Ütemterv

Programtervezési ismeretek c. tárgyhoz **(GEMAK211B vagy GEMAK211-B)**
BSc gazdaságinformatikus és programtervező informatikus alapszakok számára
Óraszám: heti 2+2, (aláírás+kollokvium, 5 kredit)
2019/20-es tanév I. félév.

Előfeltétel: –

1. hét Adatok, adattípusok, adatműveletek és adatstruktúrák. Szám adattípus.

2. hét Számrendszerek, konverziók. Adatábrázolás és realizálási konvenciók.

3. hét A logikai, halmaz, karakter, string absztrakt adattípusok és realizációjuk.

4. hét A tömb (vektor, mátrix), rekord, egyéb absztrakt adattípusok. Az algoritmus.

5. hét Iteratív és rekurzív algoritmus. A számítógépes memória. Adat és program.

6. hét Gépi kód, magasszintű nyelv, fordítás és interpreter. A verem és a procedúra.

7. hét *1. zárthelyi dolgozat megírása.*

8. hét Az algoritmus lejegyzése. A folyamatábra és a pszeudokód.

9. hét Elemi algoritmusok. Probléma megoldás, programtervezés.

10. hét Struktúrált programozás. A nem struktúráltság vizsgálata.

11. hét Feladat funkcionális felbontása. Struktogram. Tesztelés és dokumentáció.

12. hét Kiselőadások megtartása.

13. hét *2. zárthelyi dolgozat megírása.*

14. hét Pótzárthelyi dolgozat megírása.

A tárgy lezárásának módja: aláírás, kollokvium

Az aláírás feltétele:

- Az előadások felkészült, rendszeres látogatása. Legfeljebb az előadások 40%-áról (azaz 6 alkalommal) lehet hiányozni. Ellenkező esetben a hallgató nem szerezhet aláírást!
- A gyakorlatokon való részvétel, legfeljebb a gyakorlatok 30%-áról (azaz 4 alkalommal) lehet hiányozni. Ellenkező esetben a hallgató nem szerezhet aláírást!

- A félév során a két zárthelyi dolgozat (7. és 13. héten) legalább elégséges szintű (6 pontból legalább 3 pont) megírása. A zárthelyi dolgozatok megírása mindenki számára kötelező. A zárthelyi dolgozatok elméleti kérdéseket (tételek, definíciók) illetve számolási feladatokat tartalmaznak.

- Ha valamelyik zárthelyi dolgozat nem sikeres, akkor azt az utolsó héten lehet pótolni a megfelelő tananyagrészekből. Ha ez sem sikeres, akkor a későbbiekben az egész féléves anyagból kell pótolni.

A vizsga írásbeli. A vizsga 90 perces és 8 pontot lehet maximálisan megszerezni (azaz összesen 8 feladat beugró nélkül és minden feladat tökéletes megoldása 1 pontot ér). A vizsga során számonkérésre kerülnek pl. az alapalgoritmusok, melyek beugrónak számítanak a vizsgán, azaz ezek teljesítése kötelező a legalább elégséges jegy megszerzéséhez. Az elégséges jegy 4 ponttól van meg. Minden újabb pont megszerzése lényegében egy jeggyel javítja a vizsgajegyet, azaz a pontok alapján a jegyek kiosztása a következő: 0-3p elégtelen(1); 4p elégséges(2); 5p (közepes); 6p (jó); 7-8p jeles(5) az eredmény.

Meg nem engedett eszközök használata esetén a vizsga elégtelen és további vizsga abban a vizsgaidőszakban csak szóban, bizottság előtt, a tanszék által megadott időpontban lehetséges.

Miskolc, 2019. szeptember 2. Dr. Olajos Péter (a tárgy jegyzője)

Programtervezési ismeretek, 1. zh., MINTA

1. Definiálja a következő fogalmakat: **div** művelet, **mod** művelet és elemi konjunkció!

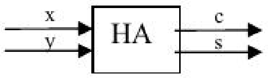
   Definíció. (Az egész hányados képzése, a div művelet) Legyen a és b egész szám ($a, b \in \mathbb{Z}$), $b \neq 0$. Az egész osztás műveletén az a/b osztás eredményének alsó egész részét értjük. Tömören:
   $$a \operatorname{div} b = \lfloor a/b \rfloor.$$

   Definíció. (Az egész maradék képzése, a mod művelet) Legyen a és b egész szám. Ekkor
   $$a \bmod b = \begin{cases} a, & \text{ha } b = 0, \\ a - \lfloor a/b \rfloor \cdot b = a - (a \operatorname{div} b) \cdot b, & \text{ha } b \neq 0. \end{cases}$$

   Definíció. (Elemi konjunkció) Változók vagy tagadottjainak a konjunkciója, melyben a változók legfeljebb egyszer fordulnak elő.

