2 operandusos művelet szükséges

Ez a művelet olyan kell, h. legyen, h. legyen értelme elvégezni.

A művelet: ⊛

$\underbrace{a_1 \circledast a_2 \circledast a_3 \circledast \dots a_n}$

— Ha ⊛ a „+" műv, akkor az eredmény → az összegzés tétele az összeget rendeli
— Ha ⊛ a „·" → a szorzatot rendeli
— Ha ⊛ a logikai v. → kétoperandusos műv.
— ⊛ : stringek összefűzése: konkatenáció: egyesíti a stringeket

leírónyelves mego.:

ÖSSZEADÁSKOR (Ha ⊛ = „+")

```
S:= Ø ;
CIKLUS J:= 1..N.
  S:= S+A[J]; → A sorozat aktuális eleme
CIKLUS VÉGE
```

Ha ⊛ = „·" ⇒ S:=Ø és az „+" helyett „·"-t ~~tett~~ kell írni

Ha ⊛ összefűzés ⇒ S:= üres stringlánc

Ha ⊛ VAGY ⇒ S:= ~~IGAZ~~ HAMIS

Ha ⊛ ÉS ⇒ S:= IGAZ

XOR esetén nem tudjuk értelmezni az összegzés tételét.

Csak ott lehet értelmezni, ahol v. neutrális eleme.

+ : 0 · : 1 ÉS : IGAZ

VAGY: HAMIS

3. előadás

pl.:

$$\sqrt{2+\sqrt{2+\sqrt{2}}}\ \ldots$$

$$\underbrace{\sqrt{2+\underbrace{\sqrt{2+s}}_{s}}}_{s}\ \ldots$$

$s := \emptyset$

ciklus $J := 1 .. N$

$s := \sqrt{2+s}$;

Azonosítók: pl.: kisz., vonalkód
↓
figyelembe kell venni az esetleges számítógépes hibákat

Adóazonosító ellenőrző algoritmusa:

magánszemélyeknek 10 számjegyű, mindig 8-cal kezdődik, és a 10. számjegy az ellenőrző.

1		10.

Nem kell, h. tároljuk a sorozat elemeit

Vesszük az első 9 jegyet, megszorozzuk a sorszámával, (8-at 1-gyel) és összeadjuk

Az utolsó jegy pedig a 11-gyel való osztás (összeg) egész maradéka.

Algoritmus:

$s := \emptyset$

ciklus $J := 1 .. 9$

$s := S + A[J] * J$

c vége

$A \to B$ (sorozat) $B[J] := A[J] * J$

$b_i := i * a_i$

EAN 13 jegyű

Európai Árucikkek Nszám

Az A[J]-ből és J-ből képzünk egy szorzatot, és a 2-vel való osztás maradéka szerint szorozzuk meg.

B[J] := A[J](J mod 2)

A 13 számjegyet hozzá kell adni az így képzett számhoz, hogy 10-zel osztható legyen.

B := (S MOD 11) = A[10] → Adóaz. jel -nél

A B akkor igaz, ha az S/11 maradéka a szám 10. eleme.

S := ∅
F[∅] := 3; F[1] := 1;
ciklus J := 1 .. 12

S := S + A[J] * F[J mod 2]

ha J mod 2 = 1 akkor
S := S + A[J]
különben
S := S + 3 * A[J];

(Ha az index páratlan, akkor F[1]-gyel kell szorozni, ha páros, akkor F[0]-val.)

ciklus vége

S := S MOD 1∅; → 10-zel oszthatónak kell lenni

(Feladat: generáljuk a hibákat!)

S + A[13]

B := ((S + A[13]) MOD 1∅) = ∅ ha nem igaz, hibás a vonalkód

Sorozathoz 1 értéket rendeltünk.

A sorozat elemei és a hozzárendelt érték azonos típusú

a.) SOROZAT → ÉRTÉK

b.) SOROZAT → RÉSZSOROZATA

c.) SOROZAT → PERMUTÁCIÓJA

rendezetlen sorozatot sorba kell rendezni.

a.) Adva van egy n elemű A sorozat, és a sorozat elemein értelmezett T tulajdonság.

pl.: számsorozatok esetében: prím, páros, páratlan

prím : $\mathbb{Z}^+$

ELDÖNTÉS TÉTELE:

Akkor igaz, ha az $\langle a_n \rangle$ $\exists$ T tul. eleme, és hamis ha nincs.

Ha a sorozatban találtam T tul.-ú elemet ⇒ nem kell tovább vizsgálni.

KONSTANS N : elemszám

VÁLTOZÓ A : TÖMB [1.. N] elemtípus

J : egész

B : logika

ALG

J := 1

CIKLUS $(J \le N) \wedge \overline{A[J]\ T\ \text{tulajdonságú}}$ ← állítás ellenkezőjét jelenti, ~~[illegible]~~ negált

↑ ROBI

J := J+1;

C VÉGE;

B := (J ≤ N);

T tulajdonságnál: kíváncsiak lehetünk, hányan vizsgáznak 5-re algo-ból.?

MEGSZÁMLÁLÁS TÉTELE:

T tul-hoz olyan egész értéket rendelünk, amely megmutatja, hogy hány T tulajdonságú elem van.

ALG.

```
    S:=0
    CIKLUS    J:=1..N
        HA (A[J] T tul.) AKKOR  S:=S+1;
        C.VÉGE ;
    A VÉGE;
```

logikai kifejezés

pl.: 3-mal való osztáskor

(A[J] MOD 3) = 0

Adott (an) n db T tul-ú elemével, tudjuk, hogy a sorozat-nak van T tul-ú eleme. Hat meg a T tul-ú elemek előfordulását ⇒ sorszámát.

