

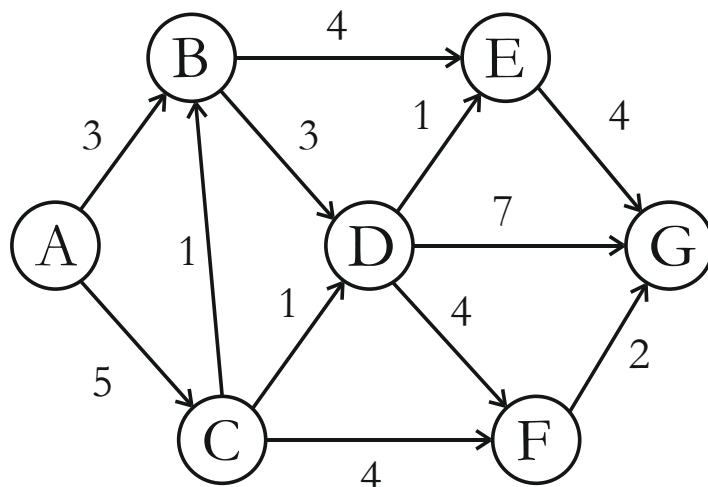


## A 2019/2020 tanévi Országos Középiskolai Tanulmányi Verseny első fordulójának feladatai

### INFORMATIKA II. (programozás) kategória

1. feladat: Mikorra érhet oda (60 pont)

Az alábbi ábrán egy sétapálya látható, amelyről tudjuk, hogy egy ember melyik szakaszt mennyi idő alatt tudja bejárni. A pályán a nyilaknak megfelelő irányba lehet haladni.



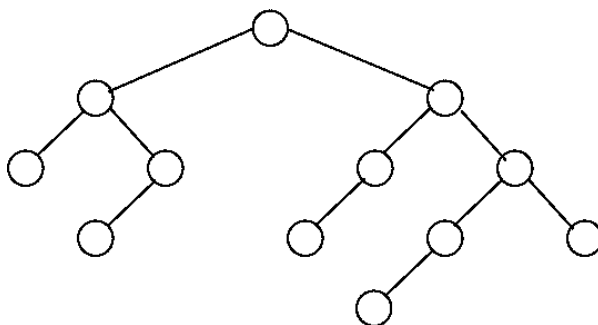
A. Add meg, hogy merre kell menni A-ból G-be, hogy G-be a lehető leghamarabb érjünk! Több megoldás esetén az összes útvonalat add meg!

B..G. Add meg, hogy az egyes pontokra (B..G) leghamarabb mennyi idő alatt lehet odaérni az A pontból indulva, illetve, hogy ezt hányféleképpen lehet megtenni!

2. feladat: Kiegyensúlyozott bináris fa (60 pont)

Bináris fát magasság-kiegyensúlyozottnak nevezünk, ha minden pontjára teljesül, hogy a bal- és jobb-részfájának magassága legfeljebb eggyel tér el. Az üres fa magassága 0, az egy pontú fa magassága 1, egyébként  $1 +$  a két részfa magasságának a maximuma.

Az ábrán látható bináris fa magasság-kiegyensúlyozott, magassága 5, és 12 pontja van.



Legfeljebb mekkora a magassága az olyan magasság-kiegyensúlyozott bináris fának, amelynek

- |                   |                     |                       |
|-------------------|---------------------|-----------------------|
| A: 4 pontja van,  | F: 19 pontja van,   | K: 2019 pontja van;   |
| B: 5 pontja van,  | G: 33 pontja van,   | L: 3333 pontja van;   |
| C: 6 pontja van   | H: 42 pontja van;   | M: 5000 pontja van;   |
| D: 7 pontja van,  | I: 100 pontja van;  | N: 9999 pontja van;   |
| E: 15 pontja van, | J: 1000 pontja van; | O: 25 252 pontja van? |

**3. feladat:** Mit csinál (70 pont)

Az alábbi algoritmus egy  $N$  elemű, 1-től  $N$ -ig indexelt  $x$  sorozatot dolgoz fel, amely 1 és  $M$  közötti egész értékeket tartalmaz. Eredménye az  $A$ ,  $B$ ,  $C$  és  $V$  változókba kerül. Az  $u$  vektort 1-től  $M$ -ig indexeljük.

```

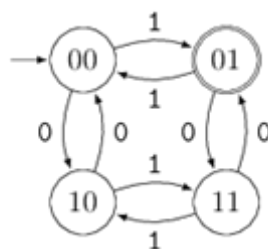
Valami (V, A, B, C) :
  m:=N; u:=(-N, ..., -N)
  Ciklus i=1-től N-ig
    Ha i-u[x[i]]<m akkor m:=i-u[x[i]]; B:=u[x[i]]; C:=i
    u[x[i]]:=i
  Ciklus vége
  V:=(m<N)
  Ha V akkor A:=x[B]
Eljárás vége.
  