   (1 pont)

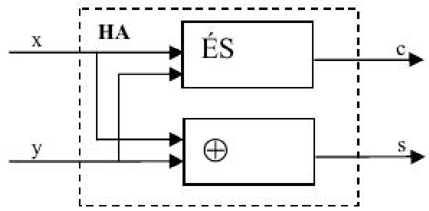
2. Adja meg a fél- és a teljes összeadót! (műveleti tábla, kapuáramkör, képzési szabályok)!

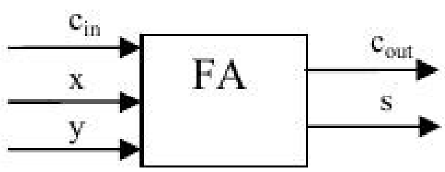


Láthatóan
$c = x \wedge y$
$s = x \oplus y$

Művelettábla

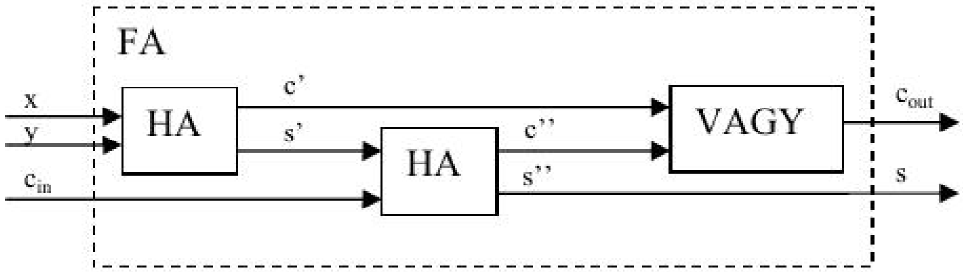
x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0





Művelettáblája:

x	y	c_{in}	c_{out}	s
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1



$$c_{\text{out}} = c' \vee c'' \text{ és } s = s''$$

(1 pont)

3. Számítsa ki a következő kifejezések értékét!

- $-122 \equiv -122 - \left\lfloor \frac{-122}{7} \right\rfloor \cdot 7 \equiv = -122 - (-18) \cdot 7 \equiv 4 \mod 7 = 4;$
- $\lfloor -230,7 \rfloor = \max(-\infty; \ldots; -232; -231) = -231;$
- $\{-15,95\} = -15,95 - \lfloor -15,95 \rfloor = -5,95 - \max(-\infty; \ldots; -17; -16) = -15,95 - (-16) = 0,05;$
- $\text{Round}(10,7) = \lfloor 10,7 + 0,5 \rfloor = \lfloor 11,2 \rfloor = \max(-\infty; \ldots; 10; 11) = 11;$
- $107 \text{ div } 2 = \lfloor \frac{107}{2} \rfloor = \lfloor 53,5 \rfloor = \max(-\infty; \ldots; 52; 53) = 53.$

(1 pont)

4. Írja fel a 12610.296875 értékét kettes és tizenhatos számrendszerben!

12610	2	
6305	0	
3152	1	
1576	0	
788	0	
394	0	
197	0	
98	1	
49	0	
24	1	
12	0	
6	0	
3	0	
1	1	
0	1	↑

