OKTATÁSI HIVATAL

A 2021/2022. tanévi Országos Középiskolai Tanulmányi Verseny első forduló

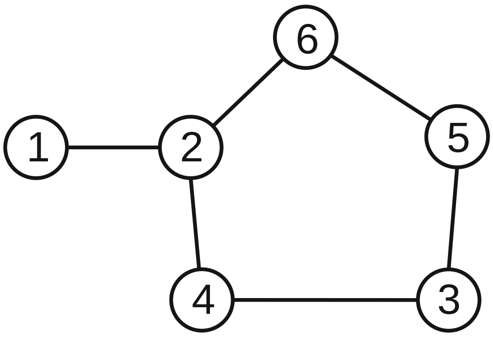
INFORMATIKA II. (PROGRAMOZÁS) KATEGÓRIA
Javítási-értékelési útmutató

Kérjük a tisztelt tanár kollégákat, hogy a dolgozatokat – az egységes értékelés érdekében – szigorúan az alábbi útmutató szerint pontozzák, a megadott részpontszámokat ne bontsák tovább! Vagyis, ha egy részmegoldásra pl. 3 pontot javaslunk, akkor arra vagy 0, vagy 3 pont adható. (Az útmutatótól eltérő megoldások is lehetnek jók.) Az egyes részmegoldásokat az útmutatóban pontosvesszővel választjuk el.

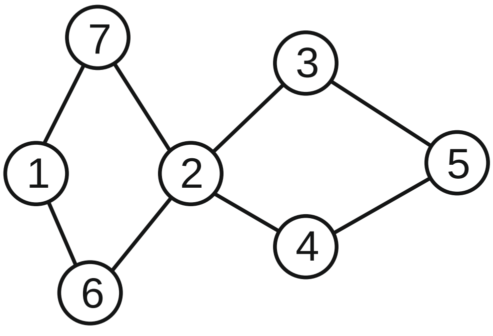
Összpontszám: 400 pont **Beküldési ponthatár: 150 pont**

1. feladat: Utak (44 pont)

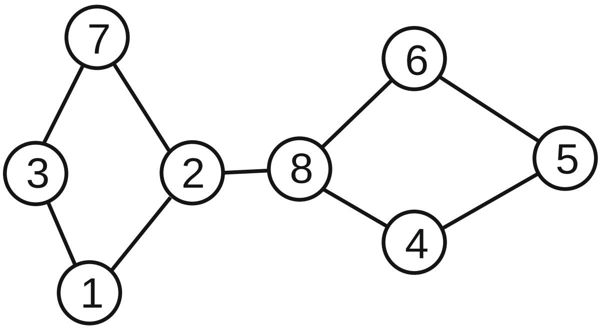
Az alábbi ábrákon sorszámozott települések és közöttük vezető utak láthatók. Egyes településekből indulva el lehet jutni az összes településre úgy, hogy közben egy települést sem érintünk kétszer. Add meg, melyek ezek a települések!



A.



B.



C.

Értékelés:

A. 1; 4; 6	3*4 pont
B. 3; 4; 6; 7	4*4 pont
C. 1; 4; 6; 7	4*4 pont

Minden hibás sorszámért 3 pont levonás, de az egyes részfeladatok pontszáma nem lehet 0-nál kisebb.

Az Országos Középiskolai Tanulmányi versenyek megvalósulását az NTP-TMV-M-21-A0002 projekt támogatja

Miniszterelnökség
Családokért Felelős Tárca Nélküli Miniszter

Nemzeti Tehetség Program

2. feladat: Mit csinál? (58 pont)

Az alábbi algoritmus bementeként kapja a K (K>1) és N értékeket, valamint N darab egész számot az X[1]...X[N] tömbelemekben.

```
D:=0; i:=1; M:=0
Ciklus amíg i≤N és D<K
  Ha X[i] páros akkor D:=D+1
                      Ha D=1 akkor j:=i
  Elágazások vége
  Ha D<K akkor i:=i+1
Ciklus vége
Ha D=K akkor
  M:=i-j+1
  A:=j
  B:=i
  Ciklus amíg i≤N
    i:=i+1
    Ciklus amíg i≤N és X[i] nem páros
      i:=i+1
    Ciklus vége
    j:=j+1
    Ciklus amíg X[j] nem páros
      j:=j+1
    Ciklus vége
    Ha i-j+1>M és i≤N akkor M:=i-j+1; A:=j; B:=i
  Ciklus vége
Elágazás vége
```

A. Mi kerül az M, A, B változókba, ha K=2, N=8, X=[3,2,5,4,3,3,2,1]?

B. Mi kerül az M, A, B változókba, ha K=3, N=8, X=[3,2,5,4,3,3,2,2]?

C. Mi a feltétele annak, hogy a végrehajtás után M értéke 0 maradjon?

D. Fogalmazd meg általánosan, hogyan függ M, A, B értéke a bemenettől!

E. Az első ciklus után az i és a j változó értéke hogyan függ a bemenettől?

F. Mi a szerepe az algoritmusban az i és a j változónak?

Értékelés:

A. M=4; A=4; B=7 — 3+3+3 pont

B. M=6; A=2; B=7 — 3+3+3 pont

C. M=0 marad, ha a tömbben nincs K darab páros szám — 5 pont

D. M= a leghosszabb, pontosan K darab páros számot tartalmazó szakasz hossza az X tömbben, aminek a két szélén páros szám van; A ezen szakasz első; B pedig utolsó számának indexe; ha több ilyen szakasz is van, akkor A és B az első ilyen szakaszhoz tartozik — 5+4+4+2 pont

E. A j az első páros szám indexe; ha van páros szám; az i a K. páros szám indexe; ha van K darab páros szám; ha nincs legalább K darab páros szám, akkor i =N+1 — 4+2+4+1+1 pont

F. A j változó a K darab páros számot tartalmazó szakasz első; az i pedig az utolsó páros számának indexét határozzák meg — 4+4 pont

3. feladat: Törpék (70 pont)

A hét törpe elhatározta, hogy számítógépet fognak játszani, de nem ért minden törpe mindenhez. Morgó és Hapci fájlból tud olvasni (a fájlok neve `egyik`, illetve `másik`), Tudor hasonlítani tud, Szundi és Kuka pedig csak fájlba tud írni. Hapci, Morgó és Tudor folyamatosan figyelnek valamilyen jelzőberendezést (kezdetben mindegyik tilosra van állítva), és ha kell, dolgoznak, ezzel szemben Szundi és Kuka csak akkor dolgozik, ha felszólítást kap (minden felszólításra elindul a programjuk), és a felszólító megvárja, amíg végeznek. Ha már nincs mit olvasniuk, azt is jelzik egy közös logikai változóban. Használnak a jelzőkön kívül két közös adatot (A és B). A két fájlban az adatok növekvő sorrendben vannak, és állományonként egy szám legfeljebb egyszer szerepelhet.

A programjuk így néz ki:

```
Hapci:
   Ciklus amíg nem kész1
     Várj amíg jelző1 szabad
     Ha van adat(egyik) akkor Olvas(egyik,A)
                              jelző1 legyen szabad
     különben kész1 legyen igaz
   Ciklus vége
Eljárás vége.

Morgó:
   Ciklus amíg nem kész2
     Várj amíg jelző2 szabad
     Ha van adat(másik) akkor Olvas(másik,B)
                              jelző2 legyen szabad
     különben kész2 legyen igaz
   Ciklus vége
Eljárás vége.

Tudor:
   Várj amíg jelző1 tilos vagy jelző2 tilos
   Ciklus amíg nem kész1 és nem kész2
     Ha A<B akkor jelző1 legyen tilos
     különben ha A>B akkor jelző2 legyen tilos
     különben Hívd Kukát dolgozni
     Várj amíg jelző1 tilos és nem kész1 vagy
               jelző2 tilos és nem kész2
   Ciklus vége
Eljárás vége.

Kuka:
   Ír(harmadik,A); jelző1 legyen tilos; jelző2 legyen tilos
Eljárás vége.
```

A. Mi lesz a `harmadik`-ban, ha `egyik=(3,5,7,11,13)` és `másik=(1,5,8,11,13)`?

B. Mi lesz a `harmadik`-ban, ha egyik=`(2,4,6,8,10)` és `másik=(2,4,8,16)`?

C. Fogalmazd meg általánosan, hogy a két fájl tartalmától függően mi kerül a harmadikba!

Átírjuk Tudor és Kuka programját, és kap munkát Vidor is:

```
Tudor:
   Várj amíg jelző1 tilos vagy jelző2 tilos
   Ciklus amíg nem kész1 és nem kész2
     Ha A<B akkor Hívd Kukát dolgozni
     különben ha A>B akkor jelző2 legyen tilos
     különben jelző1 legyen tilos; jelző2 legyen tilos
     Várj amíg jelző1 tilos és nem kész1 vagy
               jelző2 tilos és nem kész2
  Ciklus vége
  Ha kész2 akkor Hívd Vidort dolgozni
Eljárás vége.

Kuka:
   Ír(harmadik,A); jelző1 legyen tilos
Eljárás vége.

Vidor:
   Ciklus amíg nem kész1
     Várj amíg jelző1 tilos
     Hívd Kukát dolgozni
   Ciklus vége
Eljárás vége
```

D. Mi lesz a `harmadik`-ban, ha `egyik=(3,5,7,11,13)` és `másik=(1,5,8,11,13)`?

E. Mi lesz a `harmadik`-ban, ha `egyik=(2,4,6,8,10)` és `másik=(4,8,16)`?

F. Fogalmazd meg általánosan, hogy a két fájl tartalmától függően mi kerül a harmadikba! Mi lenne, ha Vidor nem dolgozna?

