10. előadás

Programozás-elmélet

Sorozathoz sorozatot rendelő feladatokkal foglalkozunk.
A bemenő sorozatot le kell másolni, s közben az elemekre vonatkozó átalakításokat lehet végezni rajta:

Input	:	$n \in \mathbb{N}_0,\ x \in H^n,\ f : H \to G$
Output	:	$y \in G^n$
Q	:	$--$
R	:	$\forall i\ (1 \leq i \leq n) : y_i = f(x_i)$

Az algoritmus:

Függvény:

$f : H - elemtípus \to G - elemtípus$

Változók:

n : egész {a feldolgozandó sorozat elemszáma}
x : tömb(1..n:H-elemtipus) {a feldolgozandó sorozat elemei}
y : tömb(1..n:G-elemtípus) {a feldolgozott sorozat}

A megoldás:

másolás(n, x, y)
  for $i = 1 : n$
    $y(i) := f(x(i))$
  end
eljárás vége

Példa: Egy szöveg minden magánhangzóját cseréljük ki az *e* betűre!

Meg kell adni egy sorozat adott tulajdonságú elemeinek számát és indexeit:

Input	:	$n \in \mathbb{N}_0,\ x \in H^n,\ T : H \to \mathbb{L},\ \chi : H \to \{0,1\}$
		$\chi(x) = 1$, ha $T(x)$ és $\chi(x) = 0$, ha $\neg T(x)$
Output	:	$DB \in \mathbb{N}_0,\ y \in [1..n]^n$
Q	:	–
R	:	$DB = \sum_{i=1}^{n} \chi(x_i) \wedge y \subset [1..n] \wedge$
		$\wedge\ \forall i\ (1 \leq i \leq DB) : T(x_{y_i})$

Figyeljük meg, hogy az y tömb hossza előre nem meghatározható, de legfeljebb n.

Az algoritmus:

Függvény:

T : *elemtípus* $\to \mathbb{L}$

Változók:

n : egész {a feldolgozandó sorozat elemszáma}
x : tömb(1..n:elemtipus) {a feldolgozandó sorozat elemei}
DB : egész {a megfelelő elemek száma}
y : tömb(1..n:egész) {a megfelelő elemek sorszámai}

A megoldás:

kiválogatás(n, x, DB, y)
  $DB := 0$
  for $i = 1 : n$
    if $T(x(i))$ **then**
      $DB := DB + 1$
      $Y(DB) = i$
    end
  end
eljárás vége

Feladat: Módosítsuk a feladatot és az algoritmust, hogy ne a sorszámokat, de magukat az elemeket gyüjtse ki.

A feladat egy sorozat elemeinek szétválogatása két részsorozatba, ahol az egyik sorozatban adott tulajdonságú elemek, míg a másikban az adott tulajdonsággal nem rendelkező elemek vannak:

Input	:	$n \in \mathbb{N}_0,\ x \in H^n,\ T : H \to \mathbb{L},\ \chi : H \to \{0,1\}$
		$\chi(x) = 1$, ha $T(x)$ és $\chi(x) = 0$, ha $\neg T(x)$
Output	:	$DBY, DBZ \in \mathbb{N}_0,\ y, z \in H^n$
Q	:	–
R	:	$DBY = \sum_{i=1}^{n} \chi(x_i) \wedge y \subset x \wedge \forall i\ (1 \leq i \leq DBY) : T(y_i) \wedge$
		$\wedge\ DBZ = n - DBY \wedge z \subset x \wedge \forall i\ (1 \leq i \leq DBZ) : \neg T(z_i)$

Az algoritmus:

Függvény:

T : *elemtípus* $\rightarrow \mathbb{L}$

Változók:

n : egész {a feldolgozandó sorozat elemszáma}

x : tömb(1..n:elemtipus) {a feldolgozandó sorozat elemei}

DBY, DBZ : egész {a megfelelő sorozatok elemszámai}

y, z : tömb(1..n:elemtípus) {a megfelelő sorozatok elemei}

A megoldás:

szétválogatás(n, x, DBY, y, z, DBZ)
$DBY, DBZ := 0, 0$
for $i = 1 : n$
if $T(x(i))$ **then**
$DBY := DBY + 1$
$Y(DBY) = x(i)$
else
$DBZ := DBZ + 1$
$Z(DBZ) = x(i)$
end
end
eljárás vége

Az alapfeladat egy n elemszámú sorozat nagyság szerinti sorbarendezése. Ez feltételezi, hogy a sorozat elemeire létezik a $\leq, <$ reláció. Számos megoldó algoritmus létezik és ezekkel külön terület (tantárgy) foglalkozik.

