

An aerial photograph of Budapest, Hungary, showing the city's architecture and the Danube River. A semi-transparent white rectangular box is centered over the image, containing the text "Programozás" and "10. előadás".

Programozás

10. előadás

Tartalom

- Halmazos tételek
 - Metszet
 - Unió
- Halmaz
 - Halmaz típus elemek felsorolásával
 - Halmaz típus darabszám vektorral
- Halmaz általánosítása: Multihalmaz
 - Multihalmaz típus elemek felsorolásával
 - Multihalmaz típus darabszám vektorral



Metszet



Feladatok:

- A télen **és** a nyáron megfigyelhető madarak**k** alapján **adjuk meg** a nem költöző madarak**at**!
- **Két** ember szabad órá**i** alapján **mondjuk meg**, hogy mikor beszélgethetnek egymással!
- **Adjuk meg** azokat az állatfajok**at**, amelyeket a budapesti **és** a veszprémi állatkertben **is** megnézhetünk!
- **Három** virágárusnál kapható virágok**k** közül **adjuk meg** azokat, amelyek **mindegyik**nél kaphatóak!



Metszet



Feladatok:

- Adjuk meg két természetes szám közös osztóit!
- A télen és a nyáron megfigyelhető madarak alapján adjuk meg a nem költöző madarakat!
- Két ember szabad órái alapján mondjuk meg, hogy mikor beszélgethetnek egymással!
- Adjuk meg azokat az állatokat, amelyeket a budapesti és a veszprémi állatkertben is megnézhetünk!

Mi bennük a közös?

Ismerünk két halmazt (tetszőleges, de azonos típusú elemekkel), meg kell adnunk azokat az elemeket, amelyek mindkét halmazban szerepelnek!
A több halmaz visszavezethető a két halmaz esetére.



Metszet

Ismerünk két halmazt (tetszőleges típusú elemekkel), meg kell adnunk azokat az elemeket, amelyek mindkét halmazban szerepelnek!

Specifikáció:

➤ Bemenet: $N, M \in \mathbb{N}$, $X_{1..N} \in H^N$, $Y_{1..M} \in H^M$

➤ Kimenet: $Db \in \mathbb{N}$, $Z_{1..\min(N,M)} \in H^{\min(N,M)}$

➤ Előfeltétel: **HalmazE**(X) és **HalmazE**(Y)

➤ Utófeltétel: $Db = \sum_{\substack{i=1 \\ X_i \in Y}}^N 1$ és

$\forall i(1 \leq i \leq Db): (Z_i \in X \text{ és } Z_i \in Y) \text{ és } \text{HalmazE}(Z)$

➤ Definíció: **HalmazE**: $H^* \rightarrow \mathbb{L}$

HalmazE(x) := nem ($\exists i(1 \leq i \leq \text{Hossz}(x)): x_i \in x_{1..i-1}$)

Az első Db elemet használva

Az elemtartalmazás egyértelmű-e.



Metszet

Specifikáció:

- Bemenet: $N, M \in \mathbb{N}, X_{1..N} \in H^N, Y_{1..M} \in H^M$
- Kimenet: $Db \in \mathbb{N}, Z_{1..\min(N,M)} \in H^{\min(N,M)}$
- Előfeltétel: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$
- Utófeltétel: $Db = \sum_{\substack{i=1 \\ X_i \in Y}}^N 1$ és
 $\forall i(1 \leq i \leq Db): (Z_i \in X \text{ és } Z_i \in Y) \text{ és } \text{HalmazE}(Z)$

Specifikáció₂:

- Utófeltétel₂: $(Db, Z) = \text{Metszet}(N, X, M, Y)$

Specifikáció₃:

- Utófeltétel₃: $(Db, \textcolor{red}{Z}) = \text{Kiválogat } \overset{N}{\underset{\substack{i=1 \\ \textcolor{red}{X}_i \in Y}}{X_i}}$

Eldöntés tétel

$$\text{Van} = \overset{M}{\exists} \underset{j=1}{X_i = Y_j}$$

Kiválogatás tétel

$$(Db, Y) = \overset{N}{\underset{\substack{i=1 \\ T(X_i)}}{\text{Kiválogat } X_i}}$$

Eldöntés tétel

$$\text{Van} = \overset{N}{\exists} \underset{i=1}{T(X_i)}$$



Metszet

Változó
i,j:Egész

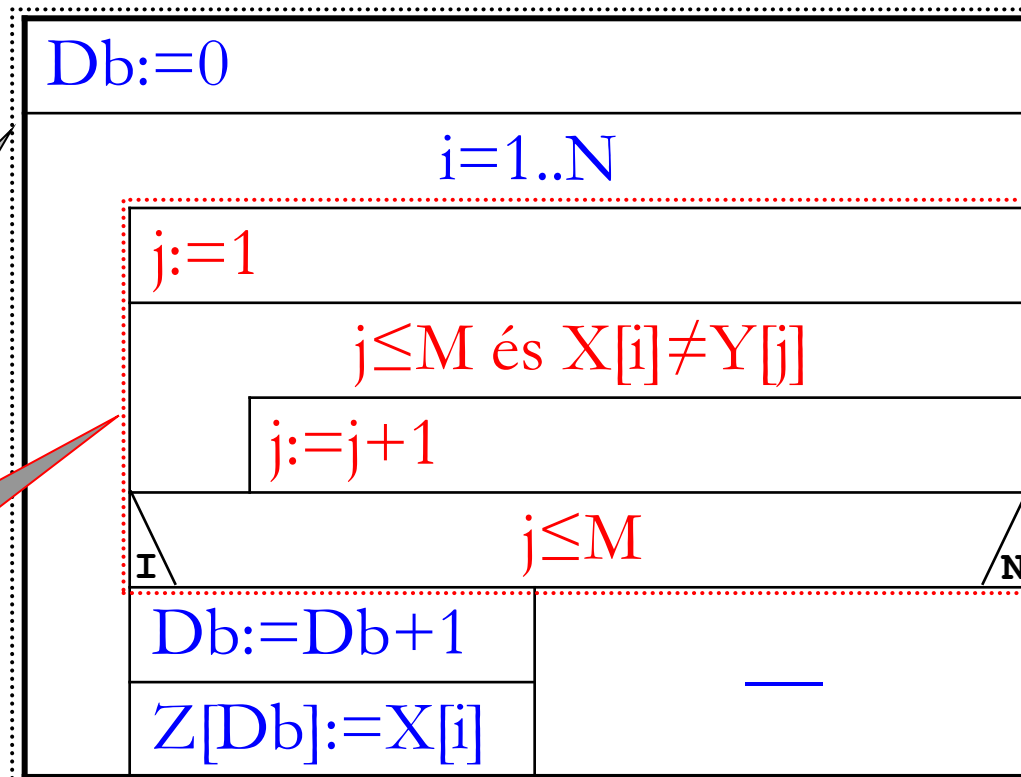
Algoritmus:

Specifikáció:

- Bemenet: $N, M \in \mathbb{N}, X_{1..N} \in H^N, Y_{1..M} \in H^M$
- Kimenet: $Db \in \mathbb{N}, Z_{1..min(N,M)} \in H^{min(N,M)}$
- Előfeltétel: $HalmazE(X)$ és $HalmazE(Y)$
- Utófeltétel: $Db = \sum_{x_i \in Y} 1$ és
 $\forall i (1 \leq i \leq Db): (Z_i \in X \text{ és } Z_i \in Y) \text{ és } HalmazE(Z)$

Kiválogatás tétel

Eldöntés tétel



Megjegyzés:

A megoldás kiválogatásban eldöntés.



Feladatok:

- A télen **és** a nyáron megfigyelhető madarak**k** alapján **adjuk meg**, hogy milyen madarak**at** figyeltek meg!
- **Két** ember szabad órái alapján **mondjuk meg**, hogy mikor tudjuk elérni valamelyiket!
- **Három** szakkör tanuló**i** alapján **soroljuk fel** a szakkörre járó**kat**!
- **Adjuk meg** azokat az állatfajok**at**, amelyeket a budapesti **vagy** a veszprémi állatkertben megnézhetünk!



Unió



Feladatok:

- Két szakkör tanulói alapján adjuk meg a szakkörre járókat!
- A télen és a nyáron megfigyelhető madarak alapján adjuk meg a megfigyelhető madarakat!
- Két ember szabad órái alapján mondjuk meg, hogy mikor tudjuk elérni valamelyiket!
- Adjuk meg azokat az állatokat, amelyeket a budapesti vagy a veszprémi állatkertben megnézhetünk!

Mi bennük a közös?

Ismerünk két halmazt (tetszőleges, de azonos típusú elemekkel), meg kell adnunk azokat az elemeket, amelyek legalább az egyik halmazban szerepelnek!

A több halmaz visszavezethető a két halmaz esetére.



Unió



Ismerünk két halmazt (tetszőleges típusú elemekkel), meg kell adnunk azokat az elemeket, amelyek legalább az egyik halmazban szerepelnek!

Specifikáció:

- Bemenet: $N, M \in \mathbb{N}$,
 $X_{1..N} \in H^N, Y_{1..M} \in H^M$
- Kimenet: $Db \in \mathbb{N}, Z_{1..N+M} \in H^{N+M}$
- Előfeltétel: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$
- Utófeltétel: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^M 1$ és
 $\forall i (1 \leq i \leq Db): (Z_i \in X \text{ vagy } Z_i \in Y)$ és
 $\text{HalmazE}(Z)$

Az első Db elemet használva



Unió



Ismerünk két halmazt (tetszőleges típusú elemekkel), meg kell adnunk azokat az elemeket, amelyek legalább az egyik halmazban szerepelnek!