Tudor programját módosítjuk és bevetjük Szendét és Szundit is:

```
Tudor:
   Várj amíg jelző1 tilos vagy jelző2 tilos
   Ciklus amíg nem kész1 és nem kész2
     Ha A<B akkor Hívd Kukát dolgozni
     különben ha A>B Hívd Szundit dolgozni
     különben jelző1 legyen tilos; jelző2 legyen tilos
     Várj amíg jelző1 tilos és nem kész1 vagy
               jelző2 tilos és nem kész2
  Ciklus vége
  Ha kész1 akkor Hívd Szendét dolgozni
  Ha kész2 akkor Hívd Vidort dolgozni
Eljárás vége.

Szundi:
   Ír(harmadik,B); jelző2 legyen tilos
Eljárás vége.

Szende:
   Ciklus amíg nem kész2
     Várj amíg jelző2 tilos
     Hívd Szundit dolgozni
   Ciklus vége
Eljárás vége
```

G. Mi lesz a `harmadik`-ban, ha `egyik=(3,5,7,11,13)` és `másik=(1,5,8,11,13)`?

H. Mi lesz a `harmadik`-ban, ha `egyik=(2,4,6,8,10)` és `másik=(4,8,16)`?

I. Fogalmazd meg általánosan, hogy a két fájl tartalmától függően mi kerül a harmadikba! Mi lenne, ha Szende nem dolgozna?

Értékelés:

A. `harmadik=(5,11,13)` 5 pont

B. `harmadik=(2,4,8)` 5 pont

C. A harmadikba a két fájl közös elemei kerülnek; növekvő sorrendben 8+2 pont

D. `harmadik=(3,7)` 5 pont

E. `harmadik=(2,6,10)` 5 pont

F. A harmadikba azok kerülnek A-ból, akik nincsenek benne B-ben; növekvő sorrendben; az első fájl azon elemei nem kerülnének az eredménybe, amelyek nagyobbak a második utolsó eleménél 8+2+5 pont

G. `harmadik=(1,3,7,8)` 5 pont

H. `harmadik=(2,6,10,16)` 5 pont

I. A harmadikba azok kerülnek, akik vagy csak A-ban, vagy csak B-ben vannak; növekvő sorrendben; a második fájl azon elemei nem kerülnének az eredménybe, amelyek nagyobbak az elsőfájl utolsó eleménél 8+2+5 pont

4. feladat: Kannák (67 pont)

Egy gazdának három kannája van, az egyik ***A*** literes, a másik ***B***, a harmadik pedig ***C*** literes. Kezdetben az első kanna tele van, a másik kettő pedig üres. Szeretne kimérni pontosan ***L*** liter vizet. Az alábbi műveleteket lehet végezni a kimérés során:

- Áttöltés az A-literesből a B-literesbe (amíg az tele nem lesz, ill. van A-ban)
- Áttöltés az A-literesből a C-literesbe (amíg az tele nem lesz, ill. van A-ban)
- Áttöltés a B-literesből az A-literesbe (amíg az tele nem lesz, ill. van B-ben)
- Áttöltés a B-literesből a C-literesbe (amíg az tele nem lesz, ill. van B-ben)
- Áttöltés a C-literesből az A-literesbe (amíg az tele nem lesz, ill. van C-ben)
- Áttöltés a C-literesből a B-literesbe (amíg az tele nem lesz, ill. van C-ben)

A. Minimum hány öntéssel tud kimérni 6 liter vizet, ha a három kanna 8, 5 és 3 literes? Adj is meg egy lehetséges lépéssort!

B. Minimum hány öntéssel tud kimérni 4 liter vizet, ha a három kanna 8, 5 és 3 literes? Adj is meg egy lehetséges lépéssort!

C. Minimum hány öntéssel tud kimérni 1 liter vizet, ha a három kanna 8, 5 és 3 literes? Adj is meg egy lehetséges lépéssort!

D. Minimum hány öntéssel tud kimérni 7 liter vizet, ha a három kanna 10, 8 és 5 literes? Adj is meg egy lehetséges lépéssort!

E. Minimum hány öntéssel tud kimérni 7 liter vizet, ha a három kanna 14, 8 és 5 literes? Adj is meg egy lehetséges lépéssort!

F. Minimum hány öntéssel tud kimérni 9 liter vizet, ha a három kanna 11, 7 és 5 literes? Adj is meg egy lehetséges lépéssort!

G. Minimum hány öntéssel tud kimérni 3 liter vizet, ha a három kanna 11, 7 és 5 literes? Adj is meg egy lehetséges lépéssort!

H. Minimum hány öntéssel tud kimérni 8 liter vizet, ha a három kanna 11, 7 és 5 literes? Adj is meg egy lehetséges lépéssort!

