy teremben két esemény.

Feladatsor IV.

1. Vegyünk egy tetszőleges legalább kétjegyű pozitív egész számot, fordítsuk meg a számjegyeit, majd adjuk az eredeti számhoz. Ismételjük ezt az eljárást addig a kapott összeggel amíg úgynevezett palindrom számot nem kapunk (a szám visszafelé olvasva önmaga). Például az 5280-re a sorozat így néz ki: 5280, 6105, 11121, 23232. Bizonyos számokra, például a 196 nem ismert, hogy létezik-e ilyen sorozat, vagyis kapunk-e valaha palindromot. Tervezzen olyan algoritmust, illetve készítsen programot, mely olyan számokat keres, amelyekre nem dönthető el egyszerűen, hogy létezik-e ilyen sorozat.
2. Oldja meg a következő rekurziót: $T(1) = 4$, és minden $n \geq 2$ kettőhatványra, $T(n) = 2T(n/2) + 3n + 2$.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	2	3	4	5	6
1	0	1	0	1	1	1
2	0	0	1	1	1	0
3	0	1	0	0	0	0
4	0	1	0	0	1	0
5	0	0	1	1	0	1
6	0	0	0	1	1	0

Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!

4. Adott egy bináris tömb, tervezzen algoritmust és készítsen programot, ami lineáris időben és állandó tárhasználattal rendezi a tömböt.
   Példa:
   Input: { 1, 0, 1, 0, 1, 0, 0, 1 }
   Output: { 0, 0, 0, 0, 1, 1, 1, 1 }
5. Tegyük fel, hogy adott egész számoknak két, A és B halmaza, és mindegyik n elemet tartalmaz. A két halmaz elemei tetszőleges sorrendbe rakhatóak. Legyen a_i az A halmaz i-edik eleme a felsorolásban, és legyen b_j B halmaz j-edik eleme a felsorolásban. Ekkor a nyeremény $\prod_{i=1}^{n} {a_i}^{b_i}$. Adjon olyan algoritmust, amely kiszámítja a lehető legnagyobb nyeremény értékét. Bizonyítsa be, hogy az algoritmus maximális nyereményt ad.

Feladatsor V.

1. Tervezzen algoritmust és írjon olyan programot, ami fordított lengyel formájú kifejezéseket értékel ki, verem segítségével! Fordított lengyel formán egy kifejezésnek olyan formáját értjük, amelyben számok illetve műveleti jelek szerepelnek egymástól szóközzel elválasztva. A kiértékelés úgy történik, hogy amennyiben a sorozatban szám következik azt betesszük a verembe, amennyiben művelet, a verem tetején lévő két értékkel elvégezzük a műveletet, és az eredmény a verembe kerül. Például a 3*(4-5*(3-15)/4)+100 kifejezés fordított lengyel formájú alakja 100 15 3 - 4 5 / * 4 - 3 * + Tesztelje a programot a 2 5 + 3 * 9 2 1 + / - kifejezéssel!
2. Oldja meg a következő rekurziót: $T(1) = 3$, és minden $n \geq 2$ háromhatványra, $T(n) = 4T(n/3) + 2n - 1$.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	0	0	1
4	0	1	0	0	1	0
5	1	0	0	1	0	0
6	0	0	0	1	0	0

Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!

4. Adott egy tömb ami csak 0, 1 és 2 elemeket tartalmazhat. Tervezzen algoritmust és készítsen programot, ami lineáris időben és állandó tárhasználattal rendezi a tömböt.
   Példa:
   Input: { 0, 1, 2, 2, 1, 0, 0, 2, 0, 1, 1, 0 }
   Output: { 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2 }
5. Mutassa meg, hogy 8 bites véletlen eloszlású adatállományt a Huffman-algoritmus nem tud tömöríteni.

Feladatsor VI.