Specifikáció₂:

- Utófeltétel₂: $(Db, Z) = \text{Unió}(N, X, M, Y)$

Specifikáció₃:

- Utófeltétel₃: $(Db, Z) = (N, X) \oplus \bigoplus_{\substack{j=1 \\ Y_j \notin X}}^M \text{Kiválogat } Y_j$

Specifikáció:

- Bemenet: $N, M \in \mathbb{N}, X_{1..N} \in H^N, Y_{1..M} \in H^M$
- Kimenet: $Db \in \mathbb{N}, Z_{1..N+M} \in H^{N+M}$
- Előfeltétel: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$
- Utófeltétel: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^M 1$ és
 $\forall i (1 \leq i \leq Db): (Z_i \in X \text{ vagy } Z_i \in Y) \text{ és } \text{HalmazE}(Z)$

\oplus az összefűzés jele.
Sorozatok összefűzése.

Másolás tétel:

X sorozat „kezdőértékkel”



Unió

Algoritmus:

Specifikáció:

- > Bemenet: $N, M \in \mathbb{N}$, $X_{1..N} \in H^N$, $Y_{1..M} \in H^M$
- > Kimenet: $Db \in \mathbb{N}$, $Z_{1..N+M} \in H^{N+M}$
- > Előfeltétel: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$
- > Utófeltétel: $Db = N + \sum_{i=1}^M 1$ és
 $\forall i (1 \leq i \leq Db): (Z_i \in X \text{ vagy } Z_i \in Y) \text{ és } \text{HalmazE}(Z)$

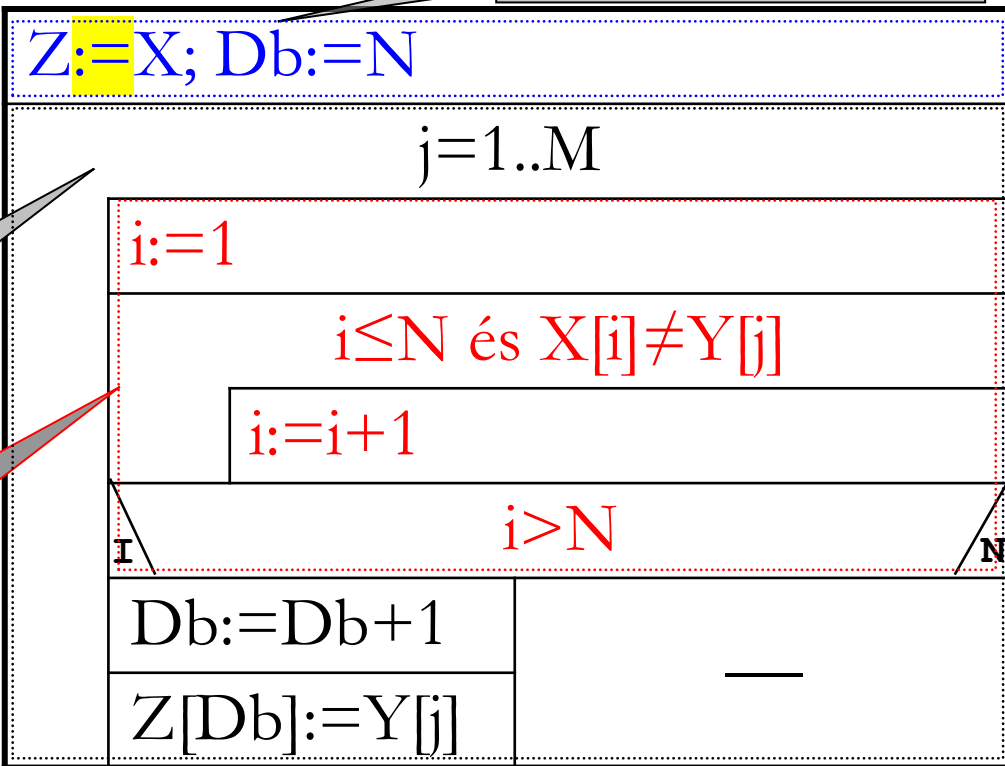
Kiválogatás tétel

Eldöntés tétel

Másolás tétel

Változó

i, j : Egész



Sorozat \rightarrow halmaz transzformáció

Egyes feladatoknál, mint pl. a metszet és unió tételnél a kiinduló adatok halmazban vannak. Ha a bemeneten tetszőleges sorozatot kapunk, akkor szükség lehet rá, hogy abból halmazt készítsünk.

Példa: N vásárlásról ismerjük, hogy egy vásárló milyen terméket vásárolt ($Be[1..N]$). Adjuk meg a vásárlásokban szereplő termékeket ($T[1..Db]$)!

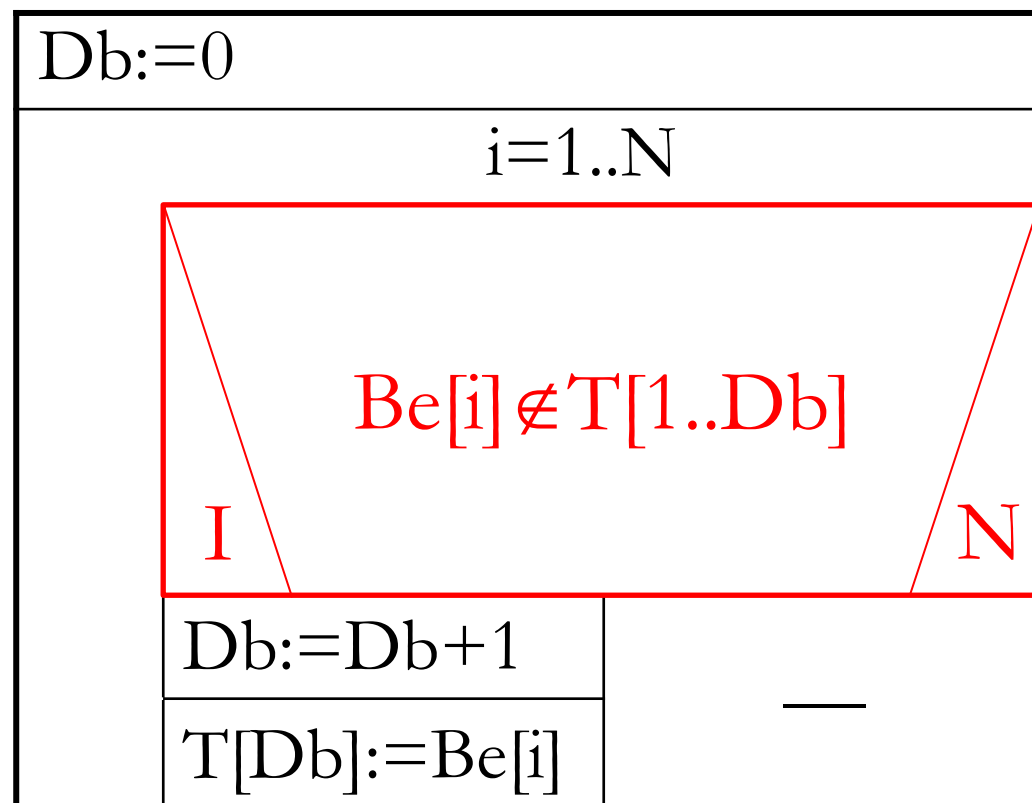
A megoldás egy **kiválogatás tétel**: válogassuk ki a bemenet azon elemeit, amelyek a kiválogatás eredményében **még nem szerepeltek** (**eldöntés**)!



Sorozat \rightarrow halmaz transzformáció



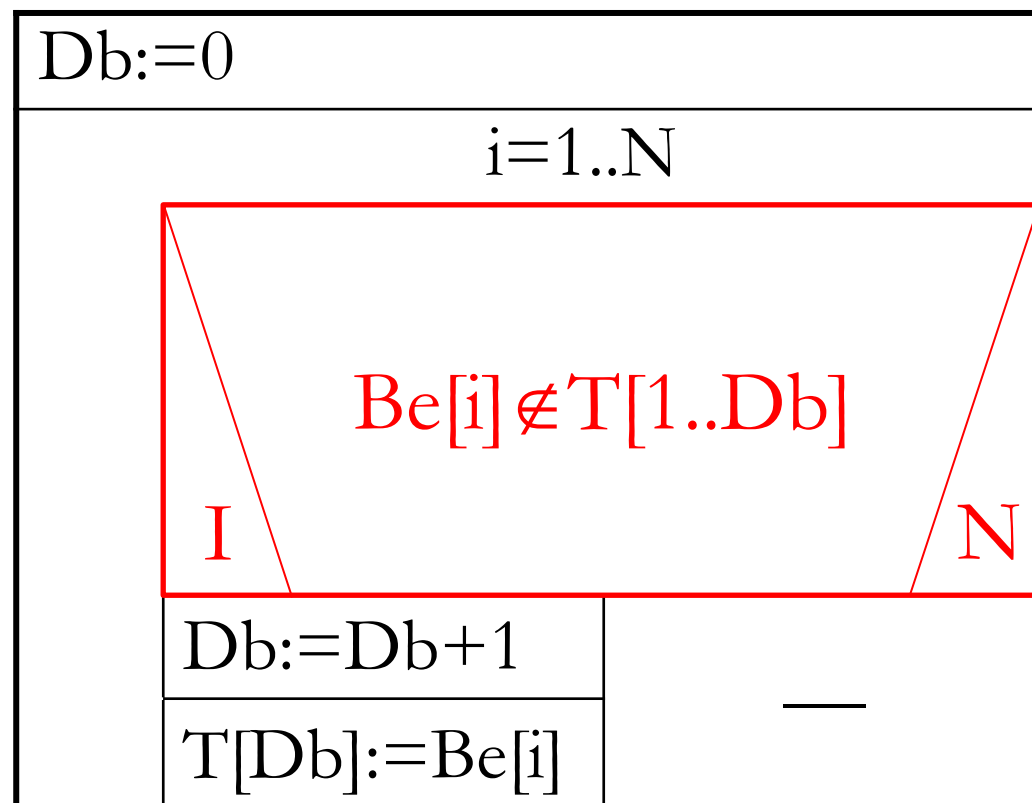
Változó
 i :Egész



Sorozat \rightarrow halmaz transzformáció



Változó
 i, j : Egész



Halmaz típus



Értékhalmaz:

Az alaphalmaz (amely az Elemtípus által van meghatározva) iteráltja („mely elemek lehetnek benne a halmazban”).