```

- Mi lesz  $A$ ,  $B$ ,  $C$  és  $V$  értéke, ha  $N=10$ ,  $x=(3, 5, 3, 2, 2, 4, 2, 8, 9, 1)$  ?
- Mi lesz  $A$ ,  $B$ ,  $C$  és  $V$  értéke, ha  $N=10$ ,  $x=(3, 5, 6, 7, 8, 2, 5, 6, 7, 8)$  ?
- Mi lesz  $u[1] \dots u[8]$ -ban, ha  $N=10$ ,  $x=(3, 5, 6, 7, 8, 2, 5, 6, 7, 8)$  ?
- Milyen  $x$  vektor esetén kap  $V$  hamis értéket?
- A ciklus  $i$ . lépése után mit tartalmaz az  $u$  vektor?
- Mi lesz  $A$ ,  $B$  és  $C$  értéke, ha  $V$  igaz értéket kap az eljárásban?

**4. feladat:** Automata (65 pont)<sup>1</sup>

Egy automata kezdetben 00 állapotban van, jeleket olvas és a jelek hatására az állapota megváltozhat. Ha 00 állapotban a bemenetére 1-es jel érkezik, akkor 01 állapotba kerül, ha 0 érkezik, akkor 10 állapotba. Ha 01 állapotban a bemenetére 0 érkezik, akkor 11-be kerül, ha pedig 1, akkor 00-ba. Ha 10 állapotban a bemenetére 1 érkezik, akkor 11-be kerül, ha pedig 0, akkor 00-ba. Ha 11 állapotban a bemenetére 0 érkezik, akkor 01-be kerül, ha pedig 1, akkor 10-ba.

Az automata az alábbi rajzzal ábrázolható:



<sup>1</sup> A feladat Friedl Katalin, Csima Judit: Nyelvek és automaták című könyvéből származik: [https://www.tan-konyvtar.hu/hu/tartalom/tamop412A/2011-0064\\_58\\_nyelvek\\_es\\_automatak/ar01s02.html](https://www.tan-konyvtar.hu/hu/tartalom/tamop412A/2011-0064_58_nyelvek_es_automatak/ar01s02.html)

A. Milyen állapotban lesz az automata az 11010 jelsorozat hatására? Add meg az egyes jelek utáni állapotot is!

B. Milyen állapotban lesz az automata a 0101010011 jelsorozat hatására? Add meg az egyes jelek utáni állapotot is!

C. Milyen jelsorozatokra lesz az automata a legvégén 00, 01, 10, illetve 11 állapotban?

D. Add meg ábrával azt a minimális állapotszámú automatát, amely képes felismerni az olyan 0 és 1 jelekből álló sorozatokat, amelyekben az 1-esek száma hárommal osztható! Vagyis legyen egy kitüntetett állapota, amelyben pontosan ezen jelsorozatok végén lesz. Dupla bekarikázással jelöld ezt az állapotot!

5. feladat: Gépközeli programozás (40 pont)

Az alábbiakban látható egy gépközeli nyelven megírt programrészlet, amely a C és B címen levő 32 bites pozitív egész számokból számol ki egy 32 bites pozitív egész számot, amit az A címre tesz. A segédszámításokhoz a processzor három, 32-bites regiszterét (a processzor tárolóegységét) használja (eax, ebx, ecx).

Egy kettes számrendszerbeli szám balra léptetése azt jelenti, hogy a legmagasabb helyiértékű bitjét törli és a végére egy 0-s bitet tesz. A jobbra léptetés a legalacsonyabb helyiértékű bitet törli és az elejére tesz egy 0-s bitet.

```
mov ecx, C ; az ecx regiszterbe tölt egy értéket a C címről
mov ebx, B ; az ebx regiszterbe tölt egy értéket a B címről
mov eax, 1 ; az eax regiszterbe 1-et tölt
xxx:  cmp ebx, 0 ; összehasonlítja ebx tartalmát a nullával
      je yyy     ; az yyy címen folytatja, ha egyenlők voltak
      bt ebx, 0  ; megnézi ebx legalacsonyabb helyiértékű bitjét
      jc zzz     ; a zzz címen folytatja, ha 1-es értékű volt
      sq ecx     ; az ecx regiszter tartalmát négyzetre emeli
      shr ebx   ; az ebx regisztert 1 bittel jobbra lépteti
      jmp xxx   ; az xxx címen folytatja
zzz:  mul ecx   ; megszorozza eax-et ecx-szel
      dec ebx   ; eggyel csökkenti ebx-et
      jmp xxx   ; az xxx címen folytatja
yyy:  mov A, eax ; az A címre teszi az eax regiszter tartalmát
```