Értékelés:

A. 3 lépés; (8,0,0) → (3,5,0) → (3,2,3) → (6,2,0) 4+4 pont

B. 6 lépés; (8,0,0) → (3,5,0) → (3,2,3) → (6,2,0) → (6,0,2) → (1,5,2) → (1,4,3) 4+5 pont

C. 4 lépés; (8,0,0) → (5,0,3) → (5,3,0) → (2,3,3) → (2,5,1) 4+4 pont

D. 3 lépés; (10,0,0) → (2,8,0) → (2,3,5) → (7,3,0) 4+4 pont

E. 7 lépés; (14,0,0) → (9,0,5) → (9,5,0) → (4,5,5) → (4,8,2) → (12,0,2) → (12,2,0) → (7,2,5) 4+5 pont

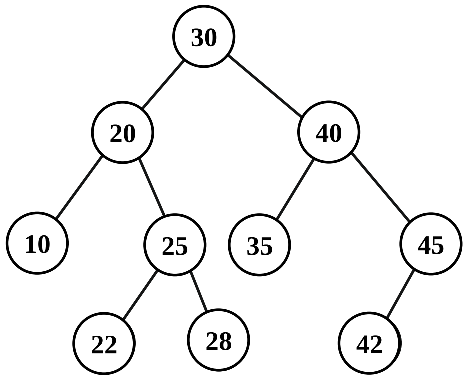
F. 3 lépés; (11,0,0) → (4,7,0) → (4,2,5) → (9,2,0) 4+4 pont

G. 4 lépés; (11,0,0) → (6,0,5) → (6,5,0) → (1,5,5) → (1,7,3) 4+4 pont

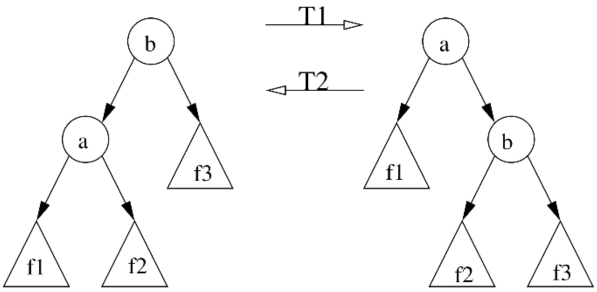
H. 5 lépés; (11,0,0) → (6,0,5) → (6,5,0) → (1,5,5) → (1,7,3) → (8,0,3) 4+5 pont

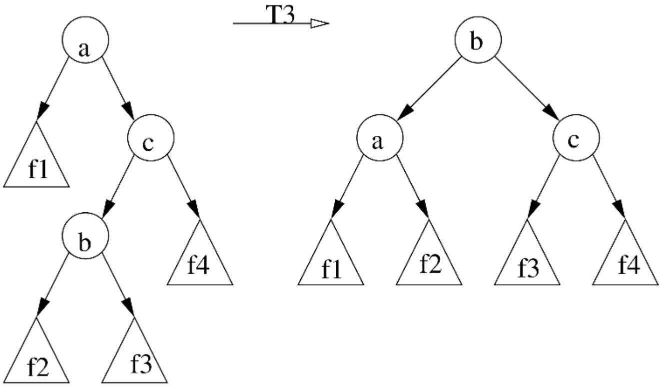
5. feladat: Kiegyensúlyozás (56 pont)

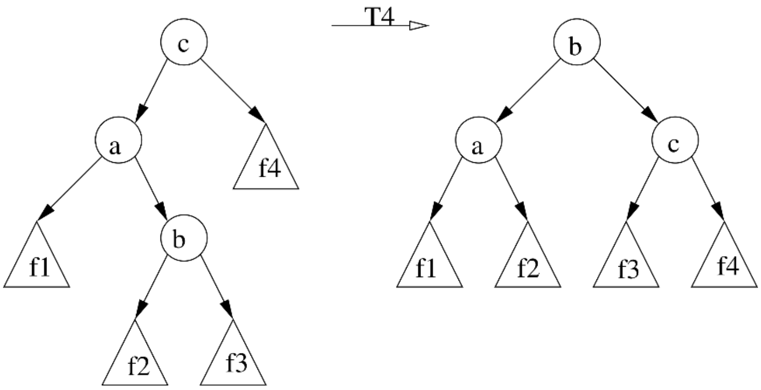
Egy keresőfa minden csomópontjára igaz, hogy tőle balra csak nála kisebb értékű elemek vannak, jobbra pedig nagyobbak, ahogy az ábrán látható.

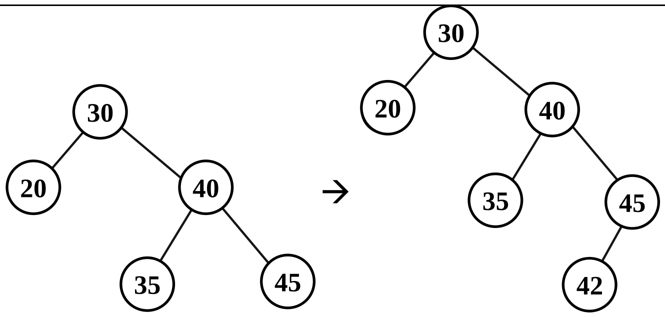
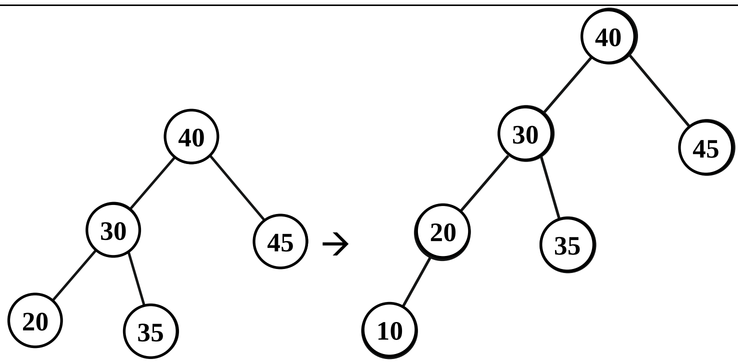