Az Elemtípus általában valamely véges diszkrét típus lehet, legtöbbször még az elemszámát is korlátozzák (<256).

Ha nyelvi elemként nem létezik, akkor a megvalósításunkban lehet nagyobb elemszámú is.



Halmaz típus

Műveletek (matematika)

- Metszet (\cap)
- Unió (\cup)
- Különbség (\setminus)
- Komplement – **nem mindig valósítható meg**
- Elem \in (elem benne van-e a halmazban) (\in)
- Rész \subseteq (egyik halmaz részhalmaza-e a másiknak) (\subset, \subseteq)



Halmaz típus

Műveletek (megvalósítás)

- Halmazba (elem hozzá vétele egy halmazhoz): $H := H \cup \{e\}$
- Halmazból (elem elhagyása egy halmazból): $H := H \setminus \{e\}$
- Beolvasás (halmaz beolvasása)
- Kiírás (halmaz kiírása),
- Üres (üres halmaz létrehozás eljárás), vagy
Üres'Halmaztípus előre definiált konstans
- ÜresE (logikai értékű függvény).



Halmaz típus

ábrázolása₁



Elemek felsorolása

Halmaz(**Elemtípus**)=

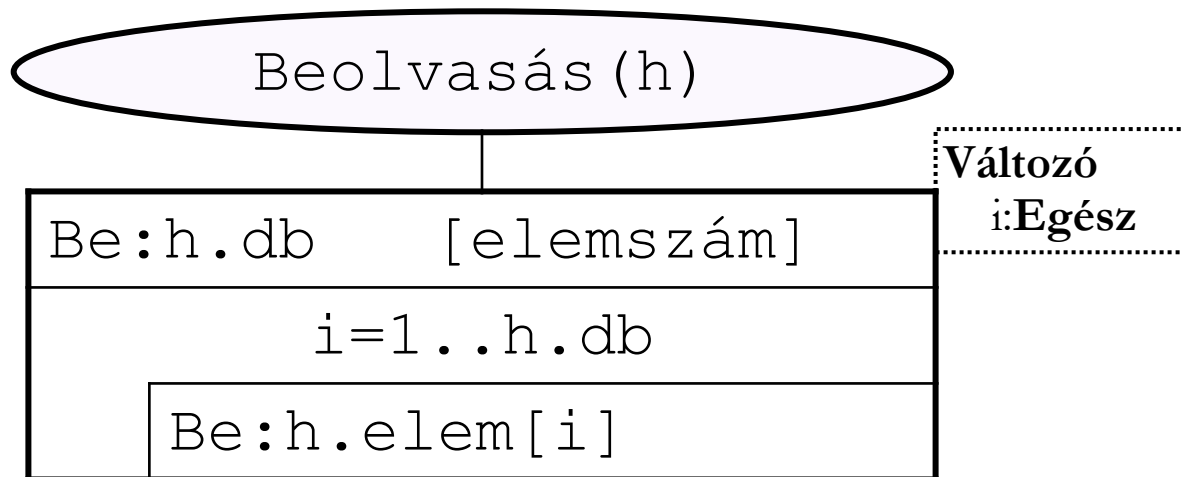
Rekord(db: **Egész**,

elem: **Tömb**[1..MaxDb:**Elemtípus**])

A halmaz elemeinek felsorolásával adjuk meg a halmazt, annyi elemű tömbben, ahány elemű éppen a halmaz (pontosabban az első db darab elemében). **Típusinvariáns: nincs értékismétlődés.**



Halmaz típus



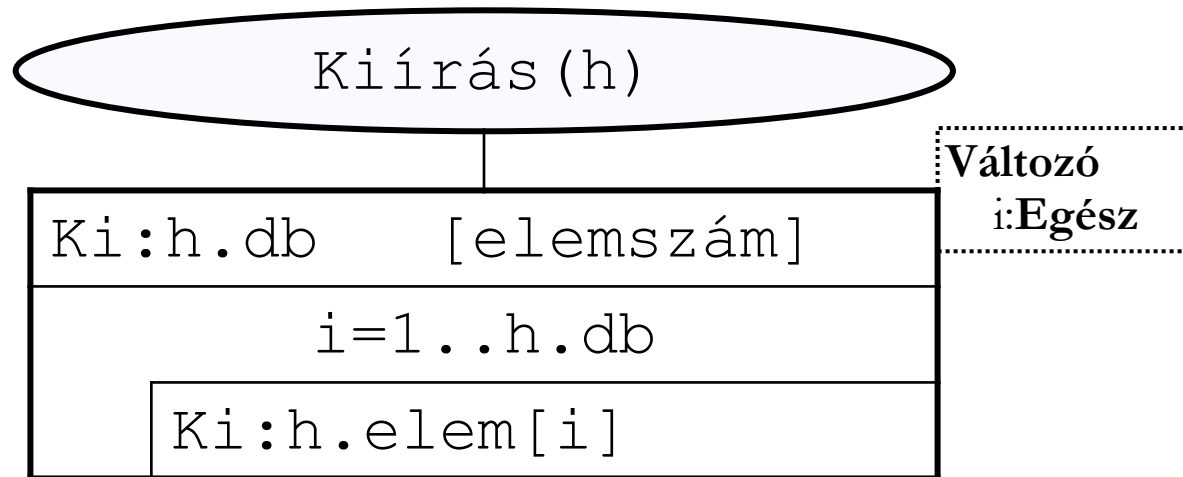
Feltesszük, hogy „halmazság” és a méretkorlát teljesül.

Műveletigény számítása:

A ciklus a halmaz elemeinek számaszor fut le, azaz a futási idő a halmaz elemszámával arányos.



Halmaz típus



Műveletigény számítása:

A ciklus a halmaz elemeinek számaszor fut le, azaz a futási idő a halmaz elemszámával arányos.



Halmaz típus

Üres (h)

$h.db := 0$

Műveletigény számítása:

Nem függ a halmaz elemszámától.

ÜresE (h) : Logikai

$ÜresE := h.db = 0$

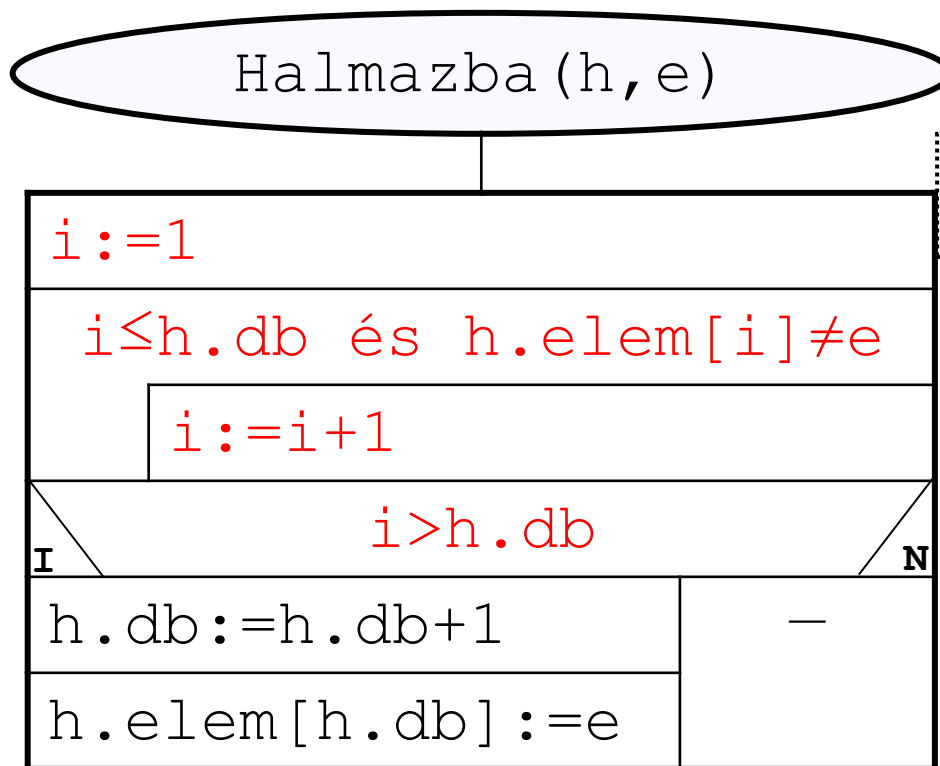
Műveletigény számítása:

Nem függ a halmaz elemszámától.