Input	:	$n \in \mathbb{N}_0,\ x \in H^n$
Output	:	$x \in H^n$
Q	:	–
R	:	x_{ki} rendezett és $x_{ki} = permutáció\,(x_{be})$

Az algoritmus:

Változók:

n : egész {a feldolgozandó sorozat elemszáma}

x : tömb(1..n:elemtipus) {a feldolgozandó sorozat elemei}

Alapötlete: Hasonlítsuk össze az első elemet a sorozat összes többi mögötte levő elemével, s ha valamelyik kisebb nála, akkor cseréljük meg őket. Ezzel elérhetjük, hogy a sorozat első helyére a legkisebb elem kerül. Folytassuk ugyanígy a sorozat második elemével, stb., utoljára pedig az utolsó előttivel.

rendezés(n, x)
  for $i = 1 : n - 1$
    for $j = i + 1 : n$
      if $x(i) > x(j)$ **then** $csere(x(i), x(j))$
    end
  end
eljárás vége

Az eljárás helyigénye: $n + 1$. Az összehasonlítások száma: $n(n-1)/2$. A cserék száma 0 és $n(n-1)/2$ között van.

Az előző rendezési eljárásban sok a felesleges csere. Ehelyett az aktuális első elemet a mögötte lévők közül egyedül a legkisebbel cseréljük ki. Ehhez a rendező ciklus belsejében egy minimumkeresést kell csinálni.

rendezés(n, x)
  for $i = 1 : n - 1$
    $MIN := i$
    for $j = i + 1 : n$
      if $x(MIN) > x(j)$ **then** $MIN := j$
    end
    $csere(x(i), x(MIN))$
  end
eljárás vége

Az eljárás helyigénye: $n + 1$. Az összehasonlítások száma: $n(n-1)/2$. A cserék száma: $n - 1$.

Alapötlete:

1. Felosztás: Az $\{A(p), \ldots, A(r)\}$ tömb $(p \leq r)$ felosztása két "összefüggő" (esetleg üres)

$$\{A(p), \ldots, A(q-1)\}, \quad \{A(q+1), \ldots, A(r)\}$$

résztömbre úgy, hogy

a baloldali résztömb elemei kisebb, vagy kisebb-egyenlők $A(q)$-nál,

a jobboldali résztömb elemei pedig nagyobb vagy egyenlők $A(q)$-nál.

A q index számítása a felosztó eljárás része.

2. Uralkodás: Az $\{A(p), \ldots, A(q-1)\}$ és $\{A(q+1), \ldots, A(r)\}$ résztömböket a felosztás (gyorsrendezés) rekurzív hívásával rendezzük.

3. Összevonás: Mivel a két résztömböt helyben rendeztük, az egész $\{A(p), \ldots, A(r)\}$ tömb rendezett.

Az algoritmus leírása:

gyorsrendezés (A, p, r)
  if $p < r$
    $q =$ **feloszt**(A, p, r)
    gyorsrendezés $(A, p, q-1)$
    gyorsrendezés $(A, q+1, r)$
  end
eljárás vége

Teljes tömb rendezése: **gyorsrendezés**$(A, 1, hossz\,(A))$.

A tömb felosztása:

A **feloszt** függvényeljárás *helyben* átrendezi az $\{A\,(p)\,,\ldots,A\,(r)\}$ résztömböt.

feloszt (A, p, r)
  $x = A\,(r)$
  $i = p - 1$
  for $j = p : r - 1$
    if $A\,(j) \leq x$
      $i = i + 1$
      $csere\,(A(i), A\,(j))$
    end
  end
  $csere\,(A\,(i+1)\,, A\,(r))$
  $feloszt = i + 1$
eljárás vége

Adott egy rendezett sorozat, amelyben egy adott értékű elem sorszámát kell meghatározni, ha az benne van a sorozatban:

Input	:	$n \in \mathbb{N}, x \in H^n, y \in H$
Output	:	$VAN \in \mathbb{L}, SORSZ \in \mathbb{N}_0$
Q	:	$rendezett\,(x)$
R	:	$VAN \equiv (\exists i \ (1 \leq i \leq n) : x_i = y) \wedge$
		$\wedge VAN \Longrightarrow (1 \leq SORSZ \leq n) \wedge x_{SORSZ} = y$

Az algoritmus:

Változók:

n : egész {a feldolgozandó sorozat elemszáma}
x : tömb(1..n:elemtipus) {a feldolgozandó sorozat elemei}
y : elemtipus { a keresett elem}
VAN : logikai {az eredmény - van-e megfelelő elem}
$SORSZ$: egész {az eredmény -az elem sorszáma}

A megoldó algoritmus:

keresés$(n, x, y, VAN, SORSZ)$
  $i := 1$
  while $(i \leq n) \wedge (x(i) < y)$
    $i := i + 1$
  end
  $VAN := (i \leq n) \wedge (x(i) = y)$
  if VAN **then** $SORSZ := i$
eljárás vége

Az algoritmus az y érték első előfordulását választja ki. Egy keresés minimum 1, maximum n, átlagosan pedig $(n + 1)/2$ összehasonlítással jár.