KIVÁLASZTÁS TÉTELE:

Mielőtt az [J]-re hivatkoznánk, TUTI, hogy lesz T tul-ú.

ALG.

```
    J:=1
    CIKLUS        A[J] T-tul-ú   (felülvonással)
      J:=J+1;
    C vége;
```

Mj.: Csak annyit mondunk, h. van T tul-ú elem, de nem tudjuk, h. mennyi.

6., <u>KIVÁLOGATÁS TÉTELE:</u>

Adott n elemű <A> -hoz T tul-ú elemeit kadeli ⇒ Kimenet <A> részsorozata.

2 sorozat → 2 vektor

I. KONSTANS N = elemtíp

VÁLTOZÓ : A, B : tömb [1.. N] elemtíp

J, I : egész

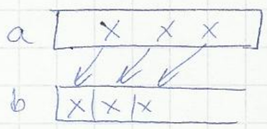
ALG: I := ∅;
ciklus J := 1 .. N

HA A[J] T tul AKKOR
I := I+1;
B[I] := A[J];

H VÉGE;

C VÉGE;

A két tömb sok helyet foglal!



II. Módosítjuk a dekl.-t: A : TÖMB [1.. N] elemtíp;

B : TÖMB [1.. N] egész;

ALG

I := ∅

CIKLUS J := 1.. N

HA A[J] T tul AKKOR

I := I+1;

B[I] := J → Nem az elemet tároljuk, hanem sorszámát.

```
      u vége;

      c vége;
```

I.

```
CIKLUS  J := 1 .. J
        KI : B[J];
   C VÉGE;
```

Kiíratjuk a T tul-iakat.

II.

```
   CIKLUS J := 1 .. J
        KI : A[B[J]];
      C VÉGE;
```

4. előadás

11.26.

MAX-imum és MIN-imum kiválasztás tétele

Adott n elemű <a>. Ez az algoritmus a sorozathoz a max.-ill min.-is értékű elem egy előfordulását rendeli. Meg kell említeni, hogy a sorozat elemeit milyen halmazból választjuk. A halmazban legyen értelmezve benne a rendezési reláció.

a,
```
MAX := A[1]
  CIKLUS J:=2..N
    HA MAX < A[J]   AKKOR  MAX:= A[J];
  ciklus vége
```

b,
```
MAXI:=1;
  CIKLUS J:=2..N;
    HA  A[MAXI] < A[J]  akkor MAXI:=
          ↳ A sorozat max i-dik eleme
  CIKLUS VÉGE;
```

	X		X		X

ha megengedjük az egyenlőséget, igaz lenne az azonos elemeknél, és végrehajtja az utasítást. ⇒ sokszor feleslegesen végrehajtja az utasítást (MAX:= A[J]);

Az a, változatnál ne engedjük meg, hogy = legyen, mert lassítja a programot.

b, Ha nem engedjük meg az =-et, az első előfordulást találja meg. Ha megengedjük, akkor az utolsó előfordulásra mutat.

Adott az $\langle a_n \rangle$, aminek nem úgy akarjuk megtalálni az utolsó előfordulást, hogy meg I engedem az = -et.

A max-ság is lehet a t tulajdonság.

<u>Kiv. tétel.</u>

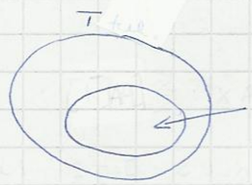
$J := 1$

$A[J] = E \leftarrow$ spec. tul. lehet $A[J] = MAX$

CIKLUS $(J \leq N) \wedge (\overline{A[J]\ t\text{-tul.}})$

$J := J + 1;$

CIKLUS VÉGE;



E: Ez a részhalmaz a T halmazon belül az ÉRTÉK.

Vagy úgy hamis a ciklus, hogy J túllép n-nen, vagy nem adott a tul.

A $J \leq N$-t kiküszöbölhetjük, mert biztos van

Ha tudjuk, hogy van benne T tul-ú elem, akkor az nem kiválasztás tétele, hanem keresés tétele.

Lin keresés: Adott $\langle a_n \rangle$, és az $\langle a_n \rangle$-en értelmezett T tulajdonság. A algoritmus a sorozathoz egy T tulajdonságú elem helyét rendeli.

$J := 1$

CIKLUS $(J \leq N) \wedge (\overline{A[J]\ T\text{tul}})$

$J := J + 1;$

CIKLUS VÉGE;

$B := (J \leq N)$

És az algoritmus is: Ha a sorozatnak több T tulajdonságú eleme van, az elsőt fogja megtalálni.

Ez a kiválasztás tétele.

Eldöntés:

$\{a_n\} \rightarrow$ log kif.

$(i \leq n) \wedge (a_i$ nem T tul$)$

Kiválasztás

$\{a_n\} \rightarrow$ egész

$(a_i$ nem T tul.$)$

$\Rightarrow$ ez a gyorsabb!

KERESÉS

$\{a_n\}$ log kif, egész

felt: $(i \leq n) \wedge (a_i$ nem T tul.$)$

Kéne egy gyorsabb kereső algoritmust találni. Ha tudjuk, hogy van $\langle a_n \rangle$-nek T tul eleme $\Rightarrow$ akkor végig ki $(i \leq n)$ teljesül.

$A[N+1] := T$ tul. elem

$\downarrow$

(hozzáelem)

Ha előbb nem, az $(n+1)$. helyen megtalálja a T tul.-ú elemet.

Kéne 1 logikai érték, mert nem mindegy hol találja meg az elemet.

Ha az eredeti $\langle a_n \rangle$-nek volt T tul.ú eleme, akkor $i \leq n$, tehát $\exists$ igaz. Ha nem volt a sorozatnak T. tul-ú eleme $\exists$ hamis. $i \nleq n \Rightarrow i = n+1$