Halmaz típus

Az **Eldöntés** programozási tétel alkalmazása



Változó
 i : Egész

Műveletigény számítása:

A ciklus a halmaz elemeinek számaszor fut le, azaz a futási idő a halmaz elemszámával arányos.



Halmaz típus

Halmazból (h, e)

Változó
i: E

i := 1

i ≤ h.db és h.elem[i] ≠ e

i := i + 1

i ≤ h.db

h.elem[i] := h.elem[h.db]

h.db := h.db - 1

—

I

N

A Keresés
programozási
tétel alkalmazása

Műveletigény számítása:

A ciklus a halmaz elemeinek számaszor fut le, azaz a futási idő
a halmaz elemszámával arányos.



Halmaz típus

ElemE (e, h) : Logikai

$i := 1$

$i \leq h.db$ és $h.elem[i] \neq e$

$i := i + 1$

ElemE := $i \leq h.db$

Változó
 i : Egész

Az **Eldöntés** programozási
tétel alkalmazása

Műveletigény számítása:

A ciklus a halmaz elemeinek számaszor fut le, azaz a futási idő a halmaz elemszámával arányos.



Halmaz típus

RészeE (a, b) : Logikai

Az **Eldöntés** programozási
tétel alkalmazása,
eldöntés tulajdonsággal

$i := 1$

$i \leq a.db$ és $ElemeE(a.elem[i], b)$

$i := i + 1$

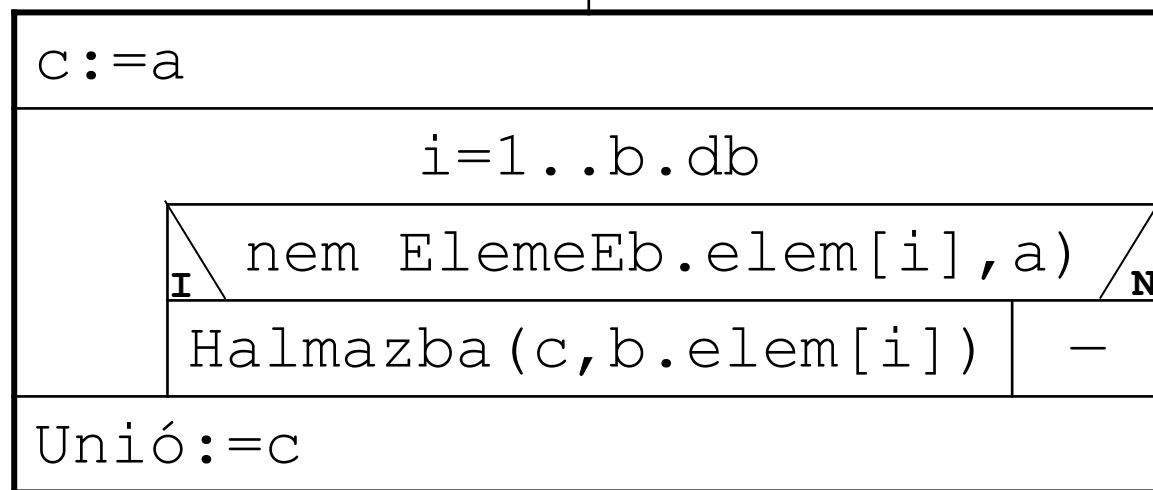
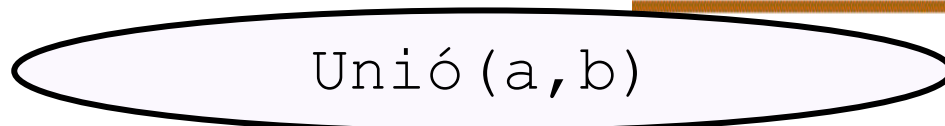
$RészeE := i > a.db$

Műveletigény számítása:

A ciklus az A halmaz elemszámaszor fut le, az Eleme függvény pedig a B halmaz elemszámaszor, azaz a futási idő a két halmaz elemszámának szorzatával arányos.



Halmaz típus



Változó
i: Egész
c: Halmaz

Másolás
+ Kiválogatás
+ Eldöntés

Műveletigény számítása:

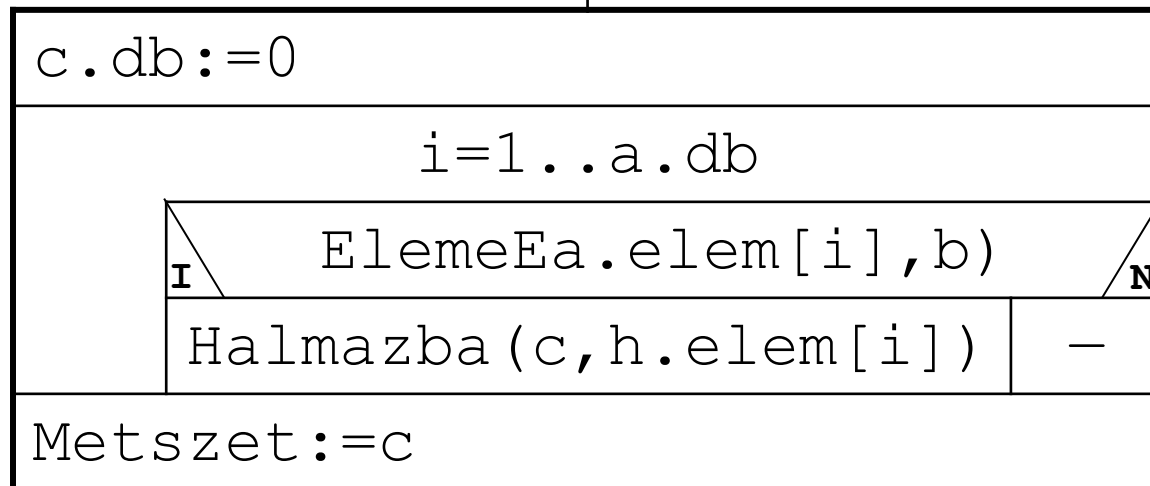
A külső ciklus a B halmaz elemszámáig fut le, az Eleme függvény pedig az A halmaz elemszámáig, azaz a futási idő a két halmaz elemszámának szorzatával arányos.



Halmaz típus



Változó
i: Egész
c: Halmaz



Kiválogatás
+Eldöntés

Műveletigény számítása:

A ciklus az A halmaz elemszámámszor fut le, az Eleme pedig legrosszabb esetben a B halmaz elemszámámszor, azaz a futási idő a két halmaz elemszámának szorzatával arányos.



Halmaz típus

Megjegyzések:

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

Arra sincs korlátozás, hogy mekkora lehet az alaphalmaz száma, amiből a halmaz elemei származnak. Csak a konkrét halmazok elemszámát korlátozzuk.



Halmaz típus

ábrázolása₂



Bittérkép – logikai vektor

Halmaz(Elemtípus)=

Tömb[Min'Elemtípus..Max'Elemtípus:Logikai]

A halmazt {igaz,hamis} (azaz benne van-e) elemekből álló vektorként értelmezzük, ahol **index**ként használjuk az **elem típusú** értéket vagy indexet számolunk belőle. Elemtípus például lehet:

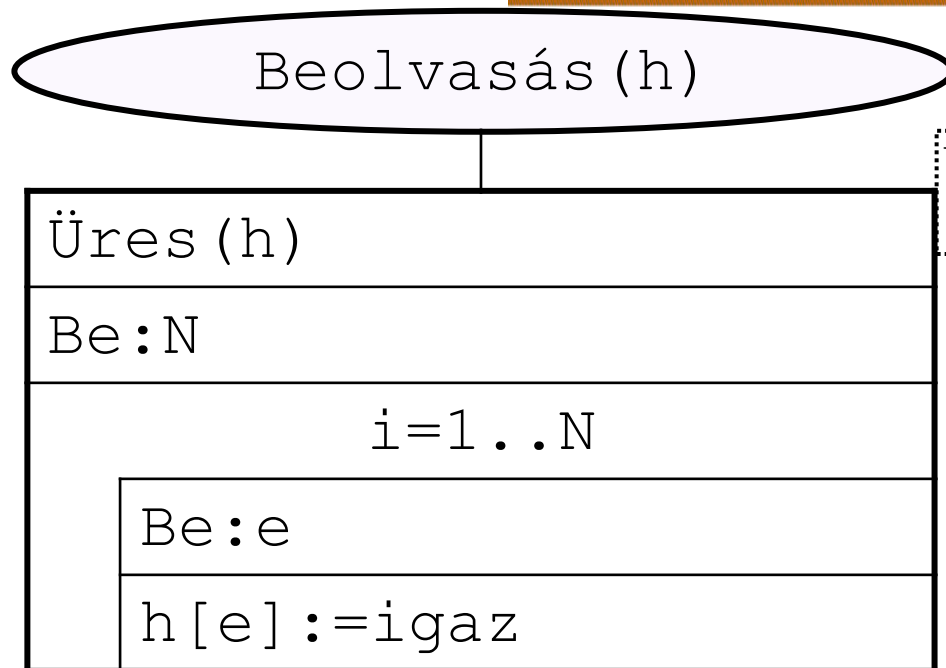
- egész számok intervalluma (-9..9)
- karakter-intervallum ("A".. "Z")

Az ilyen halmaz mindig rendezett halmaz, definiálható rajta a távolság fogalom (\rightarrow **implementálható a tömb címkiszámító függvénye**).

