

Egyszerű programozási tételek

2. előadás

Sergyán Szabolcs

`sergyan.szabolcs@nik.uni-obuda.hu`

Óbudai Egyetem

Neumann János Informatikai Kar

2011. szeptember 15.



- Szlávi Péter, Zsakó László: Módszeres programozás: Programozási tételek (Mikrológia 19). ELTE TTK, 2002



- A programozási tételek jól megválasztott, egyszerű feladatok megoldásai
 - Segítségükkel a gyakorlatban szükséges feladatok jelentős része megoldható
 - Helyességük egyszerűen bizonyítható
 - Használatuk célszerű, hiszen (mások által is) jól áttekinthető kódot eredményeznek
- Egy lehetséges csoportosításuk
 - Egy sorozathoz egy értéket rendelő feladatok
 - Egy sorozathoz egy sorozatot rendelő feladatok
 - Egy sorozathoz több sorozatot rendelő feladatok
 - Több sorozathoz egy sorozatot rendelő feladatok
- Feldolgozandó intervallum alapján megkülönböztetünk
 - Rögzített intervallumos programozási tételeket
 - Feltételig tartó programozási tételeket (ezeket a változatokat nem tárgyaljuk)



- Azonos típusú adatokat tárolunk egymás mellett
 - Tömböt alkotó adatok az **elemek**
 - Az elemekre a tömb **indexével** hivatkozunk
 - Jelölés: $A[i]$
-
- Ha a tömbnek egy indexe van, akkor egydimenziós tömb: vektor
 - Ha a tömbnek két indexe van, akkor két dimenziós tömb: mátrix
 - Index – egész számok sorozata (nullától) vagy egytől
 - Tárolás – sorfolytonosan, elemek egymást követően



Tömbök feldolgozása

- A tömbök feldolgozása általában ciklusokkal történik
- Ha a tömb minden egyes elemét feldolgozzuk: számláló ciklus
- Ha a tömb elemein valamilyen tulajdonság meglétét vizsgáljuk: elől- vagy hátultesztelő ciklus



Egyszerű programozási tételek

- 1 Sorozatszámítás
- 2 Eldöntés
- 3 Kiválasztás
- 4 Lineáris keresés
- 5 Megszámlálás
- 6 Maximumkiválasztás



Egyszerű programozási tételek

- 1 Sorozatszámítás
- 2 Eldöntés
- 3 Kiválasztás
- 4 Lineáris keresés
- 5 Megszámlálás
- 6 Maximumkiválasztás



Típusfeladatok

- 1 Egy osztály N darab tanulójának osztályzata alapján adjuk meg az osztály átlagát.
- 2 Egy M elemű betűsorozat betűit fűzzük össze egyetlen szöveg típusú változóba.
- 3 Készítsünk algoritmust, amely egy autóversenyző körönkénti ideje alapján meghatározza a versenyző egy kör megtételéhez szükséges átlagidejét.
- 4 A Balaton mentén K darab madarász végzett megfigyeléseket. Mindegyik megadta, hogy milyen madarakat látott. Készítsünk algoritmust, amely a megfigyelések alapján megadja a Balatonon előforduló madárfajokat.
- 5 Adjuk meg az első N természetes szám szorzatát.



Bemenet

A : Feldolgozandó tömb

N : Tömb elemeinek száma

Kimenet

R : Művelet eredménye

Pszudokód

Eljárás Sorozatszámítás(A , N , R)

$R := R_0$

Ciklus $i := 1$ -től N -ig

$R := R$ művelet $A[i]$

Ciklus vége

Eljárás vége



Megjegyzések

- Adatok sorozatához egy értéket rendelő függvényt helyettesít
- Minden olyan esetben használható, ha ezt a függvényt felbonthatjuk értékpárokon kiszámított függvények sorozatára
- Az induláskor használt nullértéket értelemszerűen a kérdéses függvény (esetleg a feladat) alapján kell megválasztani

	összegzés	faktoriális	elemek uniója
R_0	0	1	{ }
művelet	$R := R + A[i]$	$R := R * A[i]$	$R := R \cup A[i]$



Összegzés

Eljárás Összegzés(N, X, S)

$S := 0$

Ciklus $i = 1$ -től N -ig

$S := S + X[i]$

Ciklus vége

Eljárás vége



Egyszerű programozási tételek

- 1 Sorozatszámítás
- 2 Eldöntés**
- 3 Kiválasztás
- 4 Lineáris keresés
- 5 Megszámlálás
- 6 Maximumkiválasztás



Típusfeladatok

- 1 Döntsük el egy számról, hogy prímszám-e.
- 2 Döntsük el egy szóról a hónapnevek sorozata alapján, hogy egy hónap neve-e.
- 3 Döntsük el egy tanuló év végi osztályzatai alapján, hogy kitűnő tanuló-e.
- 4 Júniusban minden nap délben megmértük, hogy a Balaton Siófoknál hány fokok. Döntsük el a mérések alapján, hogy a víz hőfoka folyamatosan emelkedett-e.



