

Ég és Föld vonzásában – a természet titkai

Informatikai tehetség gondozás:

Multihalmaz típus

TÁMOP-4.2.3.-12/1/KONV



Érték-halmaz: az alaphalmaz (amely az *Elemtípus* és egy *darabszám* által van meghatározva) iteráltja („mely elem hány-szoros multiplicitással van benne a multihalmazban”).

Műveletek: + (egyesítés - unió), *max* (értékek uniója, multiplicitások maximuma), *min* (értékek uniója, multiplicitások minimuma¹), * (metszet), *mindközös?* (a két multihalmaz az elemek multiplicitásától eltekintve azonos-e), - (különbség), *Eleme* (egy elem benne van-e a multihalmazban), *multiplicitás* (egy elem hány-szoros multiplicitással van benne a multihalmazban), *Üres* (üres multihalmaz létrehozás: eljárás), vagy *Üres'Halmaztípus* előre definiált konstans, *Üres?* (logikai értékű függvény).

Relációk: = , < , ≤ , ≥ , > , ≠ (parciális rendezés: a tartalmazás, azon belül pedig az elemszám alapján).

Például:

```
Típus Fajta=Rekord(név: Szöveg, multi: Egész)
        Állomány=Multihalmaz(Fajta)
```

```
Változó A: Állomány
```

```
A:=Állomány(("nyúl", 3), ("kecske", 5))
```

Multihalmazok ábrázolása többféleképpen is megoldható. Közülük tekintsük át a legfontosabbakat:

Elemek felsorolása

```
Típus halmazelem=Rekord(érték: Elemtípus, multi: Egész)
```

```
Multihalmaz(Elemtípus)=Rekord(db: Egész,
                                elem: Tömb(1..MaxDb:Halmazelem))
```

Egy felsorolásként adjuk meg a multihalmazt, annyi elemű tömbben, ahány elemű éppen a multihalmaz (pontosabban az első Db darab elemében).

Nézzük meg a *Multihalmaz* típus néhány műveletét!

Elemek felsorolása esetén a multihalmaz műveleteket az alábbi módon valósíthatjuk meg:

```
Eljárás Beolvasás(Változó a: Multihalmaz(Elemtípus)):
```

```
  Be: a.db [a multihalmaz elemszáma]
```

```
  Ciklus i=1-től a.db-ig
```

```
    Be: a.elem[i].érték, a.elem[i].multi
```

```
  Ciklus vége
```

```
Eljárás vége.
```

Műveletigény számítása: a ciklus a multihalmaz elemértékeinek számaszor fut le, azaz a futási idő a **multihalmaz elemszámával arányos**.

¹ Ha van ennek egyáltalán értelme.

Eljárás Kiírás (**Konstans** a: Multihalmaz (Elemtípus)):
Ki: a.db [a multihalmaz elemszáma]
Ciklus i=1-től a.db-ig
 Ki: a.elem[i].érték, a.elem[i].multi
Ciklus vége
Eljárás vége.

Műveletigény számítása: a ciklus a multihalmaz elemértékeinek számaszor fut le, azaz a futási idő a **multihalmaz elemszámával arányos**.

Eljárás Üres (**Változó** a: Multihalmaz (Elemtípus)):
a.db:=0
Eljárás vége.

Műveletigény számítása: nem függ a **multihalmaz elemszámától**.

Függvény Üres? (**Konstans** a: multihalmaz (Elemtípus)):
Üres? := (a.db=0)
Függvény vége.

Műveletigény számítása: nem függ a **multihalmaz elemszámától**.

Eljárás Multihalmazba (**Változó** a: Multihalmaz (Elemtípus),
Konstans e: Elemtípus):
i:=1
Ciklus amíg i≤a.db és a.elem[i].érték≠e
 i:=i+1
Ciklus vége
Ha i≤a.db **akkor** a.elem[i].multi:=a.elem[i].multi+1
különben a.db:=a.db+1; a.elem[a.db].érték:=e
 a.elem[a.db].multi:=1
Eljárás vége.

Műveletigény számítása: arányos a **multihalmaz elemszámával**.