és

	2	0,296875
↓	0	0,59375
	1	0,1875
	0	0,375
	0	0,75
	1	0,5
	1	0

Ezek alapján $12610,296875 = 11000101000010,010011_2 =$

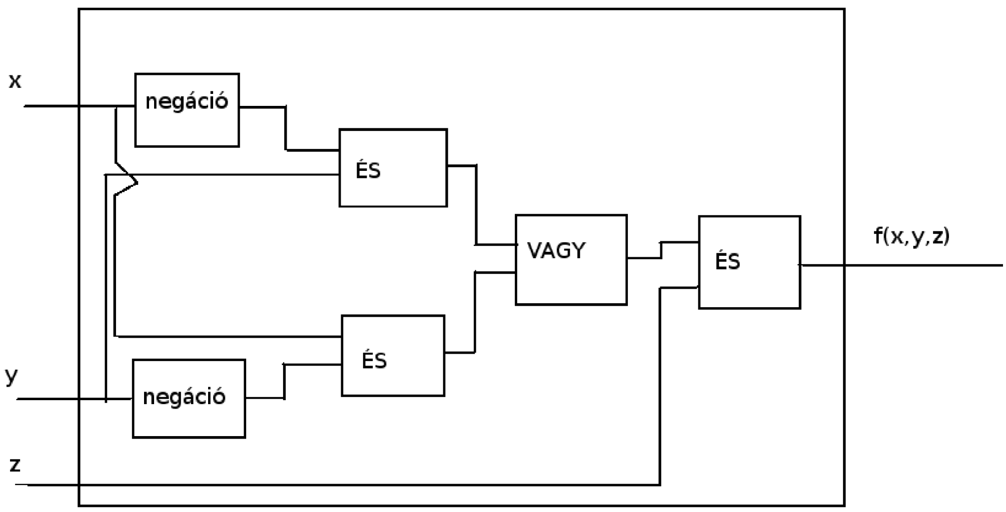
$$= \overbrace{\boxed{00}11}^{3}\ \overbrace{0001}^{1}\ \overbrace{0100}^{4}\ \overbrace{0010}^{2},\overbrace{0100}^{4}\overbrace{11\boxed{00}}^{C}{}_2 = 3142,4\mathrm{C}_{16}.$$

(1 pont)

5. Adott az $f(x, y, z)$ függvény művelettáblája. Készítsen kapuáramkört ehhez a függvényhez a $\wedge$, az $\vee$ és a negáció segítségével!

x	y	z	$f(x,y,z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	$\boxed{1}$
1	0	0	0
1	0	1	$\boxed{1}$
1	1	0	0
1	1	1	0

Csak az 1-es végeredményeket figyeljük (már bekeretezve vannak)!
Ekkor $f(x,y,z) = (\overline{x} \wedge y \wedge z) \vee (x \wedge \overline{y} \wedge z) = ((\overline{x} \wedge y) \vee (x \wedge \overline{y})) \wedge z (= (x \oplus y) \wedge z)$.



(1 pont)

6. Milyen bájtokon ábrázolható az „U+288F” UTF szimbólum UTF-8 kódolással?

A 288F az 10 1000 1000 1111 bináris alakban.
A tárolásához 14 bit szükséges.
Az UTF-8 ábrázoláshoz 3 byte szükséges.
3 bájtos keret: 1110.... 10...... 10......
Kitöltve a biteket: 11100010 10100010 10001111
Hexadecimálisan a karakter az E2 A2 8F bájtokon ábrázolható.

(1 pont)

Összes pont: 6 pont

Programtervezési ismeretek, 2. zh., MINTA

Hallgató neve: .... **Neptun kód:** ....

1. Definiálja meg a következő fogalmakat: valódi program, a vezérlőgráf lebontása. (1 pont)

   Megoldás:

   Valódi program: Egy programot valódi programnak nevezünk, ha:

   1. programgráfja véges számú nem zérus bemenő éllel és kimenő éllel rendelkezik,

   2. programgráfjának csomópontjai predikátumok, függvények és gyűjtők,

   3. programgráfjának minden csomópontján keresztül vezet legalább egy útvonal, amely egy bemenő éllel és egy kimenő éllel rendelkezik (a programgráf kezdő élétől a kilépési élig vezet).

   A vezérlőgráf lebontása: A vezérlőgráf lebontásának nevezzük azt az eljárást, melynek során a struktúrált alapszerkezetek valamelyikét egy függvény csomóponttal helyettesítjük, és ezt mindaddig folytatjuk, amíg ez lehetséges. Az egyetlen csomópontból álló gráfot egyetlen éllel helyettesítjük. (Az egyetlen irányított él neve üres program.)

2. Írja le a következő elemi algoritmusokat pszeudó-kóddal: **lineáris keresés** (Eldöntés és kiválasztás együtt. Adott T tulajdonságú elemet megtalálni az input sorozatban, ha van benne. Ha nincs, akkor az az eredmény, hogy nincs ilyen.), **megszámolás** (A T tulajdonságú elemek számát kell meghatároznunk.)! (1 + 1 pont)

   Megoldás:

   IV. Lineáris keresés: Eldöntés és kiválasztás együtt. Adott T tulajdonságú elemet megtalálni az input sorozatban, ha van benne. Ha nincs, akkor az az eredmény, hogy nincs ilyen.