Bemenet

- A : Feldolgozandó tömb
- N : Tömb elemeinek száma
- T : Tulajdonság függvény

Kimenet

- VAN : Logikai változó

Pszudokód

Eljárás $Eldöntés(A, N, T, VAN)$

$i := 1$

Ciklus amíg ($i \leq N$) és $\neg(A[i]$ teljesíti T -t)

$i := i + 1$

Ciklus vége

$VAN := (i \leq N)$

Eljárás vége

Megjegyzések

- A T tulajdonság helyes megválasztásával a tétel sokféle szituációban alkalmazható
- A "minden elem T tulajdonságú" feladatot egyszerűen visszavezethetjük az eldöntésre a T tulajdonság tagadásával
- A sorozatszámításnál megismert módszerrel ellentétben ez az algoritmus az első T tulajdonságú elem megtalálása után már nem folytatja a keresést



Prím teszt

Eljárás Eldöntés(N , $PRIM$)

$i := 2$

Ciklus amíg ($i \leq N - 1$) és $\neg(i$ osztója N -nek)

$i := i + 1$

Ciklus vége

$PRIM := (i > N - 1)$

Eljárás vége



Monoton növekedés

Eljárás Eldöntés(N , X , $MONOTON$)

$i := 1$

Ciklus amíg ($i \leq N - 1$) és ($X[i] \leq X[i + 1]$)

$i := i + 1$

Ciklus vége

$MONOTON := (i > N - 1)$

Eljárás vége



Egyszerű programozási tételek

- 1 Sorozatszámítás
- 2 Eldöntés
- 3 Kiválasztás**
- 4 Lineáris keresés
- 5 Megszámlálás
- 6 Maximumkiválasztás



Típusfeladatok

- 1 Ismerjük egy hónap nevét, a hónapnevek sorozata alapján mondjuk meg a sorszámát.
- 2 Adjuk meg egy egynél nagyobb természetes szám legkisebb, 1-től különböző osztóját.
- 3 A naptárban található névnapok alapján adjuk meg a legjobb barátunk névnapját.



Bemenet

A : Feldolgozandó tömb
 N : Tömb elemeinek száma
 T : Tulajdonság függvény

Kimenet

$SORSZ$: Első T tulajdonságú
elem indexe

Pszudokód

Eljárás Kiválasztás(A , N , T , $SORSZ$)

$i := 1$

Ciklus amíg $\neg(A[i]$ teljesíti T -t)

$i := i + 1$

Ciklus vége

$SORSZ := i$

Eljárás vége

Megjegyzések

- Az eldöntéssel ellentétben ez visszaadja az első T tulajdonságú elem sorszámát
- A tétel feltételezi, hogy **biztosan van legalább egy ilyen tulajdonságú elem**
- Sorszám helyett visszaadhatjuk az elem értékét is, de célszerűbb a sorszám használata (ez alapján az elem is egyszerűen meghatározható)



Legjobb barát névnapja

Eljárás Kiválasztás(N , X , $BARAT$, NAP)

$i := 1$

Ciklus amíg ($X[i].nev \neq BARAT$)

$i := i + 1$

Ciklus vége

$NAP := X[i].nevnap$

Eljárás vége



Egyszerű programozási tételek

- 1 Sorozatszámítás
- 2 Eldöntés
- 3 Kiválasztás
- 4 Lineáris keresés**
- 5 Megszámlálás
- 6 Maximumkiválasztás



Típusfeladatok

- 1 Ismerjük egy üzlet egy havi forgalmát: minden napra megadjuk, hogy mennyi volt a bevétel és mennyi a kiadás. Adjunk meg egy olyan napot – ha van –, amelyik nem volt nyereséges.
- 2 A Budapest-Nagykanizsa vasúti menetrend alapján két adott állomáshoz adjunk meg egy olyan vonatot, amellyel el lehet jutni átszállás nélkül az egyikről a másikra.
- 3 Egy tetszőleges (nem 1) természetes számnak adjuk meg egy osztóját, ami nem az 1 és nem is önmaga.