A sorozat rendezettségét a következőképpen használhatjuk ki. Vizsgáljuk meg első lépésben a sorozat középső elemét. Ha ez a keresett elem, akkor készen vagyunk. Ha a keresett elem ennél kisebb, akkor csak az ezt megelőzőek között lehet, tehát a keresést a továbbiakban a sorozat első felére kell alkalmazni. Ha a keresett elem ennél nagyobb, akkor ugyanezen elv miatt a keresést a sorozat második felére kell alkalmazni.

keresés$(n, x, y, VAN, SORSZ)$
  $e, u := 1, n$
  do
    $k := entier\left(\left(e + u\right)/2\right)$
    if $y < x\left(k\right)$ **then** $u := k - 1$
    if $y > x\left(k\right)$ **then** $e := k + 1$
  until $(e \leq u) \wedge (x\left(k\right) \neq y)$
  $VAN := (e \leq u)$
  if VAN **then** $SORSZ := k$
eljárás vége

Az algoritmus az y érték valamelyik előfordulását választja ki. Egy keresés minimum 1, maximum $\log_2 n + 1$, átlagosan pedig $\log_2 n$ összehasonlítással jár. Innen az eljárás elnevezése.

Ha programozási tételeket egymásután kell alkalmaznunk, akkor gyakran előnyös a két megoldó algoritmus egybeépítése. Néhány esetet tárgyalunk.

A másolással bármelyik programozási tétel egybeépíthető: A programozási tételben szereplő x_i hivatkozást kicseréljük $g(x_i)$-re, ahol g a másolásnál szereplő függvényt jelöli.
1. Példa: számsorozat elemeinek négyzetösszege=másolás (négyzetre emelés)+ sorozatszámítás (összegzés)

Az általános feladat a következő:

Input	:	$n \in \mathbb{N}_0$, $x \in H^n$, $F : G^* \to G$, $g : H \to G$
Output	:	$S \in G$
Q	:	$\exists F_0 \in G$ és $\exists f : G \times H \to G$ és
		$F(y_1, \ldots, y_n) = f(F(y_1, \ldots, y_{n-1}), y_n)$ és $F() = F_0$
R	:	$S = F(g(x_1), \ldots, g(x_n))$

Az algoritmus:

Függvény:

g : H-elemtípus$\rightarrow$ G-elemtípus

Változók:

n : egész {a feldolgozandó sorozat elemszáma}

x : tömb(1..n:$H-$elemtipus) {a feldolgozandó elemek}

F_0 : G-elemtípus {a művelet nulleleme}

S : G-elemtípus {az eredmény}

A megoldás:

Másolás_sorozatszámítás(n, x, s)
  $s := F_0$
  for $i = 1 : n$
    $s := f\left(s, g\left(x\left(i\right)\right)\right)$
  end
eljárás vége

2. Példa: Másolás és maximumkiválasztás egybeépítése a maximális elem értékének és indexének meghatározásával.
A feladat leírása:

Input	:	$n \in \mathbb{N}_0$, $x \in H^n$, H rendezett halmaz $(\exists <, \leq)$, $g : H \to G$
Output	:	$MAX \in \mathbb{N}$
Q	:	$n \geq 1$
R	:	$1 \leq MAX \leq n \wedge \forall i\ (1 \leq i \leq n) : g\left(x_{MAX}\right) \geq g\left(x_i\right)$

Az algoritmus:

Függvény:

g : H-elemtípus$\rightarrow$ G-elemtípus

Változók:

n : egész {a feldolgozandó sorozat elemszáma}
x : tömb(1..n:H-elemtipus) {a feldolgozandó sorozat elemei}
MAX : egész {a maximális értékű elem sorszáma}
$MAXERT$: G-elemtípus {a maximális érték}

Másolás_maximumkiválasztás$(n, x, MAX, MAXERT)$
  $MAX, MAXERT := 1, g(x(1))$
  for $i = 2 : n$
    if $MAXERT < g(x(i))$ **then** $MAX, MAXERT := i, g(x(i))$
  end
eljárás vége