   A pszeudokód:

Lineáris keresés(@X, @Van, @$Sorszám$)
//Input: X − sorozat $x_i \in \mathbf{A}$
//Output: $Sorszám \in \mathbb{Z}$ − a T tulajdonságú elem indexe
//$Van = igaz$ esetén van T tulajdonságú elem és akkor
//$Sorszám$ egy ilyen elemnek az indexe.
//$Van = hamis$ esetén nincs T tulajdonságú elem és akkor
//$Sorszám$ értelmezhetetlen.
//T : $\mathbf{A} \to \mathbf{L}$
$i \leftarrow 1$
WHILE ($i \leq$ Hossz[X]) AND $\overline{\mathrm{T}(x_i)}$ **DO**
    INC(i)
$Van \leftarrow$ ($i \leq$ Hossz[X])
IF Van
    THEN $Sorszám \leftarrow i$
        RETURN(Van, $Sorszám$)
    ELSE **RETURN**(Van)

V. Megszámolás: A T tulajdonságú elemek számát kell meghatároznunk.

A pszeudokód:

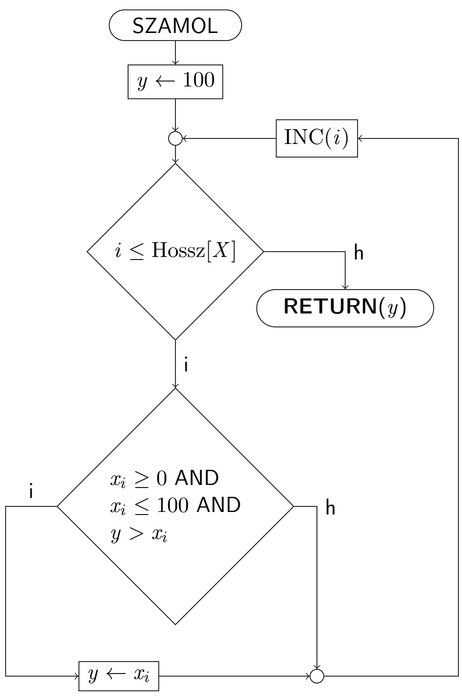
Megszámolás(@X, @$Darab$)
//Input: X – sorozat $x_i \in \mathbf{A}$
//Output: $Darab \in \mathbb{Z}$ – a T tulajdonságú elemek száma
//T: $\mathbf{A} \rightarrow \mathbf{L}$
$Darab \leftarrow 0$
FOR $i \leftarrow 1$ **TO** Hossz[X] **DO**
    IF T(x_i)
        THEN INC($Darab$)
RETURN($Darab$)

3. Írjon egy procedúrát, amely egy bemenetként kapott, valós értékeket tároló tömbre meghatározza a $[0; 100]$ intervallumban lévő értékek közül a legkisebbet (feltételezhetjük, hogy legalább egy elem biztosan van a tömbben, amely a $[0; 100]$-ba esik)! Írja fel a pszeudókódot és készítsen folyamatábrát! (1 pont)

Megoldás:

SZAMOL(@X, @c)
// Input: $X \in \mathbb{R}^n$
// Output: $y \in \mathbb{R}$
$y \leftarrow 100$
FOR $i \leftarrow 1$ TO Hossz[X] DO
    IF $x_i \geq 0$ AND $x_i \leq 100$ AND $y > x_i$
        THEN $y \leftarrow x_i$
RETURN(y)