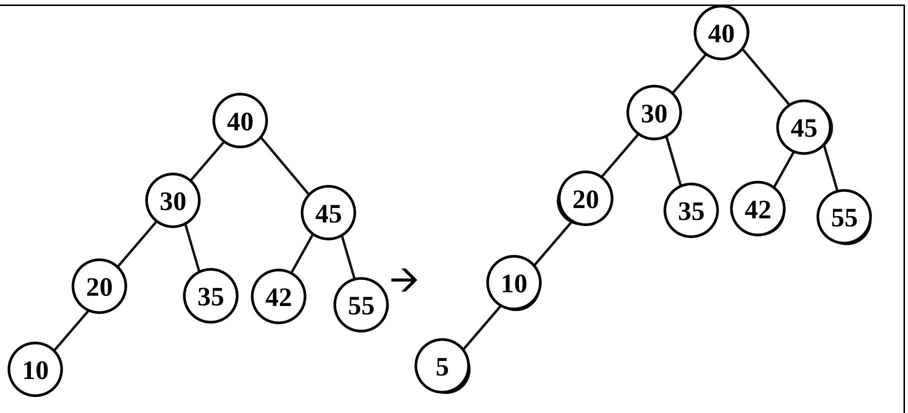
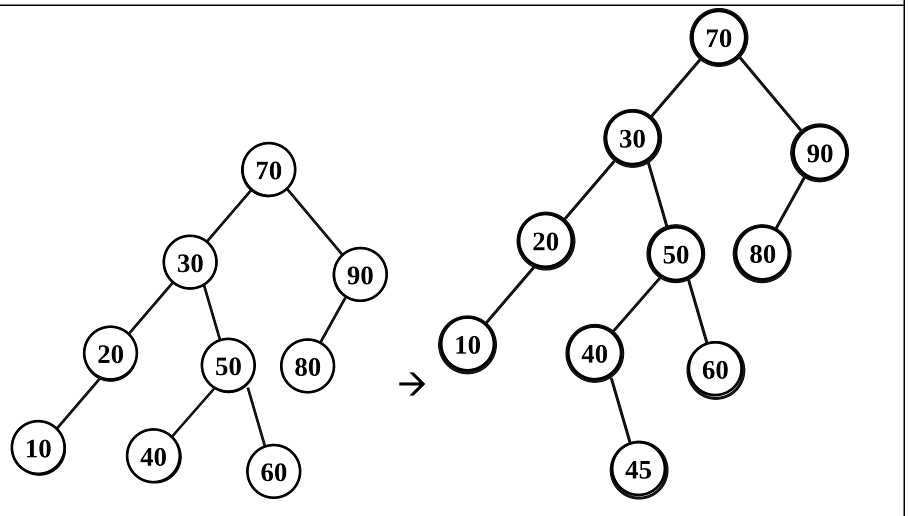
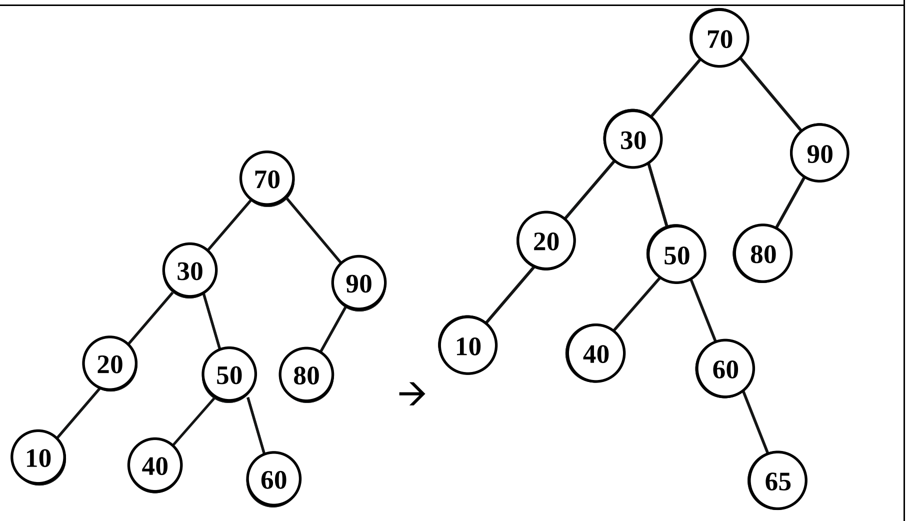
Kiegyensúlyozott az a bináris fa, amelynek tetszőleges pontjában „gyökerező” részfáinak magassága legfeljebb eggyel tér el egymástól (AVL-fa). Ha egy fa kiegyensúlyozottsága elromlik, akkor azt helyre lehet állítani a keresőfa tulajdonság megtartásával, a fa ügyes transzformálásával. Ha a fa egy része válik kiegyensúlyozatlanná, akkor a transzformációt a legalsó kiegyensúlyozatlan részre kell elvégezni! Ezek a szabályos transzformációk:

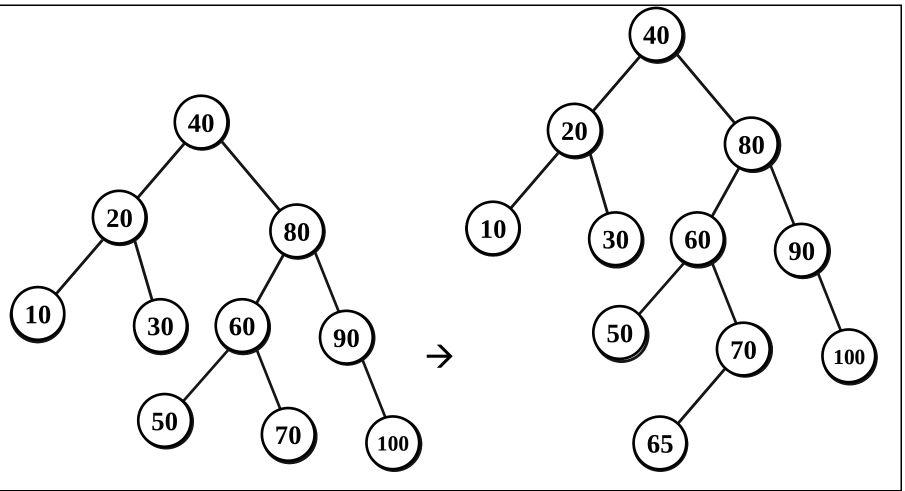
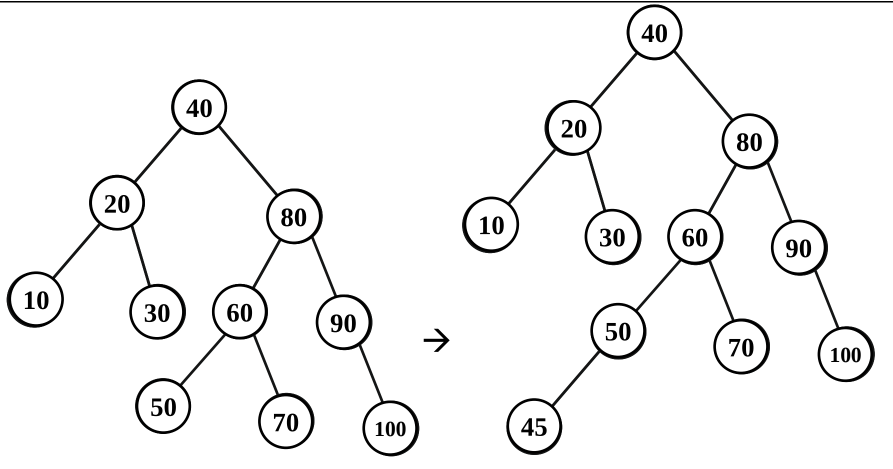




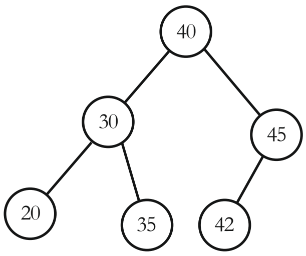


A. Ha ebbe a keresőfába a 42-t szúrjuk be, akkor így romlana el a kiegyensúlyozottsága. Rajzold le a kiegyensúlyozott fát egy szabályos transzformáció után!	
B. Ha ebbe a keresőfába a 10-et szúrjuk be, akkor így romlana el a kiegyensúlyozottsága. Rajzold le a kiegyensúlyozott fát egy szabályos transzformáció után!	

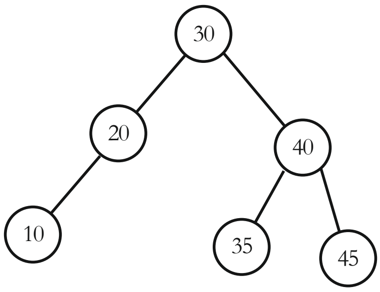
C. Ha ebbe a keresőfába az 5-öt szúrjuk be, akkor így romlana el a kiegyensúlyozottsága. Rajzold le a kiegyensúlyozott fát egy szabályos transzformáció után!	
D. Ha ebbe a keresőfába a 45-öt szúrjuk be, akkor így romlana el a kiegyensúlyozottsága. Rajzold le a kiegyensúlyozott fát egy szabályos transzformáció után!	
E. Ha ebbe a keresőfába a 65-öt szúrjuk be, akkor így romlana el a kiegyensúlyozottsága. Rajzold le a kiegyensúlyozott fát egy szabályos transzformáció után!	

F. Ha ebbe a keresőfába a 65-öt szúrjuk be, akkor így romlana el a kiegyensúlyozottsága. Rajzold le a kiegyensúlyozott fát egy szabályos transzformáció után!	
G. Ha ebbe a keresőfába a 45-öt szúrjuk be, akkor így romlana el a kiegyensúlyozottsága. Rajzold le a kiegyensúlyozott fát egy szabályos transzformáció után!	