Bemenet

A : Feldolgozandó tömb
 N : Tömb elemeinek száma
 T : Tulajdonság függvény

Kimenet

VAN : Logikai változó
 $SORSZ$: Első T tulajdonságú elem indexe

Pszudokód

Eljárás Keresés(A , N , T , VAN , $SORSZ$)

$i := 1$

Ciklus amíg ($i \leq N$) és $\neg(A[i]$ teljesíti T -t)

$i := i + 1$

Ciklus vége

$VAN := (i \leq N)$

Ha VAN akkor

$SORSZ := i$

Elágazás vége

Eljárás vége

Megjegyzések

- Tekinthető az eldöntés és a kiválasztás tétel ötvözetének is: választ ad arra, hogy van-e T tulajdonságú elem a sorozatban, és ha van, akkor visszaadja a sorszámát is
- Értelemszerűen így nem feltételezi, hogy biztosan van ilyen elem a listában. Ha nincs, akkor a *VAN* változó értéke hamis, ilyenkor a *SORSZ* változó nem kap értéket.



Nem nyereséges nap

Eljárás Keresés(K , B , N , VAN , $SORSZ$)

$i := 1$

Ciklus amíg ($i \leq N$) és ($B[i] - K[i] > 0$)

$i := i + 1$

Ciklus vége

$VAN := (i \leq N)$

Ha VAN akkor

$SORSZ := i$

Elágazás vége

Eljárás vége



Egyszerű programozási tételek

- 1 Sorozatszámítás
- 2 Eldöntés
- 3 Kiválasztás
- 4 Lineáris keresés
- 5 Megszámlálás**
- 6 Maximumkiválasztás



Típusfeladatok

- 1 Családok létszáma, illetve jövedelme alapján állapítsuk meg, hogy hány család él a létminimum alatt.
- 2 Egy futóverseny időeredményei alapján határozzuk meg, hogy a versenyzők hány százaléka teljesítette az olimpiai induláshoz szükséges szintet.
- 3 Adjuk meg egy szöveg magánhangzóinak számát.



Megszámlálás

Bemenet

- A : Feldolgozandó tömb
- N : Tömb elemeinek száma
- T : Tulajdonság függvény

Kimenet

- DB : T tulajdonságú elemek száma

Pszudokód

Eljárás Megszámlálás(A , N , T , DB)

$DB := 0$

Ciklus $i := 1$ -től N -ig

Ha ($A[i]$ teljesíti T -t) akkor

$DB := DB + 1$

Elágazás vége

Ciklus vége

Eljárás vége

Megjegyzések

- Amennyiben nincs T tulajdonságú elem a listában, akkor értelemszerűen 0 kerül a DB változóba
- Valójában egy sorozatszámítás, amely minden T tulajdonságú elem esetén 1-et hozzáad a DB értékéhez



Olimpiai indulási szintet teljesítők százalékos aránya

Eljárás Megszámlálás(N , ID , $SZINT$, $SZAZ$)

$DB := 0$

Ciklus $i := 1$ -től N -ig

Ha ($ID[i] \leq SZINT$) akkor

$DB := DB + 1$

Elágazás vége

Ciklus vége

$SZAZ := Kerekit(100 * DB / N)$

Eljárás vége



Egyszerű programozási tételek

- 1 Sorozatszámítás
- 2 Eldöntés
- 3 Kiválasztás
- 4 Lineáris keresés
- 5 Megszámlálás
- 6 Maximumkiválasztás



Típusfeladatok

- 1 Egy kórházban megmérték minden beteg lázát. Adjuk meg, hogy ki a leglázasabb.
- 2 Egy család havi bevételei és kiadásai alapján adjuk meg, hogy melyik hónapban tudtak a legtöbbet megtakarítani.
- 3 Egy osztály tanulóinak nevei alapján adjuk meg a névsorban legelső tanulót.



Maximumkiválasztás

Bemenet

A : Feldolgozandó tömb

N : Tömb elemeinek száma

Kimenet

MAX : Maximális elem indexe

Pszudokód

Eljárás Maximumkiválasztás(A , N , MAX)

$MAX := 1$

Ciklus $i := 2$ -től N -ig

Ha $A[i] > A[MAX]$ akkor

$MAX := i$

Elágazás vége

Ciklus vége

Eljárás vége

Megjegyzések

- Reláció megfordításával értelemszerűen minimumkiválasztás lesz a tétel célja
- Sorszám helyett visszaadhatjuk az elem értékét is, de célszerűbb a sorszám használata (ez alapján az elem is azonnal meghatározható)
- Feltételezzük, hogy legalább egy elem létezik a listában
- Több maximális elem esetén az elsőt adja vissza



Névsorban legelső tanuló

Eljárás Minimumkiválasztás(X, N, MIN, NEV)

$MIN := 1$

Ciklus $i := 2$ -től N -ig

Ha $X[MIN] > X[i]$ akkor

$MIN := i$

Elágazás vége

Ciklus vége

$NEV := X[MIN]$

Eljárás vége