Eljárás Multihalmazból (**Változó** a: Multihalmaz (Elemtípus),
Konstans e: Elemtípus):
i:=1
Ciklus amíg i≤a.db és a.elem[i].érték≠e
 i:=i+1
Ciklus vége
Ha i≤a.db **akkor**
 Ha a.elem[i].multi=1 **akkor** a.elem[i]:=a.elem[a.db]
 a.db:=a.db-1
 különben a.elem[i].multi:=a.elem[i].multi-1
Eljárás vége.

Műveletigény számítása: a ciklus a multihalmaz elemeinek számaszor fut le, azaz a futási idő a **multihalmaz elemszámával arányos**.

Függvény eleme (**Konstans** e: Elemtípus,
a: Multihalmaz(Elemtípus)): Logikai
i:=1
Ciklus amíg i≤a.db és e≠a.elem[i].érték
i:=i+1
Ciklus vége
leme:=i≤a.db
Függvény vége.

Műveletigény számítása: a ciklus az A multihalmaz elemszámához fut le, azaz a futási idő a multihalmaz elemszámával arányos.

Függvény multiplicitás (**Konstans** a: Multihalmaz(Elemtípus),
e: Elemtípus): Egész
i:=1
Ciklus amíg i≤a.db és e≠a.elem(i).érték
i:=i+1
Ciklus vége
Ha i≤a.db akkor multiplicitás:=a.elem(i).multi
különben multiplicitás:=0
Függvény vége.

Műveletigény számítása: a ciklus az A multihalmaz elemszámához fut le, azaz a futási idő a multihalmaz elemszámával arányos.

Függvény benne (**Konstans** e: Halmazelem,
a: Multihalmaz(Elemtípus)): Logikai
i:=1
Ciklus amíg i≤a.db és e≠a.elem[i]
i:=i+1
Ciklus vége
benne:=i≤a.db és e.multi≤a.elem[i].multi
Függvény vége.

Műveletigény számítása: a ciklus az A multihalmaz elemszámához fut le, azaz a futási idő a multihalmaz elemszámával arányos.

Függvény része (**Konstans** a,b: Multihalmaz(Elemtípus)): Logikai
i:=1
Ciklus amíg i≤a.db és benne(a.elem[i],b)
i:=i+1
Ciklus vége
része:=i≤a.db
Függvény vége.

Műveletigény számítása: a külső ciklus az A, a benne műveletben levő belső ciklus a B multihalmaz elemszámához fut le, azaz a futási idő a két multihalmaz elemszáma szorzatával arányos.

Operátor unió(**Konstans** a,b: Multihalmaz(Elemtípus)):
 Multihalmaz(Elemtípus)

Változó c: Multihalmaz(Elemtípus)
 c:=a

Ciklus i=1-től b.db-ig
 j:=1
Ciklus amíg j≤a.db és b.elem(i)≠a.elem(j)
 j:=j+1
Ciklus vége
Ha j>a.db **akkor** c.db:=c.db+1: c.elem(c.db):=b.elem(i)
különben
 c.elem(j).multi:=c.elem(j).multi+b.elem[i].multi
Ciklus vége
 unió:=c

Operátor vége.

Függvény max(**Konstans** a,b: Multihalmaz(Elemtípus)):
 Multihalmaz(Elemtípus)

Változó c: Multihalmaz(Elemtípus)
 c:=a

Ciklus i=1-től b.db-ig
 j:=1
Ciklus amíg j≤a.db és b.elem(i)≠a.elem(j)
 j:=j+1
Ciklus vége
Ha j>a.db **akkor** c.db:=c.db+1: c.elem(c.db):=b.elem(i)
különben ha b.elem[i].multi>c.elem(j).multi **akkor**²
 c.elem(j).multi:=b.elem[i].multi

Ciklus vége
 max:=c

Függvény vége.