Kérdés: tároljuk-e a halmaz elemszámát is?



Halmaz típus



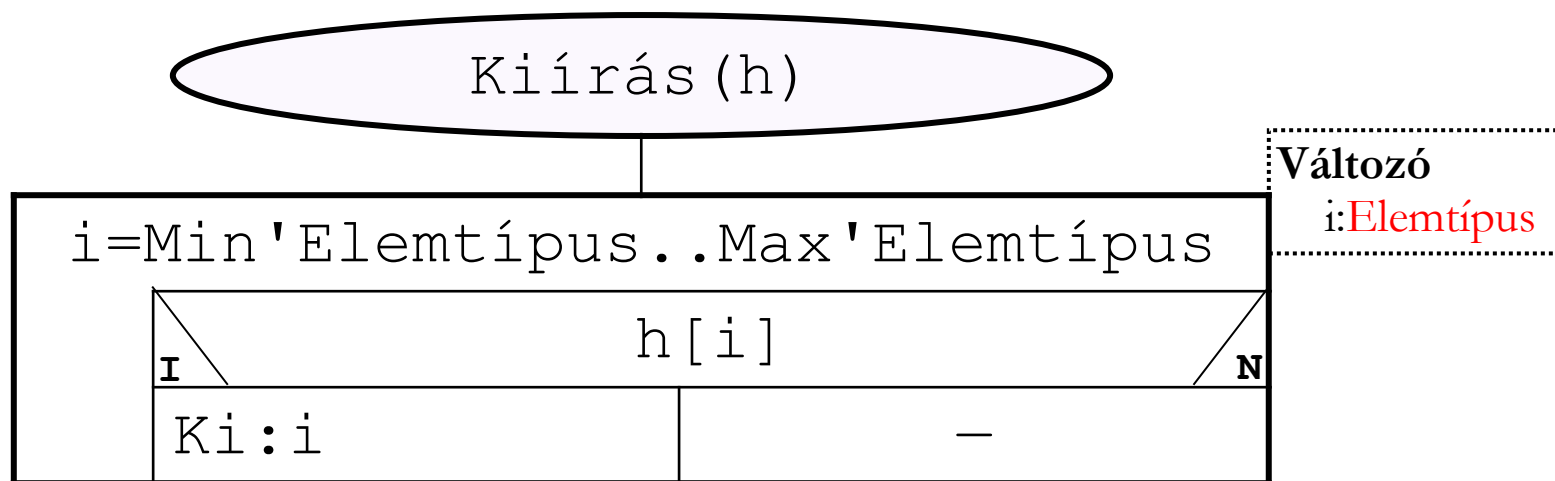
Változó
i: Egész

Műveletigény számítása:

Az Üres műveletigénye + a ciklus. A ciklus a halmaz elemeinek számaszor fut le, azaz a futási idő a halmaz elemszámával arányos.



Halmaz típus



Műveletigény számítása:

A ciklus a halmaz lehetséges elemeinek számaszor fut le, azaz a futási idő a halmaz elemtípusának számosságával arányos.

Mi lenne, ha tárolnánk a halmaz legkisebb és legnagyobb elemét is?



Halmaz típus

Üres (h)

A **Másolás** programozási tétel alkalmazása

```
i = Min 'Elemtípus' .. Max 'Elemtípus'  
h[i] := hamis
```

Műveletigény számítása:

A ciklus a halmaz lehetséges elemeinek számaszor fut le, azaz a futási idő a halmaz elemtípusának számosságával arányos.



Halmaz típus

ÜresE(h) : Logikai

Az **Eldöntés** programozási
tétel alkalmazása

$i := \text{Min}'\text{Elemtípus}$

$i \leq \text{Max}'\text{Elemtípus}$ és nem $h[i]$

$i := i + 1$ [=következő Elemtípusú érték]

$\text{ÜresE} := i > \text{Max}'\text{Elemtípus}$

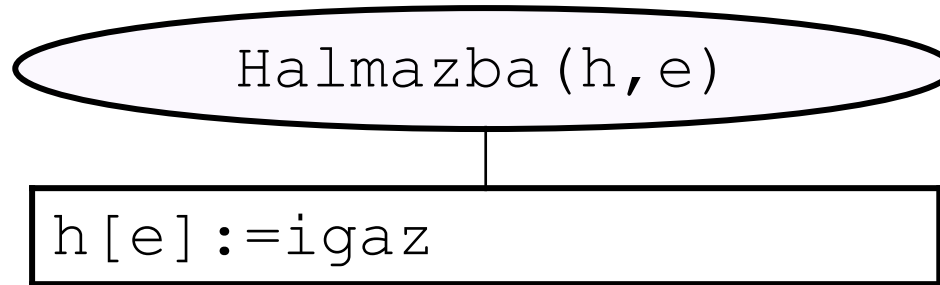
Műveletigény számítása:

A ciklus a halmaz lehetséges elemeinek számaszor fut le, azaz a futási idő a halmaz elemtípusának számosságával arányos.

Ha elemszámot tárolnánk, gyors lehetne ($\text{Db}=0?$).

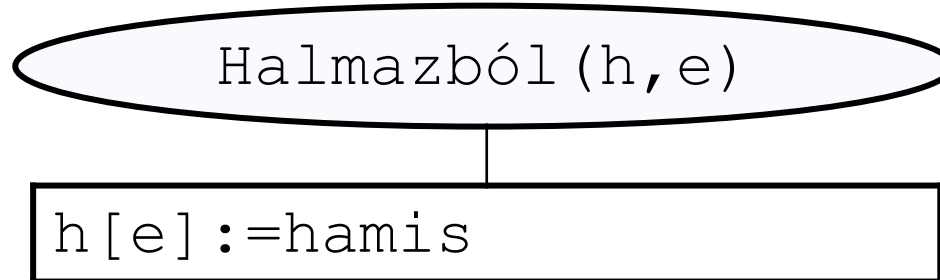


Halmaz típus



Műveletigény számítása:

Nem függ a halmaz elemszámától.

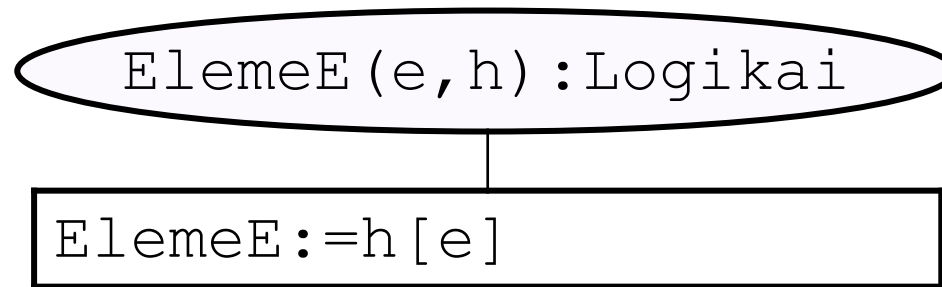


Műveletigény számítása:

Nem függ a halmaz elemszámától



Halmaz típus



Műveletigény számítása:

Nem függ a halmaz elemszámától.



Halmaz típus

RészeE (a, b) : Logikai

$i := \text{Min}'\text{Elemtípus}$

$i \leq \text{Max}'\text{Elemtípus}$ és
nem (a[i] és nem b[i])

$i := i + 1$

RészeE := $i > \text{Max}'\text{Elemtípus}$

Az **Eldöntés** programozási
tétel alkalmazása

Műveletigény számítása:

A ciklus a halmaz lehetséges elemeinek számaszor fut le, azaz a futási idő a halmaz elemtípusának számosságával arányos. Gyorsabb az előző ábrázolásnál, ha ez kisebb a két elemszám szorzatánál.



Halmaz típus

Unió (a, b)

```
i=Min'Elemtípus..Max'Elemtípus  
  c[i]:=a[i] vagy b[i]
```

```
Unió:=c
```

A **Másolás** programozási tétel alkalmazása:

Műveletigény számítása:

A ciklus a halmaz lehetséges elemeinek számaszor fut le, azaz a futási idő a halmaz elemtípusának számosságával arányos. Gyorsabb az előző ábrázolásnál, ha ez kisebb a két elemszám szorzatánál.



Halmaz típus

Metszet (a, b)

A **Másolás** programozási tétel alkalmazása

```
i=Min'Elemtípus..Max'Elemtípus
```

```
c[i]:=a[i] és b[i]
```

```
Metszet:=c
```

Műveletigény számítása:

A ciklus a halmaz lehetséges elemeinek számaszor fut le, azaz a futási idő a halmaz elemtípusának számosságával arányos. Gyorsabb az előző ábrázolásnál, ha ez kisebb a két elemszám szorzatánál.