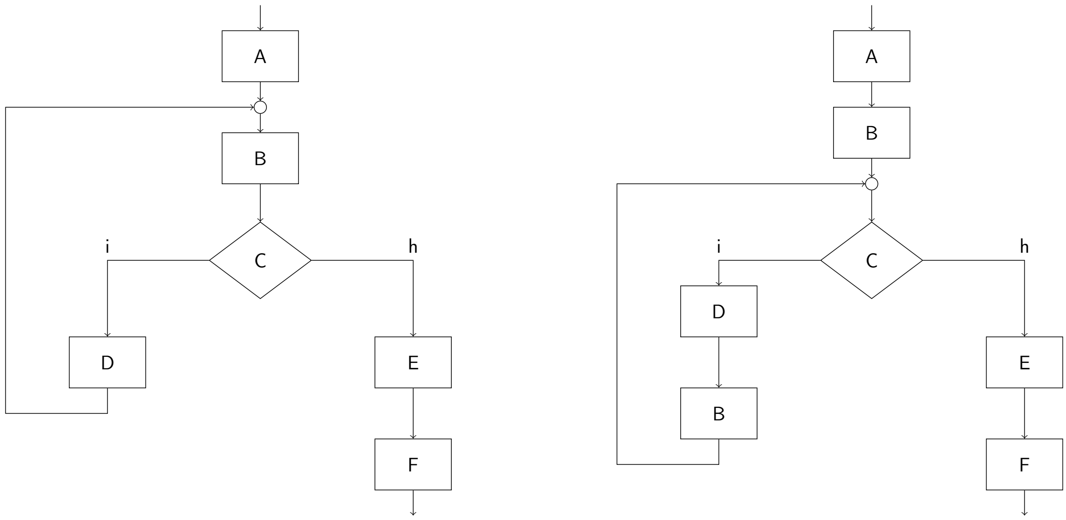
A folyamatábra:



4. Rajzolja fel a következő, tömör élhalmazos formában megadott programgráfot! Rajzoljon fel egy olyan vele ekvivalens struktúrált programgráfot, amelyben csak a definíció szerinti három alapszerkezet fordulhat elő!

   $P = \{\text{START} \to A,\ A \to B,\ B \to C,\ C \to (D,\ E),\ D \to B,\ E \to F,\ F \to \text{STOP}\}$ (1 pont)

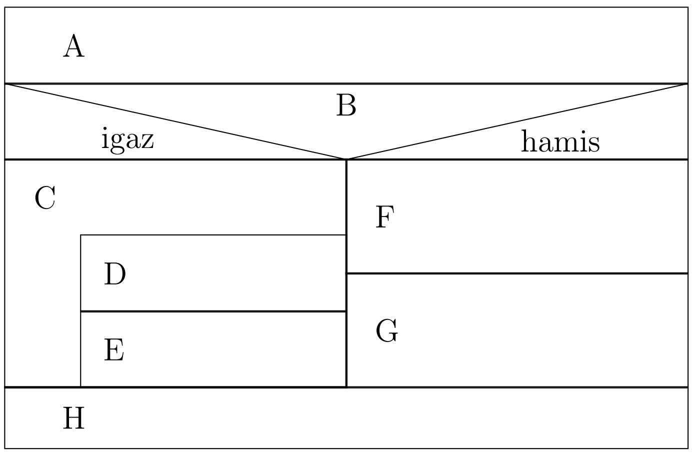
   Megoldás: (bal oldali az eredeti, a jobb oldali a struktúrált)



5. A következő formula egy struktúrált program formulája. Számítsa ki ennek a ciklikus bonyolultságát és készítse el a struktogramját!

   $\mathbf{S}(A, \mathbf{E}(B; \mathbf{C}(C; \mathbf{S}(D, E)), \mathbf{S}(F, G), H)$ (1 pont)

   Megoldás:



   A ciklikus bonyolultság tétele alapján a ciklikus bonyolultság $2 + 1 = 3$, mert két prédikátum van a formulában (elágazás és ciklus).

Összes pont: 6 pont

Programtervezési ismeretek, vizsga, MINTA

Beugró feladat: Definiálja a következőket: **a mutató (pointer) absztrakt adattípus**, **vezérlőgráf lebontása**.

Definíció. (Mutató (pointer) absztrakt adattípus) A mutató absztrakt adattípus valamely adatra mutat a memóriában azáltal, hogy az adat memóriabeli címét (byte-sorszám) tárolja. Adattal együtt van értelme használni. Ha nem mutat adatra, akkor NIL mutatónak nevezzük. A mutatóhoz hozzátartozik az a típus is, amilyen típusú adatra mutat.

Definíció. (A vezérlőgráf lebontása) A vezérlőgráf lebontásának nevezzük azt az eljárást, melynek során a struktúrált alapszerkezetek valamelyikét egy függvény csomóponttal helyettesítjük, és ezt mindaddig folytatjuk, amíg ez lehetséges. Az egyetlen csomópontból álló gráfot egyetlen éllel helyettesítjük. (Az egyetlen irányított él neve üres program.)

(Sikeres. — Nem sikeres.)

1. Írja le a pozitív valós számokon megadott négyzetgyökvonási algoritmust (Newton iteráció) és bizonyítsa be, hogy valóban működik!