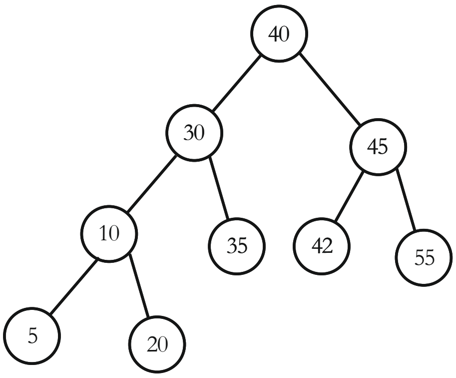
Értékelés:



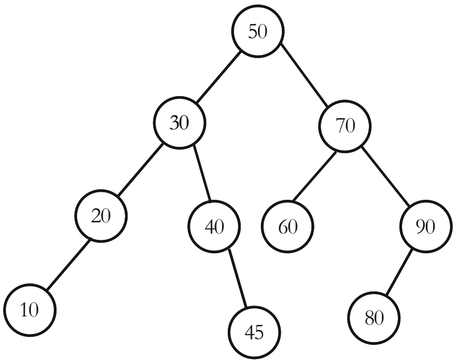
A. 8 pont



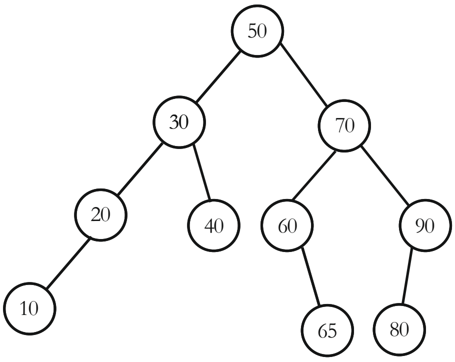
B. 8 pont



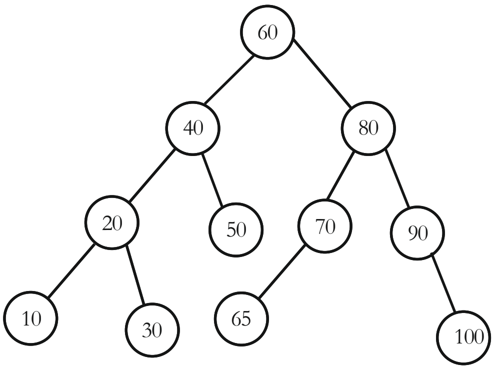
C. 8 pont



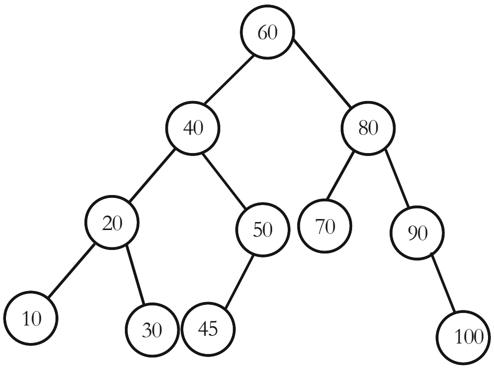
D. 8 pont



E. 8 pont

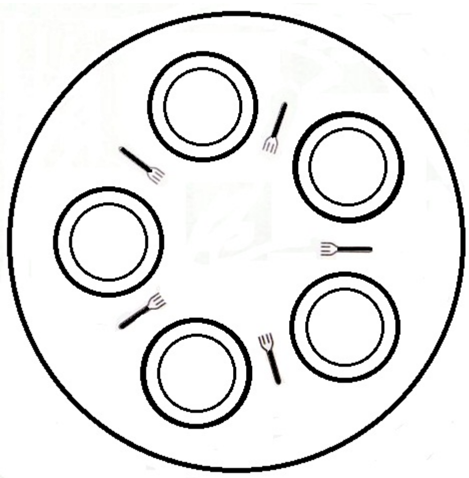


F. 8 pont



G. 8 pont

6. feladat: Étkező filozófusok (45 pont) [1]



Adott öt filozófus (0-tól 4-ig sorszámozva), akik egy asztal körül ülnek és beszélgetnek, filozofálgatnak egymással. Mindegyik filozófus előtt van egy tál spagetti, hogyha megéhezne, tudjon enni. Mindegyik tányér mellett van egy villa. Egy apró probléma van: túlságosan csúszósra sikerült a spagetti, így két villára van szükség az evéshez.

Mindegyik filozófus gondolkodik, majd ha megéhezik, próbálja megszerezni a tányérja melletti villákat, hogy egyen. Ha evett, visszateszi a villákat a helyükre, és folytatja a gondolkodást. A feladat az, hogy készítsünk olyan programot, ami szimulálja ezt a folyamatot!