Sorozat \rightarrow multihalmaz transzformáció



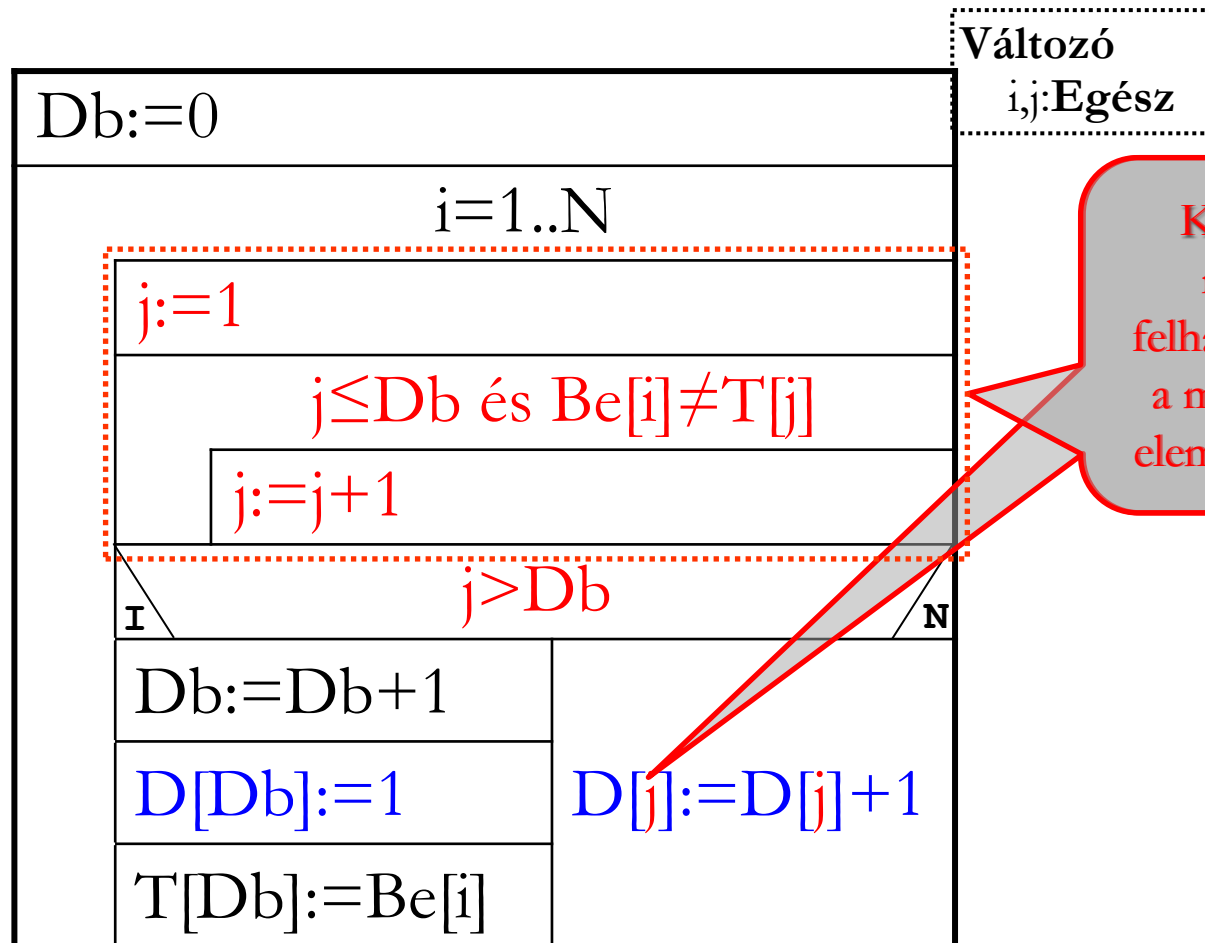
Egyes esetekben a bemenetbeli sorozatból multihalmazt kell készítenünk, ahol az elemek értéke mellett a számosságukat is tároljuk.

Példa: N vásárlásról ismerjük, hogy egy vásárló milyen terméket vásárolt ($Be[1..N]$). Adjuk meg a vásárlásokban szereplő termékeket ($T[1..Db]$) és számukat ($D[1..Db]$)!

A megoldás egy **kiválogatás tétel**: válogassuk ki a bemenet azon elemeit, amelyek a kiválogatás eredményében **még nem szerepeltek** (**eldöntés** \rightarrow **keresés**), s e közben **számláljunk** is (**megszámolás**)!



Sorozat \rightarrow multihalmaz transzformáció



Multihalmaz típus



Értékhalmoz:

Az alaphalmaz (amely az Elemtípus és egy darabszám által van meghatározva) iteráltja („mely elem hányszoros multiplicitással van benne a multihalmazban”).



Multihalmaz típus

Alapműveletek:

- Multihalmazba (elem hozzávétele egy multihalmazhoz):
 $H := H \cup \{(e, 1)\}$
- Multihalmazból (elem elhagyása egy multihalmazból):
 $H := H \setminus \{(e, 1)\}$
- Beolvasás (multihalmaz beolvasása)
- Kiírás (multihalmaz kiírása),
- Üres (üres multihalmaz létrehozás eljárás), vagy
- ÜresE (logikai értékű függvény).



Multihalmaz típus



Alapműveletek:

- $\text{Elem} \in E$ (egy elem benne van-e a multihalmazban) (\in)
- $\text{Benne} E$ (egy elem legalább adott multiplicitással benne van-e a multihalmazban)
- Multiplicitás (egy elem hányszoros multiplicitással van benne a multihalmazban)



Multihalmaz típus

Multihalmaz \times Multihalmaz műveletek:

- Metszet (\cap) (értékek metszete, multiplicitások minimuma)
- Unió (\cup) (értékek uniója, multiplicitások összege)
- Különbség (\setminus) (értékek különbsége, multiplicitások különbsége; nincs benne egy elem, ha a multiplicitások különbsége 1-nél kisebb)
- Max (multiplicitások maximuma),
- RészeE (egyik multihalmaz részhalmaza-e a másiknak) (\subset , \subseteq)
- MindközösE (a két multihalmaz az elemek multiplicitásától eltekintve azonos-e)



Multihalmaz típus

Példa:

Típus

ÁllatFajta=Szöveg

Állatok=Multihalmaz (ÁllatFajta)

Konstans

sok:Egész (10)

Változó

A:Állatok

A:=Állatok(("lúd",13),("disznó",1))	
I	"disznó"∈A és Multiplicitás(A,"lúd")≥sok
N	
Ki:"Sok lúd disznót győz"	–



Multihalmaz típus ábrázolása₁

Elemek felsorolása:

Típus

Halmazelem (Elemtípus) = **Rekord** (érték: Elemtípus,
multi: Egész)

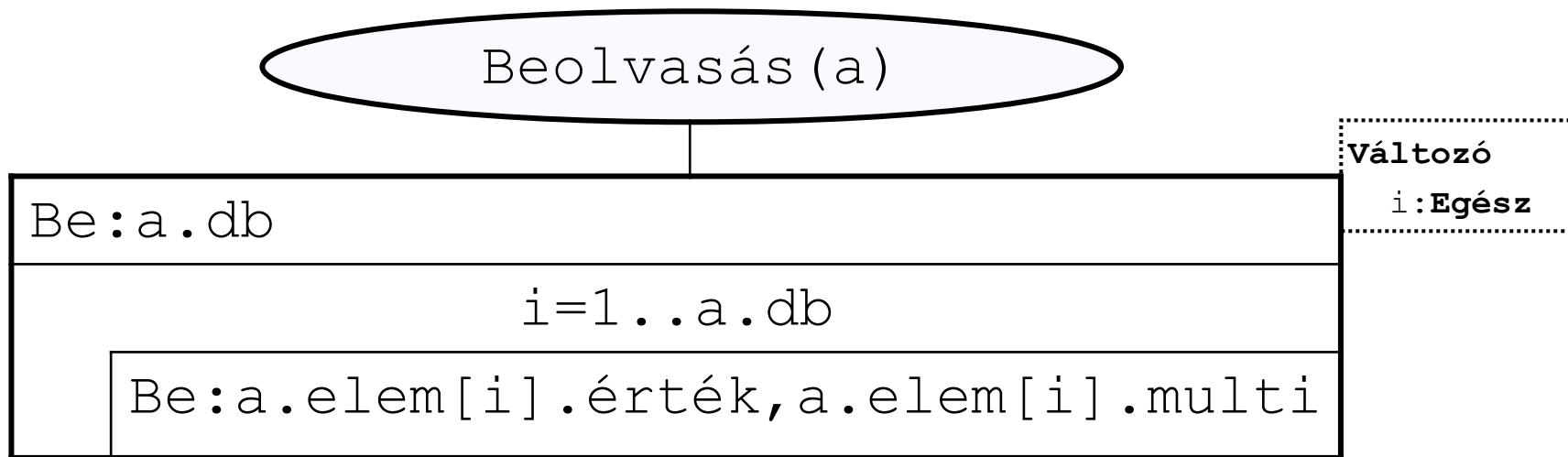
Multihalmaz (Elemtípus) = **Rekord** (
db: Egész,
elem: **Tömb** [1 .. MaxDb: Halmazelem (Elemtípus)])

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

Csak a legalább 1 multiplicitású elemeket tároljuk!