   Négyzetgyökvonás iterációval
   NÉGYZETGYÖK (s, @z)
   1. Input paraméter: $s \in \mathbb{R}$, $s \geq 0$
   2. Output paraméter: $z \in \mathbb{R}$, $z \geq 0$
   3. $z \leftarrow 1$
   4. **REPEAT** $x \leftarrow z$
   5. $\quad z \leftarrow (x + s/x)/2$
   6. $\quad$ **UNTIL** $|z - x| < \varepsilon$
   7. **RETURN** (z)

   Feladat: határozzuk meg egy pozitív s valós számra az $x = \sqrt{s}$ számot, azaz a szám négyzetgyökét valamilyen előre megadott $\varepsilon > 0$ pontossággal. A bemutatandó megoldó algoritmus egy elvileg tetszőleges pozitív számból kiindulva számsorozatot képez, amely meglehetősen gyorsan konvergál a végeredményhez.

   (a) Választunk egy tetszőleges pozitív x_0 valós számot és legyen $k = 0$. (Az $x_0 = 1$ mindig megfelelő, csak esetleg a kapott sorozat kezdetben lassan kezd közelíteni a megoldáshoz.)

   (b) Képezzük az $x_{k+1} = \frac{1}{2} \cdot \left(x_k + \frac{s}{x_k}\right)$ számot és k értékét eggyel növeljük.

   (c) Ha $|x_{k+1} - x_k| < \varepsilon$, akkor megállunk és $\sqrt{s} \approx x_k$, egyébként pedig folytatjuk a 2-es pontnál.

Első lépésként azt mutatjuk meg, hogy a sorozat határértéke valóban $\sqrt{s}$, ha a sorozat konvergál. Tegyük fel tehát a konvergenciát és legyen $\lim\limits_{k\to\infty} x_k = x^*$. Akkor az $x_{k+1} = \frac{1}{2} \cdot \left(x_k + \frac{s}{x_k}\right)$ iterációs formulában mindkét oldalon elvégezve a határátmenetet k szerint az alábbi adódik.

$$\lim_{k\to\infty} x_{k+1} = \lim_{k\to\infty} \frac{1}{2} \cdot \left(x_k + \frac{s}{x_k}\right).$$

Innen kapjuk, hogy

$$x^* = \frac{1}{2} \cdot \left(x^* + \frac{s}{x^*}\right).$$

Átrendezve a formulát az $x^* = \dfrac{s}{x^*}$ alakra jutunk, amelynek megoldása éppen a $x^* = \sqrt{s}$. Ezzel megállapíthatjuk, hogy amennyiben konvergál az eljárás, akkor valóban a keresett értékhez konvergál.

Az elv az lesz, hogy alkalmazzuk a tételt, amely szerint monoton csökkenő és alulról korlátos sorozatnak mindig van határértéke. Ki kell tehát mutatnunk először, hogy a sorozat alulról korlátos. Mivel $x_{k+1} = \frac{1}{2} \cdot \left(x_k + \frac{s}{x_k}\right)$ két pozitív szám számtani átlaga és a nemnegatív számok számtani közepe mindig nagyobb, mint a mértani közép, legfeljebb egyenlő lehet vele, ha a két szám megegyezik, ezért írhatjuk, hogy $x_{k+1} = \frac{1}{2} \cdot \left(x_k + \frac{s}{x_k}\right) \geq \sqrt{x_k \cdot \frac{s}{x_k}} = \sqrt{s}$. Tehát a sorozat minden tagja (kivéve esetleg az elsőt) nagyobb, mint $\sqrt{s}$, vagy egyenlő vele, vagyis a $\sqrt{s}$ egy alsó korlát. (A sorozat első véges sok tagjának a viselkedése a konvergenciát nem befolyásolja, legfeljebb elhagyjuk őket a sorozatból.)

Képezzük két egymást követő tag különbségét.

$x_{k+1} - x_k = \frac{1}{2} \cdot \left(x_k + \frac{s}{x_k}\right) - x_k = \frac{s}{2x_k} - \frac{x_k}{2} = \frac{s-x_k^2}{2x_k} = \frac{(\sqrt{s}+x_k)(\sqrt{s}-x_k)}{2x_k} \leq 0$, mert a nevező pozitív, a számlálóban az első tényező pozitív, a második pedig negatív, hiszen $\sqrt{s}$ alsó korlát. Emiatt $x_{k+1} - x_k \leq 0$, azaz $x_{k+1} \leq x_k$, ami a monoton csökkenést jelenti.