Operátor metszet(**Konstans** a,b: Multihalmaz(Elemtípus)):
 Multihalmaz(Elemtípus)

Változó c: Multihalmaz(Elemtípus)
 c.db:=0

Ciklus i=1-től a.db-ig
 j:=1
Ciklus amíg j≤b.db és b.elem(j)≠a.elem(i)
 j:=j+1
Ciklus vége
Ha j≤b.db **akkor**
 c.db:=c.db+1: c.elem(c.db):=a.elem(i)
Ha b.elem(j).multi<a.elem(i).multi **akkor**
 c.elem(c.db).multi:=b.elem(j).multi

Elágazások vége
Ciklus vége
 metszet:=c

Operátor vége.

² A min művelet ugyanez, csak ebben a sorban a > relációt <-re kell cserélni.

A megoldás alapvető problémája, hogy sehol sem ellenőrizhető, hogy a multihalmazban valóban csak a benne előfordulható elemek vannak.

Az így ábrázolt multihalmazok elemtípusára semmilyen megkötést nem kell tennünk, hiszen egy tömbben bármilyen elem elhelyezhető.

Még arra sincs korlátozás, hogy mekkora lehet az alaphalmaz elemszáma, amiből a multihalmaz elemei származnak. Csak annyi a megkötésünk, hogy a konkrét multihalmazok elemszámát korlátozzuk.

Darabszám vektor

Vegyünk fel egy annyi elemből álló sorozatot, amennyi a multihalmaz lehetséges elem fajtáinak száma! Képezzük le a multihalmaz elemtípusát az 1..Max'Elemzésám típusra! Legyen az i . elem x értékű, ha az i . lehetséges elem x -szer van benne van a multihalmazban, illetve 0, ha nincs benne!

Multihalmaz(Elemtípus)=**Tömb**(Min'Elemtípus..Max'Elemtípus: Egész)

Ebben az esetben a multihalmazműveleteket egész aritmetikai műveletekre visszavezetve valósítjuk meg:

```
Eljárás Beolvasás(Változó a: Multihalmaz(Elemtípus)):
  Be: N [a multihalmaz elemszáma]
  Üres(a)
  Ciklus i=1-től N-ig
    Be: e,db; a[e]:=db
  Ciklus vége
Eljárás vége.
```

Műveletigény számítása: a ciklus a multihalmaz elemeinek számszor fut le, azaz a futási idő a multihalmaz **elemszámával arányos**.

```
Eljárás Kiírás(Konstans a: Multihalmaz(Elemtípus)):
  Ciklus i=Min'Elemtípus-től Max'Elemtípus-ig
    Ha a[i]>0 akkor Ki: i,a[i]
  Ciklus vége
Eljárás vége.
```

Műveletigény számítása: a ciklus a multihalmaz típus lehetséges elemeinek számszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

```
Eljárás Üres(Változó a: Multihalmaz(Elemtípus)):
  Ciklus i=Min'Elemtípus-től Max'Elemtípus-ig
    a[i]:=0
  Ciklus vége
Eljárás vége.
```

Műveletigény számítása: a ciklus a multihalmaz típus lehetséges elemeinek számszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

Függvény Üres?(**Konstans** a: Multihalmaz(Elemtípus)):
 i:=Min'Elemtípus
Ciklus amíg i≤Max'Elemtípus **és nem** eleme(i,a)
 i:=i+1
Ciklus vége
 Üres? := (i>Max'Elemtípus)
Függvény vége.

Műveletigény számítása: a ciklus a multihalmaz típus lehetséges elemeinek számaszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

Eljárás Multihalmazba(**Változó** a: Multihalmaz(Elemtípus),
Konstans e: Elemtípus):
 a[e]:=a[e]+1
Eljárás vége.

Műveletigény számítása: nem függ a multihalmaz **elemszámától**.

Eljárás Multihalmazból(**Változó** a: multihalmaz(Elemtípus),
Konstans e: Elemtípus):
Ha a[e]>0 **akkor** a[e]:=a[e]-1
Eljárás vége.

Műveletigény számítása: nem függ a multihalmaz **elemszámától**.

Függvény eleme(**Konstans** e: Elemtípus,
 a: Multihalmaz(Elemtípus)): Logikai
 eleme:=a[e]>0
Függvény vége.

Műveletigény számítása: nem függ a multihalmaz **elemszámától**.