1. Tegyük fel egy bináris fát úgy reprezentálunk, hogy egy B vektor a baloldali leszármazottakra, egy J vektor a jobboldali leszármazottakra mutató mutatókat tartalmazza, míg az elemeket egy E vektorban tároljuk. A B és J tömbök egész számokat tartalmaznak és ezek az E vektor indexeit jelentik. A gyökérelem az első vektorelem. Egy másik ismert reprezentáció a szekvenciális reprezentáció, amikor egyetlen vektorban ábrázoljuk a fát, a vektor első eleme tartalmazza a gyökérelemet, és ha egy elem az I. vektorelemben található, akkor a bal és jobb leszármazottjai rendre az 2*i, 2*i+1 indexű vektorelemben vannak. Ahol a fa nem folytatódik, oda speciális jelet teszünk. Tervezzen algoritmust és készítsen programot, amely az első reprezentációval megadott bináris fát a második reprezentáció szerintire alakítja át, majd kiírja az így kapott vektort, végül mindkét reprezentáció szerinti fát bejárja JOBBRÉSZFA-GYÖKÉR-BALRÉSZFA sorrendben és ezzel ellenőrzi, hogy az átalakítás jól sikerült-e.
   Példa fa:
   B: (2,6,4,0,0,12,8,0,10,0,0,0,0)
   J: (3,7,5,0,0,13,9,0,11,0,0,0,0)
   E: ('O','A','E','G','M','!','T','T','D','O','L','!','M')

2. Oldja meg a következő rekurziót: $T(1) = 2$, és minden $n \geq 2$ háromhatványra, $T(n) = 4T(n/3) + 3n - 5$.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	**2**	**3**	**4**	**5**	**6**
1	0	0	1	1	1	0
2	1	0	1	0	0	0
3	0	1	0	0	1	0
4	0	1	0	0	0	1
5	1	1	0	1	0	1
6	1	1	0	1	0	0

Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!

4. Tervezzen algoritmust és készítsen programot, ami egy egész számokat tartalmazó tömbben átrendezi az elemeket úgy, hogy a tömb minden második eleme nagyobb

legyen, mint a bal és a jobb oldali elemek. Tegyük fel, hogy nincsenek azonos elemek a tömbben.
Mi a megoldás időbonyolultsága?

Pl.:
Input: 1, 2, 3, 4, 5, 6, 7
Output: 1, 3, 2, 5, 4, 7, 6

Input: 9, 6, 8, 3, 7
Output: 6, 9, 3, 8, 7

Input: 6, 9, 2, 5, 1, 4
Output: 6, 9, 2, 5, 1, 4

5. Mutassa meg, hogy egy hasítótábla esetén, ha U a lehetséges kulcsok univerzuma, n az adott helyzetben előforduló kulcsok száma, m a táblázat mérete és |U| > nm, akkor van U-nak olyan n méretű részhalmaza, melynek elemei ugyanarra a résre képződnek le, és ezért a láncolásos ütközésfeloldás legrosszabb keresési ideje Θ(n).

Feladatsor VII.

1. Tervezzen algoritmust és készítsen programot, ami tapasztalati úton elemzi a bináris keresés algoritmusát. A program véletlen adatokra futtassa le sokszor a keresést valamilyen méretű tömböt alkalmazva, majd számítsa ki mennyi az átlagos lépésszám.

2. Bizonyítsa be, hogy ha $f_1(n) = O\big(g_1(n)\big)$ és $f_2(n) = O\big(g_2(n)\big)$ akkor $f_1(n) + f_2(n) = O\big(g_1(n) + g_2(n)\big)$.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	**2**	**3**	**4**	**5**	**6**
1	0	0	1	0	1	1
2	1	0	1	0	0	1
3	1	0	0	0	1	0
4	0	1	0	1	0	1
5	0	1	0	0	0	1
6	1	0	0	1	0	0

Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!
4. Tervezzen algoritmust és készítsen programot, ami egy egész számokat tartalmazó tömbben megkeresi a maximális hosszúságú résztömböt, amelyben az elemek összege egy adott érték.
Pl.

Input: 5, 6, -5, 5, 3, 5, 3, -2, 0, Összeg: 8

Output: -5, 5, 3, 5
Mi a megoldás időbonyolultsága?

5. Mit ad vissza a következő függvény? Fejezze ki n függvényében az eredményt. Mi a függvény futási ideje?

```
function rejtely(n)
  r := 0;
  for i := 1 to n − 1 do
       for j := i + 1 to n do
            for k := 1 to j do
                 r := r + 1
  return(r)
```

Feladatsor VIII.