A. Mi lesz az A címen, ha C=42, B=1?

B. Mi lesz az A címen, ha C=3, B=4?

C. Mi lesz az A címen, ha C=2, B=10?

D. Hogyan függ a kiszámított A értéke B-től és C-től?

E. Milyen esetben lesz az eredmény értelmezhetetlen (hibás)?

F. Hányszor lép a program a zzz címre és ez mitől függ?

G. Hányszor lép a program az xxx címre és ez mitől függ?

6. feladat: Hasító táblák (50 pont)

A hasító tábla egy olyan adatszerkezet, amely segítségével hatékonyan kereshetünk egy kulcs alapján egy adathalmazban. A kereséshez használt kulcsot előbb mindig egy számmá képezzük. A tárolásra egy egyszerű, 0-tól  $K-1$ -ig indexelt tömböt használunk. Beszúrásakor egy  $N$  kulcsú adatot a tömb  $(N \bmod K)$  . pozíciójába szeretnénk tárolni, azonban, ha az már foglalt, akkor sorra a következő pozíciókkal próbálkozunk, ha a tömb végére érünk, akkor az elején folytatjuk a szabad hely keresését.

Például egy 7 elemű hasító táblába szeretnénk a 44-es kulcsú adatot beszúrni. Ez a  $44 \bmod 7=2$ . pozícióra kerül.

		44				
--	--	----	--	--	--	--

Ezután egy 130-as kulcsú adatot szúrunk be, ami a  $130 \bmod 7=4$  . pozícióra kerül.

		44		130		
--	--	----	--	-----	--	--

Ezután a 2-es beszúrásával folytatjuk, ami a  $2 \bmod 7=2$  . pozícióra kerülne, de ez foglalt, ezért próbálkozunk a következő pozícióval, ami szabad.

		44	2	130		
--	--	----	---	-----	--	--

A keresés a beszúráshoz hasonlóan működik. A kulcs alapján számított helyen keressük az adatot. Például, ha a 2-es kulcshoz tartozó adatra vagyunk kíváncsiak, akkor a  $2 \bmod 7=2$  . pozícióban keressük először, ott másik elem van, ezért a következő pozíción folytatjuk, ahol meg is találjuk a keresett értéket, ehhez 2 tömbelem olvasására volt szükség. Ha például a 18-as kulcsot keressük, azt a 4-es pozícióban kell keresnünk, ahol nem ez az elem szerepel, ezért a következő helyen próbálkozunk, az viszont üres, így 2 tömbelem olvasásával megtudtuk, hogy a keresett kulcs nem szerepel az adatok között.

Lehetőség van törlésre is. A keresés segítségével megkeressük az adott kulcsú adatot, majd bejelöljük, hogy az adott elem törölve lett. Azért van szükség a törölt mezők megkülönböztetésére, mert a keresést ezeknél még nem hagyhatjuk abba. Beszúrásakor viszont tehetjük az elemet törölt pozícióra. Például a 44 törlése után így néz ki a tömb:

		törölt	2	130		
--	--	--------	---	-----	--	--

Egymás után végezd el a következő műveleteket egy kezdetben üres, 13 elemű tömböt használó hasító táblán! A beszúrásoknál add meg, hogy melyik pozícióba kerül az adott elem, a kereséseknél pedig, hogy hány tömbelemet kellett olvasni, ahhoz hogy megtaláljuk azt, vagy megbizonyosodjunk a keresett elem hiányáról!

- |                |               |                                       |
|----------------|---------------|---------------------------------------|
| A. Beszúr: 134 | G. Beszúr: 5  | L. Keres: 11                          |
| B. Beszúr: 39  | H. Keres: 265 | M. Keres: 5                           |
| C. Beszúr: 6   | I. Keres: 22  | Töröl: 12                             |
| D. Beszúr: 265 | J. Keres: 5   | N. Keres: 12                          |
| E. Beszúr: 12  | K. Keres: 135 | O. Beszúr: 25                         |
| F. Beszúr: 388 | Töröl: 265    | P. Rajzold le a tömb végső állapotát! |