Függvény multiplicitás(**Konstans** a: Multihalmaz(Elemtípus),
 e: Elemtípus): Egész
 multiplicitás:=a[e]
Függvény vége.

Műveletigény számítása: nem függ a multihalmaz **elemszámától**.

Függvény része(**Konstans** a,b: Multihalmaz(Elemtípus)):
Logikai
 i:=Min'Elemtípus
Ciklus amíg i≤Max'Elemtípus **és** a[i]≤b[i]
 i:=i+1
Ciklus vége
 része:=i>Max'Elemtípus
Függvény vége.

Műveletigény számítása: a ciklus a multihalmaz típus lehetséges elemeinek számaszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

Operátor metszet(**Konstans** a,b: Multihalmaz(Elemtípus)) :
 Multihalmaz(Elemtípus)
Változó c: Multihalmaz(Elemtípus)
Ciklus i=Min'Elemtípus-tól Max'Elemtípus-ig
 c[i]:=min(a[i],b[i])
Ciklus vége
 metszet:=c
Operátor vége.

Műveletigény számítása: a ciklus a multihalmaz típus lehetséges elemeinek számszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

Operátor unió(**Konstans** a,b: Multihalmaz(Elemtípus)) :
 Multihalmaz(Elemtípus)
Változó c: Multihalmaz(Elemtípus)
Ciklus i=Min'Elemtípus-tól Max'Elemtípus-ig
 c[i]:=a[i]+b[i]
Ciklus vége
 unió:=c
Operátor vége.

Műveletigény számítása: a ciklus a multihalmaz típus lehetséges elemeinek számszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

Operátor max(**Konstans** a,b: Multihalmaz(Elemtípus)) :
 Multihalmaz(Elemtípus)
Változó c: Multihalmaz(Elemtípus)
Ciklus i=Min'Elemtípus-tól Max'Elemtípus-ig
 c[i]:=max(a[i],b[i])
Ciklus vége
 unió:=c
Operátor vége.

Műveletigény számítása: a ciklus a multihalmaz típus lehetséges elemeinek számszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

Operátor különbség(**Konstans** a,b: Multihalmaz(Elemtípus)) :
 Multihalmaz(Elemtípus)
Változó c: Multihalmaz(Elemtípus)
Ciklus i=Min'Elemtípus-tól Max'Elemtípus-ig
 c[i]:=max(a[i]-b[i],0)
Ciklus vége
 különbség:=c
Operátor vége.

Műveletigény számítása: a ciklus a multihalmaz típus lehetséges elemeinek számszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

Feladatok multihalmazokra

a Nemes Tihamér OITV-ről és az Informatika OKTV-ről

1. feladat

Egy almatermelő N ($1 \leq N \leq 100$) fajta almát termel, ismerjük, hogy melyik fajtából mennyit. Egy kereskedő M ($1 \leq M \leq 100$) fajta almát szeretne venni tőle, azt is tudjuk, hogy melyik fajtából mennyit.

Készíts programot (alma.pas, alma.c, ...), amely megadja, hogy (A) a termelőnek melyik fajtából mennyi marad, valamint hogy (B) a kereskedő melyik fajtából mennyit tud vásárolni!

Példa:

Bemenet:

Termelő fajtái száma: 3	Kereskedő fajtái száma: 2
1. fajta neve: jonagold	1. fajta neve: golden
1. fajta mennyisége: 100	1. fajta mennyisége: 50
2. fajta neve: golden	2. fajta neve: starking
2. fajta mennyisége: 30	2. fajta mennyisége: 100
3. fajta neve: idared	
3. fajta mennyisége: 500	

Kimenet:

Termelőnél marad:	Kereskedő vesz:
jonagold 100	golden 30
idared 500	

Az A feladat megoldása a termelő és a kereskedő multihalmazának különbsége, a B feladat megoldása pedig a két multihalmaz metszete.

A:

`x:=különbség(termelő, kereskedő)`

Eljárás vége.

B:

`x:=metszet(termelő, kereskedő)`

Eljárás vége.

2. feladat

Egy programozási versenyen minden versenyző választhat egy programozási nyelvet, amin dolgozni fog.