1. Tervezzen algoritmust és készítsen programot, ami tapasztalati úton elemzi a következő véletlen lineáris keresési algoritmust. Ha az adatokat tartalmazó tömb mérete n, akkor először az algoritmus generál egy véletlen számot 1 és n között. Ha az adott indexű elem megegyezik a keresett elemmel, akkor kész vagyunk. Ha nem, akkor a maradék n-1 indexet használva generál újabb véletlen indexet az algoritmus, és így folytatódik tovább a keresés, amíg az összes indexet nem generálta, vagy meg nem találta az elemet. A program véletlen adatokra futtassa le sokszor az algoritmust valamilyen méretű tömböt alkalmazva, majd számítsa ki mennyi az átlagos lépésszám.

2. Bizonyítsa be, hogy ha $f(n) = O\big(g(n)\big)$ és $g(n) = O\big(h(n)\big)$ akkor $f(n) = O\big(h(n)\big)$.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	**2**	**3**	**4**	**5**	**6**
1	0	0	1	1	1	1
2	0	0	1	0	0	0
3	0	1	0	0	1	0
4	1	1	1	1	0	1
5	0	0	1	0	0	1
6	1	0	1	0	1	0

Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!

4. Adott egy sakktábla, készítsünk algoritmust és programot, amely meghatározza, hogy egy huszárnak minimum hány lépésre van szüksége ahhoz, hogy egy adott mezőről elérjen egy adott célmezőt.

   Input: N = 8 (8 x 8-s tábla), Induló mező = (7, 0) Cél mező = (0, 7)
   Output: 6

5. Bizonyítsa be, hogy a következő rekurzív algoritmus $3^n - 2^n$-t számolja ki minden *n>=0*-ra.

```
function g(n)
  if n ≤ 1 then
       return(n)
  else
       return(5*g(n - 1) - 6*g(n - 2))
```

Feladatsor IX.

1. Tegyük fel egy mozi pénztárában m+k ember áll sorba, közülük m-nek ezerforintosa, k-nak ötszázforintosa van. A mozijegyek ára egységesen 500 Ft és nyitáskor nincs pénz a pénztárban. Feltételezzük, hogy mindenki csak egy darab jegyet vásárol. Tervezzen algoritmust és készítsen programot, ami meghatározza az összes olyan lehetséges sorban állást, amikor a sor nem akad el váltópénz hiánya miatt. A sor akkor akad el, ha a pénztáros nem tud visszaadni egyezerforintossal fizető személynek. Összesen legfeljebb 15 ember állhat a sorban.

2. Igaz-e hogy $2^{2n} = O(2^n)$? Indokolja a választ.
3. Adott egy 6 pontból álló irányított gráf, amely csúcsmátrixos ábrázolással a következőképpen néz ki:

	1	**2**	**3**	**4**	**5**	**6**
1	0	0	1	1	1	1
2	1	0	1	0	0	0
3	0	0	0	0	0	0
4	1	1	0	0	0	1
5	0	0	0	1	0	1
6	1	1	1	1	1	0

Készítsen programot, ami előállítja a gráfot és a képernyőre rajzolja!

4. A leghosszabb alternáló részsorozat probléma egy adott sorozat olyan részsorozatának megkeresését jelenti, amelyben az elemek váltakozó sorrendben vannak, és a részsorozat a lehető leghosszabb. Tervezzen algoritmust és készítsen programot, ami megkeresi egy adott sorozat leghosszabb alternáló részsorozatát. Például ha A[] = [8, 9, 6, 4, 5, 7, 3, 2, 4] akkor a leghosszabb alternáló részsorozata a [8, 9, 6, 7, 3, 4] lehet, mivel (8 < 9 > 6 < 7 > 3 < 4).
5. Mit ad vissza a következő függvény? Fejezze ki n függvényében az eredményt. Mi a függvény futási ideje?

```
function rejtely(n)
  r := 0;
  for i := 1 to n do
       for j := 1 to i do
            for k := j to i+j do
                 for l := 1 to i+j-k do
                 r := r + 1
  return(r)
```