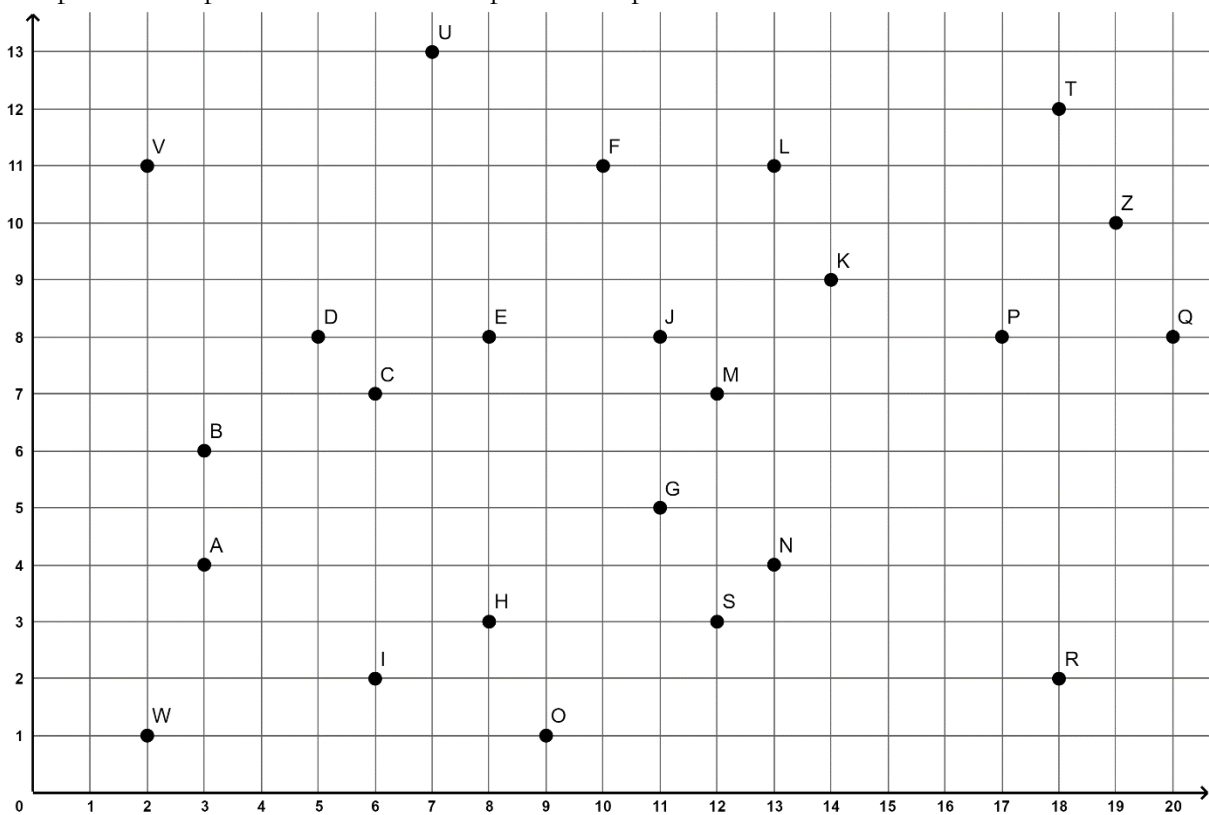
7. feladat: Szenzorhálózat (55 pont)

Repülőgépről ledobálva telepítünk szenzorokat egy területen. A szenzoroknak szükségük van egymás közötti kommunikációra, azonban a jeladók teljesítménye miatt két szenzor csak akkor képes egymással kommunikálni, ha a légvonalbeli távolságuk legfeljebb 5 egység. Távlabbi szenzorpárok ezért csak úgy tudnak egymásnak üzeni, ha más szenzorokon keresztül teszik ezt meg.

Kétféle mohó stratégiával választhatják a szenzorok a csomagok továbbításakor a következő szenzort:

- **Távolság:** Az elérhető szenzorok közül – akár önmagát választva - annak továbbítjuk a csomagot, amely a célponthoz legközelebb van.
- **Irány:** Az elérhető szenzorok közül - önmagát kivéve - annak továbbítjuk a csomagot, amely leginkább a cél irányában van, azaz amelynek az aktuális pontból nézve a célhoz képest vett szögelfordulása minimális.

A képen látható pozíciókba sikerült a repülőről telepítenünk a szenzorokat.



- Add meg a csomag által bejárt útvonalat a K szenzorból az I szenzorba küldéskor **távolság** alapú útválasztás esetén!
- Add meg a csomag által bejárt útvonalat az U szenzorból a W szenzorba küldéskor **távolság** alapú útválasztás esetén!
- Add meg a csomag által bejárt útvonalat a T szenzorból a V szenzorba küldéskor **távolság** alapú útválasztás esetén! Mi történik?
- Add meg a csomag által bejárt útvonalat a W szenzorból az M szenzorba küldéskor **irány** alapú útválasztás esetén!
- Add meg a csomag által bejárt útvonalat a P szenzorból a D szenzorba küldéskor **irány** alapú útválasztás esetén!
- Add meg a csomag által bejárt útvonalat a V szenzorból az R szenzorba küldéskor **irány** alapú útválasztás esetén! Mi történik?