Egy kézenfekvő megoldás lehet az alábbi, ahol a `kell_villa` művelet addig várakozik, amíg valaki kezében van a villa, a villát pedig a `nemkell_villa` művelettel lehet letenni:

```
filozófus(i):
 Ciklus
  gondolkodom()
  kell_villa(i)   // A bal oldali villát kell megszerezni.
  kell_villa((i+1) mod 5)  // A jobb oldali villa is kell.
  eszem()
  nemkell_villa(i)    // A bal oldali villát visszarakom.
  nemkell_villa((i+1) mod 5) // A jobb oldali villát vissza.
 Ciklus vége
Eljárás vége.
```

Sajnos kialakulhat éhezési helyzet, azaz szakszóval holtpont, amikor senki nem tud eljutni az evéshez.

A. Fogalmazd meg, milyen esetben lehetséges ez a holtpont?

B. Maximum hány filozófus tudna egyszerre enni?

C. Maximum hány filozófus foghat meg legalább 1 villát egyszerre, hogy ne alakuljon ki holtpont? Magyarázd meg, hogy miért! Mennyi időegység alatt végeznek az evéssel ebben az esetben, ha mindenki egyszer eszik és egy evés 1 időegységig tart? Magyarázd meg, hogy miért!

D. Legjobb esetben mennyi időegység alatt végeznek az evéssel ebben az esetben, ha mindenki egyszer eszik és egy evés 1 időegységig tart? Fogalmazd meg, hogy hogyan!

[1] https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2010-0011_szamalap2/lecke6_lap4.html tankönyv alapján.

Értékelés:

A. Ha mind az 5 filozófus felveszi a bal oldali villát, mielőtt bárki a jobb oldaliért nyúlna 5 pont

B. 2 filozófus tud egyszerre enni 5 pont

C. Maximum 4 filozófus foghat egyszerre villát; mert aki mindkét villát fogja, annak a jobb oldali szomszédja bal villára fog várni, a többiek pedig fel tudják venni a bal oldali villájukat 5+5 pont

Öt időegység kell ilyenkor az evés befejezéséhez; mert az első után csak a bal oldali szomszédjának lesz két villája, ha ő végzett, akkor az ő bal oldalijának ... és így tovább 5+5 pont

D. Három időegység alatt végezhetnek; ha először két nem szomszédos filozófus fog két-két villát, miután végeztek újabb két nem szomszédos, s végül az ötödik 8+7 pont

7. feladat: Sorozat (60 pont)

Az alábbi algoritmus egy sorozat értékeit számolja ki:

```
a[1]:=1
Ciklus i=1-től n-1-ig
  Ha valami(i,i) akkor a[i+1]:=a[i]+2 különben a[i+1]:=a[i]+1
Ciklus vége
```

A számítás használja a `valami`, logikai értékű függvényt:

```
valami(m,n):
  Ha m<1 akkor valami:=hamis
  különben ha a[m]=n akkor valami:=igaz
  különben valami:=valami(m-1,n)
Függvény vége.
```

A. Mi kerül az a vektorba, ha `n=10`?

B. Mi a feladata a `valami(m,n)` függvénynek?

C. A `valami` függvény kiszámítása hatékonyabbá tehető egyetlen elágazás feltétele megváltoztatásával. Melyik feltétel és mire cserélendő? Miért?

D. A `valami` függvény rekurzió helyett ciklussal is megoldható az alábbi struktúrában. Egészítsd ki, hogy hatása azonos legyen a rekurzív megvalósítással!

```
valami(m,n):
  Ciklus amíg[                              ]  {*}
    [                              ] {**}
  Ciklus vége
  Ha[                            ]  akkor valami:=hamis {***}
                                 különben valami:=igaz
Függvény vége.
```

E. Egy újabb tömb bevezetésével, egy elemének vizsgálatával a `valami` függvény hívása megszüntethető. Az új tömb elemei egyszerű értékadással számíthatók. Írd le, hogy hogyan?

Értékelés:

A. `1, 3, 4, 6, 8, 9, 11, 12, 14, 16` 10*2 pont

B. Eldönti, hogy az `n` érték szerepel-e az `a` vektorban 8 pont

C. Az `m<1` feltétel helyett; `a[m]<n` legyen; ha `a[m]<n`, akkor az a vektorban már nem lehet az `n` érték, mert az a vektor elemei növekvő sorrendben vannak 4+4+4 pont

```
valami(m,n):
  Ha a[m]<n akkor valami:=hamis
  különben ha a[m]=n akkor valami:=igaz
  különben valami:=valami(m-1,n)
Függvény vége.
```

D. * helyére: `m≥1 és a[m]≠n` 4 pont
** helyére: `m:=m-1` 4 pont
*** helyére: `m<1` 4 pont
Megjegyzés: A C részfeladatban szereplő algoritmus is módosítható.

E. Tároljuk a `b` tömb `b[j]` elemében, hogy a j érték szerepel-e már az `a` tömbben! Ilyen lehet pl. a megoldás: 8 pont

```
a[1]:=1; b[1]:=igaz
Ciklus i=1-től n-1-ig
  Ha b[i] akkor a[i+1]:=a[i]+2 különben a[i+1]:=a[i]+1
  b[a[i+1]]:=igaz
Ciklus vége
```

Más, ezzel ekvivalens megoldás is elképzelhető.