Multihalmaz típus

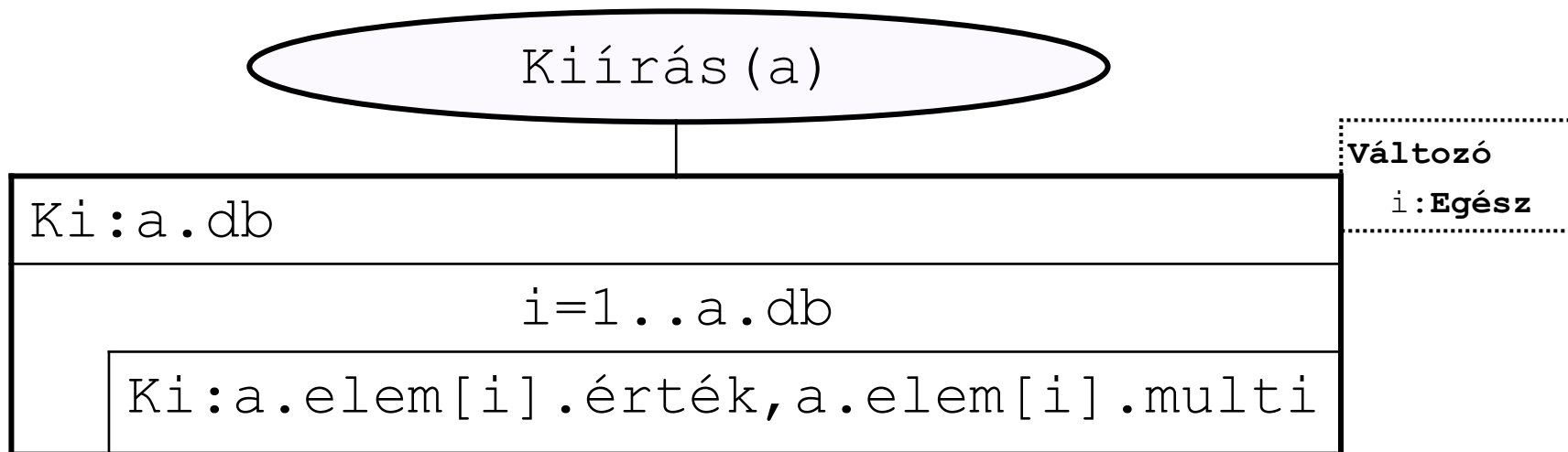


Műveletigény számítása:

A ciklus a multihalmaz elemértékeinek számaszor fut le, azaz a futási idő a multihalmaz elemszámával arányos.



Multihalmaz típus



Műveletigény számítása:

A ciklus a multihalmaz elemértékeinek számaszor fut le, azaz a futási idő a multihalmaz elemszámával arányos.



Multihalmaz típus

Üres (a)

a.db := 0

Műveletigény számítása:

Nem függ a multihalmaz elemszámától.

ÜresE (a) : **Logikai**

ÜresE := a.db = 0

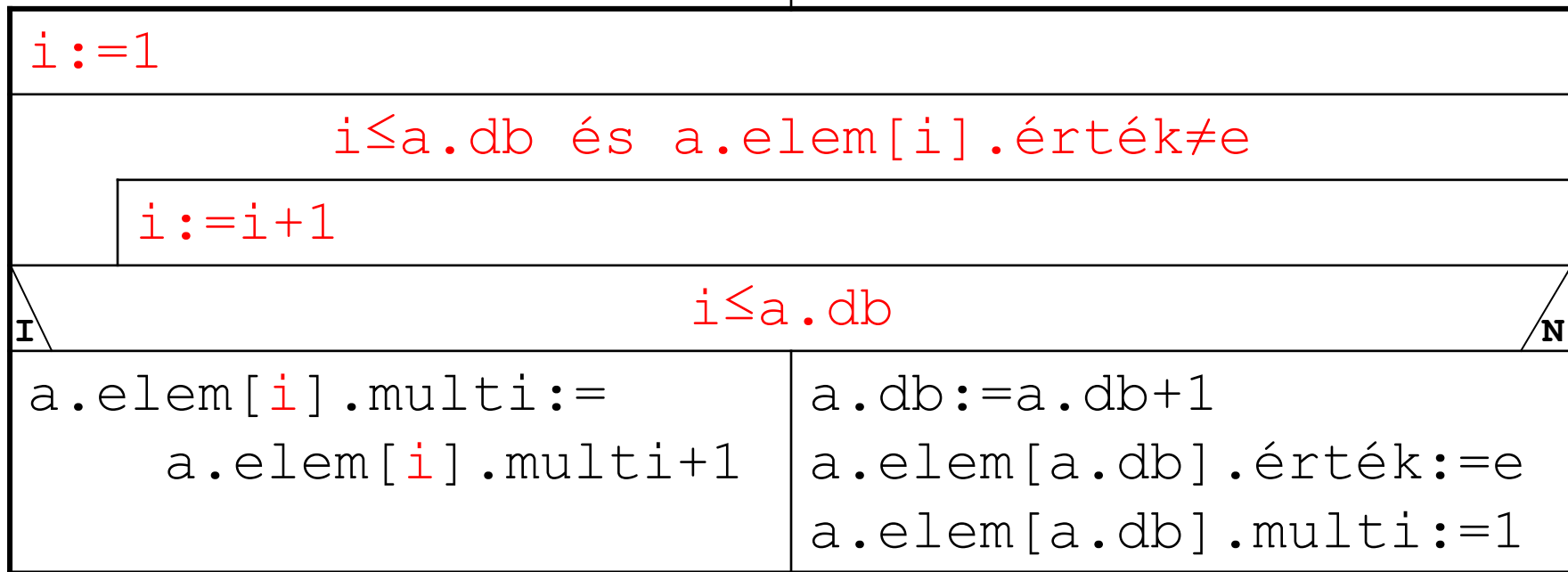
Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

Multihalmazba (a, e)



Műveletigény számítása:

Arányos a multihalmaz elemszámával (**keresés tétel**).



Multihalmaz típus

Multihalmazból (a, e)

i:=1			
i≤a.db és a.elem[i].érték≠e			
i:=i+1			
I	i≤a.db		N
I	a.elem[i].multi=1	N	—
a.elem[i]:=a.elem[a.db]		a.elem[i].multi:=a.elem[i].multi-1	
a.db:=a.db-1			

Válto

i : E

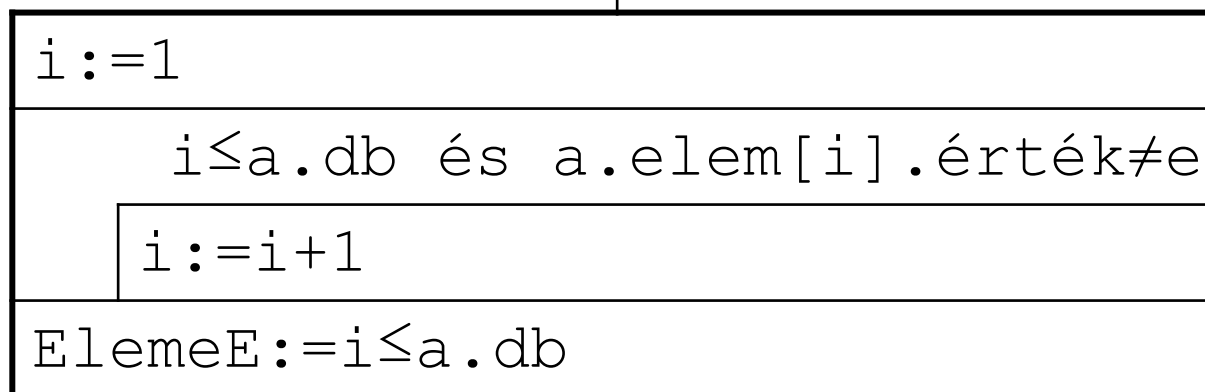
Műveletigény számítása:

Arányos a multihalmaz elemszámával (**keresés tétel**).



Multihalmaz típus

$\text{ElemE}(e, a) : \text{Logikai}$



Változó

$i : \text{Egész}$

Műveletigény számítása:

Arányos a multihalmaz elemszámával (eldöntés tétel).



Multihalmaz típus

Multiplicitás(a, e) : **Egész**

<i>i</i> := 1	
<i>i</i> ≤ a.db és a.elem[<i>i</i>].érték ≠ e	
<i>i</i> := <i>i</i> + 1	
<i>i</i> ≤ a.db	
I	N
Multiplicitás := a.elem[<i>i</i>].multi	Multiplicitás := 0

Változó
i : Egész

Műveletigény számítása:

Arányos a multihalmaz elemszámával (**keresés tétel**).



Multihalmaz típus

Halmazelem
típusú:
(érték, multi)

BenneE (e, a) : **Logikai**

i := 1

i ≤ a.db és a.elem[i].érték ≠ e.érték

i := i + 1

BenneE := i ≤ a.db és e.multi ≤ a.elem[i].multi

Változó

i : Egész

Műveletigény számítása:

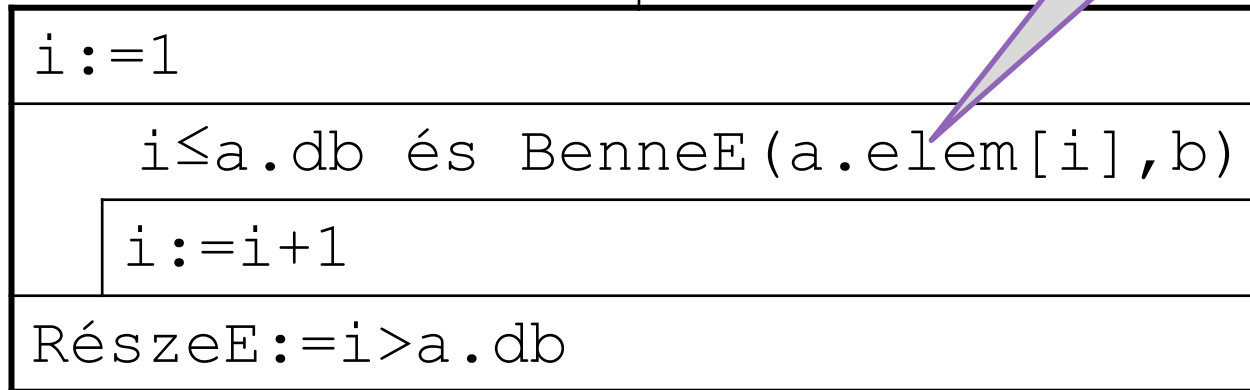
Arányos a multihalmaz elemszámával (keresés tétel).