Készíts programot (NYELVEK.PAS, NYELVEK.C, ...) a következő feladat megoldására, A programod olvassa be a választható nyelvek számát ($1 \leq M \leq 10$) és a versenyen induló tanulók számát ($1 \leq N \leq 100$), majd a választható nyelveket, s legvégül az egyes tanulók által válasz-

tott nyelveket! Ezután adja meg, hogy mely tanulók választottak illegális nyelvet (olyat, ami nem szerepelt a felsoroltak között), mely nyelveket nem választotta senki, s **melyik választott nyelvet hányan választották!**

Példa:

Nyelvek száma: 3 ⇒ Illegális nyelv: 3. versenyző
 Versenyzők száma: 5 Nem választott nyelv: Logo
 Választható nyelvek: Választott nyelvek:
 Pascal Pascal: 3 versenyző
 Logo C++: 1 versenyző
 C++
 Választott nyelvek:
 Pascal
 Pascal
 Delphi
 C++
 Pascal

Itt a harmadik részfeladat egy multihalmaz előállítás, majd kiírása: (a megoldás a választott nyelvek beolvasásával kezdődik)

```
C(N) :
    Üres(a)
    Ciklus i=1-től N-ig
        Be: x; Multihalmazba(a,x)
    Ciklus vége
    Kiírás(a)
Eljárás vége.
```

3. feladat

A kukutyini állatkert N, a rátóti pedig M fajta ($N, M \leq 100$) állatot tart. Tudjuk, hogy melyik állatkertben melyik fajtából hány példány van. A két állatkert olyan állatokat cserélhet egymással, amelyekből mindkettőnek legalább K példánya van, illetve olyan állatokat ajándékozhat a másiknak, amelyekből legalább L példánya van, a másiknak pedig egyetlen példánya sincs.

Készíts programot, amely beolvassa a kukutyini, majd a rátóti állatkert adatait, végül pedig K és L értékét! Mind a két esetben be kell olvasni az állatfajták számát, majd az egyes állatfajták nevét és példányszámát. A program írja ki, hogy (A) mely állatfajtákból cserélhet a két állatkert, illetve (B) mely állatfajtákból ajándékozhat a kukutyini a rátótinak, s (C) melyekből a rátóti a kukutyininak!

Példa:

A kukutyini fajták száma? 4 A rátóti fajták száma? 5
 1. állat? kecske 1. állat? kecske

példányszáma? 12	példányszáma? 8
2. állat? nyúl	2. állat? pulyka
példányszáma? 41	példányszáma? 16
3. állat? ló	3. állat? szamár
példányszáma? 8	példányszáma? 1
4. állat? liba	4. állat? ló
példányszáma? 76	példányszáma? 4
	5. állat? liba
	példányszáma? 60

Mekkora létszám fölött lehet cserélni? 6

Mekkora létszám fölött lehet ajándékozni? 12

Csere:	Kukutyin ajándékoz:	Rátót ajándékoz:
kecske	nyúl	pulyka
liba		

Az A részfeladat megoldása a két multihalmaz metszetének azon elemei, amelyek multiplicitása legalább K, a B és a C részfeladaté pedig azon elemek, amelyek egyik multihalmazban legalább L a másikban pedig 0 multiplicitással fordulnak elő.

A:

`x:=metszet(kukutyin,rátót)`

Ciklus `i=1-től x.db-ig`

Ha `x.elem[i].multi≥K` **akkor** **Ki:** `x.elem[i].érték`

Ciklus vége

Eljárás vége.

B:

Ciklus `i=1-től kukutyin.db-ig`

Ha `kukutyin.elem[i].multi≥L` **és**

`multiplicitás(rátót,kukutyin.elem[i].érték)=0`

akkor **Ki:** `kukutyin.elem[i].érték`

Ciklus vége

Eljárás vége.

C:

Ciklus `i=1-től rátót.db-ig`

Ha `rátót.elem[i].multi≥L` **és**

`multiplicitás(kukutyin,rátót.elem[i].érték)=0`

akkor **Ki:** `rátót.elem[i].érték`

Ciklus vége

Eljárás vége.