(1+2 pont)

2. Fogalmazza meg a programgráf ciklikus bonyolultságának tételét és bizonyítsa is be!

Tétel. *A programgráf ciklikus bonyolultságának tétele A P programgráf ciklikus bonyolultsága kiszámítható a következő módon is: $m(P) = p + 1$, ahol p a programgráf predikátumainak a száma.*

Bizonyítás: Jelölje a P programgráf predikátumainak a számát p, a függvények (szekvenciák) számát f és a gyűjtők számát g! Az élek száma továbbra is e, a csúcsok száma c legyen! Ezen jelölések mellett előbb belátjuk a következő **állítást**.

Állítás:

1. $e = 3p + f + 1$,

2. $g = p$.

Az állítás belátása egyszerű. A csúcsok száma nyilvánvalóan $c = p+f+g$. Az élek számát kétféleképpen fogjuk meghatározni. Egyszer vesszük az egyes csomópontokba bevezető éleket, amelyekhez még hozzávesszük a STOP-ba vezető élt, a programgráf kimenő élét, majd másodszor vesszük a csomópontokból kivezető éleket és hozzávesszük a START-ból induló kezdőélt, a programgráf bemenő élét. Eszerint egyrészt $e = p+f+2g+1$, másrészt $e = 2p + f + g + 1$.

(1+2 pont)

3. Írja le a bináris keresés algoritmusát pszeudokóddal!

//—— Bináris_keresés procedúra ——
//Bináris keresés valós számok rendezett tömbjében nyílt intervallumos módszerrel.
Bináris_keresés(@**A**, *Elem*, @*Index*)
Input paraméter: $\mathbf{A} \in \mathbb{R}^n$, //a valós számok tömbje, $a_i \leq a_j$,
//ha $i < j$
$\mathit{Elem} \in \mathbb{R}$//a keresett elem, valós szám
Output paraméter: $\mathit{Index} \in \mathbb{N}$ //a megtalált elem indexe,
//zérus, ha nincs meg
Használt lokális változók: $\mathit{alsó}, \mathit{felső} \in \mathbb{N}$ //keresési
//indexintervallum alsó és felső határai
$\mathit{középső} \in \mathbb{N}$ //vizsgált elem indexe
$\mathit{Nincs\_meg} \in \mathbf{L}$ //*igaz*, ha még nem, *hamis*, ha már megtaláltuk
1. $\mathit{alsó} \leftarrow 0$
2. $\mathit{felső} \leftarrow \mathrm{Hossz}[\mathbf{A}] + 1$
3. $\mathit{középső} \leftarrow \left\lfloor \frac{\mathit{alsó}+\mathit{felső}}{2} \right\rfloor$
4. $\mathit{Nincs\_meg} \leftarrow \mathit{igaz}$
5. **WHILE** ($\mathit{alsó} < \mathit{középső}$) AND $\mathit{Nincs\_meg}$ **DO**
6. $\quad\quad$ **IF** $\mathit{Elem} = a_{\mathit{középső}}$
7. $\quad\quad\quad$ **THEN** $\mathit{Index} \leftarrow \mathit{középső}$
8. $\quad\quad\quad\quad\quad$ $\mathit{Nincs\_meg} \leftarrow \mathit{hamis}$
9. $\quad\quad\quad$ **ELSE** **IF** $\mathit{Elem} > a_{\mathit{középső}}$
10. $\quad\quad\quad\quad\quad$ **THEN** $\mathit{alsó} \leftarrow \mathit{középső}$
11. $\quad\quad\quad\quad\quad$ **ELSE** $\mathit{felső} \leftarrow \mathit{középső}$
12. $\quad\quad$ $\mathit{középső} \leftarrow \left\lfloor \frac{\mathit{alsó}+\mathit{felső}}{2} \right\rfloor$
13. **IF** $\quad$ $\mathit{Nincs\_meg}$
14. $\quad\quad$ **THEN** $\mathit{Index} \leftarrow 0$
15. **RETURN**(Index)
//—— Bináris_keresés procedúra vége ——

(2 pont)

Összes pont: 8 pont [0-3p elégtelen(1); 4p elégséges(2); 5p (közepes); 6p (jó); 7-8p jeles(5)]