



Belépő a tudás közösségébe

Informatika szakköri segédanyag



Multihalmaz típus

**Bende Imre, Heizlerné Bakonyi Viktória, Menyhárt László,
Szlávi Péter, Törley Gábor, Zsakó László**

Szerkesztő: Abonvi-Tóth Andor, Zsakó László

A kiadvány „A felsőoktatásba bekerülést elősegítő készségfejlesztő és kommunikációs programok megvalósítása, valamint az MTMI szakok népszerűsítése a felsőoktatásban” (EFOP-3.4.4-16-2017-006) című pályázat keretében készült 2018-ban.



Eötvös Loránd Tudományegyetem
Informatikai Kar

SZÉCHENYI 2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

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

Műveletek: **+** (egyesítés - unió), *max* (értékek uniója, multipllicitások maximuma), *min* (értékek uniója, multipllicitások minimuma¹), ***** (metszet), *mindkét?* (a két multihalmaz az elemek multipllicitásától eltekintve azonos-e), **-** (különbség), *Elem* (egy elem benne van-e a multihalmazban), *multipllicitás* (egy elem hány-szoros multipllicitással 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).

Példa:

```
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.

```
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.
```

¹ Ha van ennek egyáltalán értelme.

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)): Logikai

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

eleme:=i≤a.db

Függvény vége.

Műveletigény számítása: a ciklus az A multihalmaz elemszamaszor 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 akkor2
        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.

```

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'Elemszám típusra! Legyen az i . elem x értékű, ha az i . lehetséges elem x -szer van benne van a multihalmazban, illetve 0, ha nincs benne!

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


```
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 számaszor 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 számaszor fut le, azaz a futási idő a **multihalmaz típus elemszámával arányos**.

```
Operátor Maximum(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
    maximum:=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**.

Minta kódok.

C++	cpp/feladat.cpp
C#	cs/feladat.cs
Java	java/feladat.java
Pascal	pas/feladat.pas
Python	py/feladat.py



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, 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

- 1. fajta neve: jonagold
- 1. fajta mennyisége: 100
- 2. fajta neve: golden
- 2. fajta mennyisége: 30
- 3. fajta neve: idared
- 3. fajta mennyisége: 500

Kereskedő fajtái száma: 2

- 1. fajta neve: golden
- 1. fajta mennyisége: 50
- 2. fajta neve: starking
- 2. fajta mennyisége: 100

Kimenet:

Termelőnél marad:

jonagold 100

Kereskedő vesz:

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:
 $x := \text{különbség}(\text{termelő}, \text{kereskedő})$

Eljárás vége.

B:
 $x := \text{metszet}(\text{termelő}, \text{kereskedő})$

Eljárás vége.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Sorozatok
Feladat	2. Alma

2. feladat

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

Készíts programot 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 választott 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 \Rightarrow 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 előállítására, majd kiírására: (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.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Sorozatok
Feladat	63. Nyelvek programozási versenyen, C. feladat

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

1. állat? kecske
példányszáma? 12

2. állat? nyúl
példányszáma? 41

3. állat? ló
példányszáma? 8

4. állat? liba
példányszáma? 76

A rátóti fajták száma? 5

1. állat? kecske
példányszáma? 8

2. állat? pulyka
példányszáma? 16

3. állat? szamár
példányszáma? 1

4. állat? ló
példányszáma? 4

5. állat? liba
példányszáma? 60

Mekkora létszám fölött lehet cserélni? 6

Mekkora létszám fölött lehet ajándékozni? 12

Csere:

kecske
liba

Kukutyin ajándékoz:

nyúl

Rátót ajándékoz:

pulyka

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.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Sorozatok
Feladat	1. Állatkertek