Multihalmaz típus

Halmazelem
típusú:
(érték, multi)

RészeE(a, b) : **Logikai**



Változó

i : Egész

Műveletigény számítása:

A külső ciklus az 'a', a BenneE műveletben levő belső ciklus a 'b' multihalmaz elemszámáig fut le, azaz a futási idő a két multihalmaz elemszáma szorzatával arányos.



Multihalmaz típus

Unió (a, b)

Válto

i, j:

c:Mu

c := a

i = 1 .. b.db

j := 1

j ≤ a.db és b.elem[i].érték ≠ a.elem[j].érték

j := j + 1

I

j > a.db

N

c.db := c.db + 1

c.elem[c.db] :=
b.elem[i]

c.elem[j].multi :=

c.elem[j].multi +
b.elem[i].multi

Unió := c



Multihalmaz típus

Max (a, b)

Válto

i, j:

c: Mu

c := a

i = 1 .. b.db

j := 1

j ≤ a.db és b.elem[i].érték ≠ a.elem[j].érték

j := j + 1

I	j > a.db	N
c.db := c.db + 1 c.elem[c.db] := b.elem[i]	b.elem[i].multi > c.elem[j].multi	N
	c.elem[j].multi := b.elem[i].multi	—

Max := c



Multihalmaz típus

Metszet (a, b)

c.db := 0

i = 1 .. a.db

j := 1

j ≤ b.db és b.elem[j].érték ≠ a.elem[i].érték

j := j + 1

j ≤ b.db

I

N

c.db := c.db + 1; c.elem[c.db] := a.elem[i]

b.elem[j].multi < a.elem[i].multi

c.elem[c.db].multi := b.elem[j].multi

—

Metszet := c

vál.

i, j

c.db



Multihalmaz típus ábrázolása₂

Darabszám vektor:

Típus

Multihalmaz (Elemtípus) =

Tömb [**Min** 'Elemtípus' .. **Max** 'Elemtípus' : Egész]

Vegyünk fel egy annyi elemből álló sorozatot, amennyi a multihalmaz lehetséges elem fajtáinak száma!

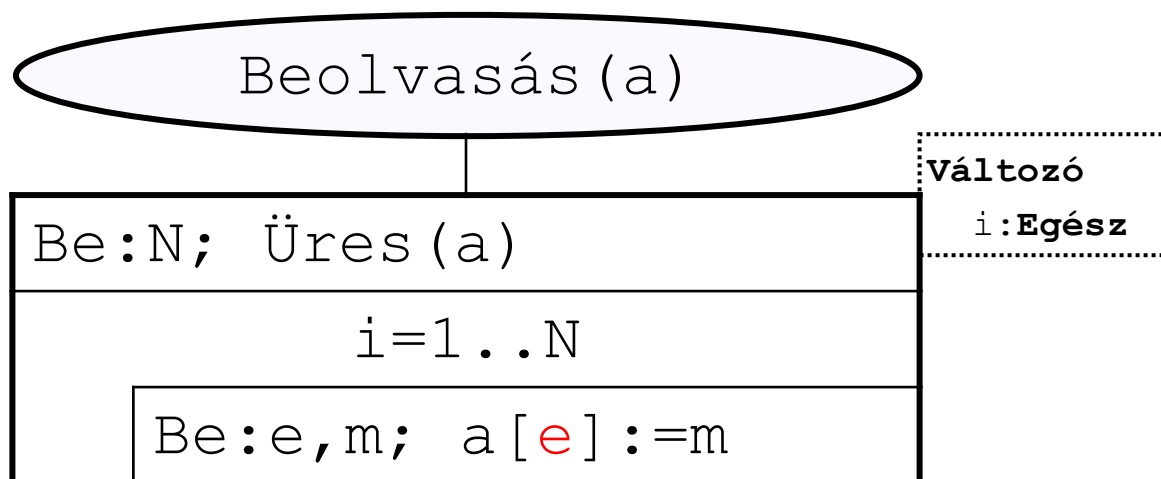
Legyen az i . elem x értékű, ha az i . lehetséges elem x -szer van benne van a multihalmazban, illetve 0, ha nincs benne!

Az Elemtípusnak diszkrétnek, azaz végesnek és „felsorolhatónak” kell lennie! Ilyenekkel fogunk indexelni!

Meggondolandó lenne ábrázolni a tárolt elemek számát is!



Multihalmaz típus



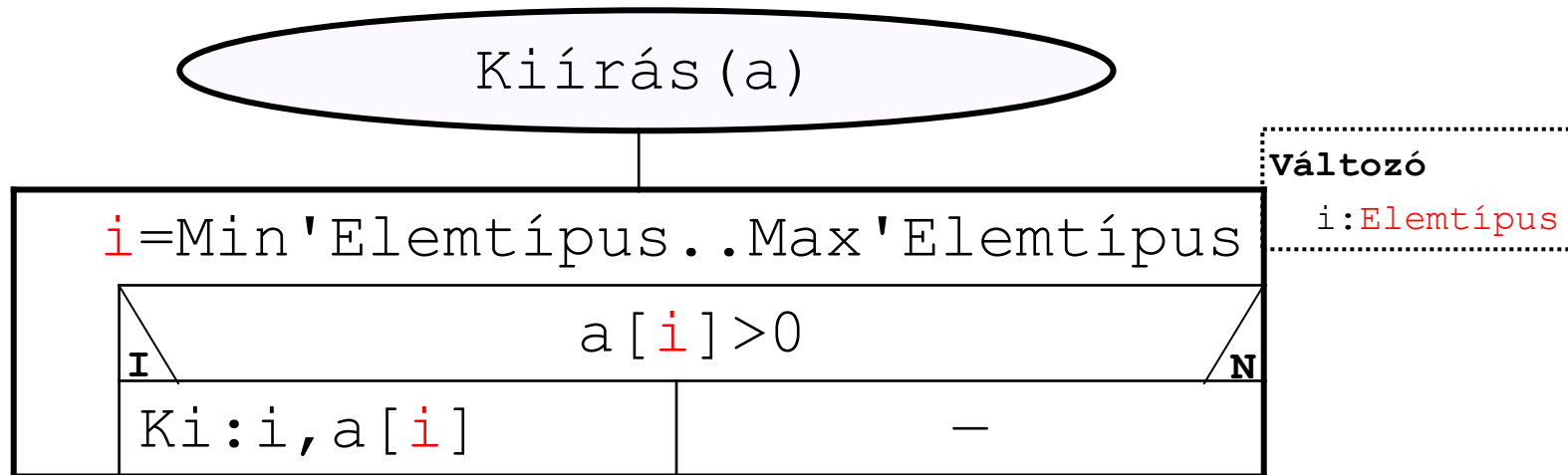
Műveletigény számítása:

A ciklus a multihalmaz elemértékeinek számaszor fut le, azaz a futási idő a multihalmaz elemszámával arányos.

A többi elemet azonban „0-ra kell állítani”: Üres (a), ami az alaphalmaz számosságával arányos műveletigényű.



Multihalmaz típus

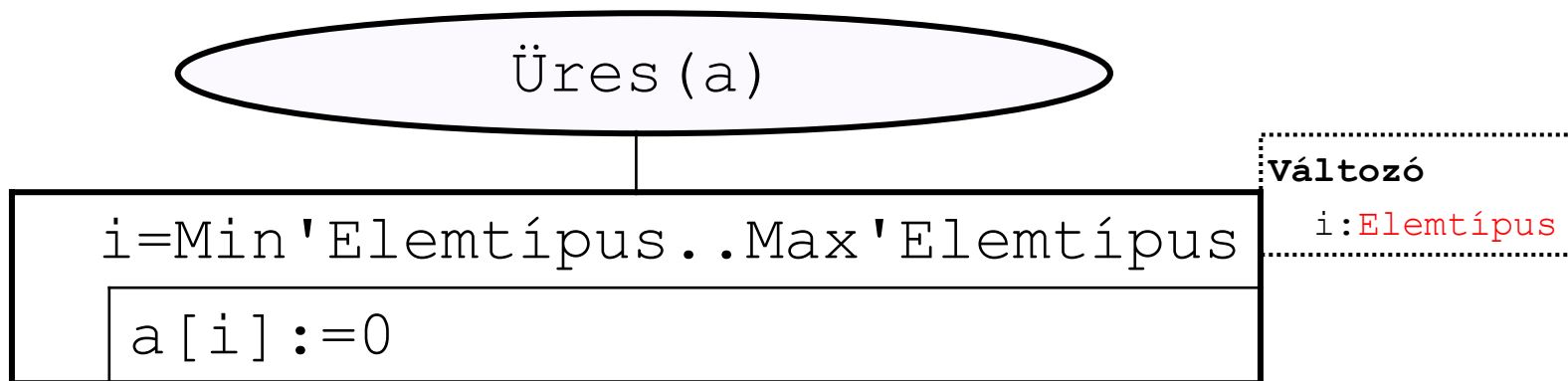


Műveletigény számítása:

A ciklus a multihalmaz elemtípusának számosságaszor fut le, azaz a futási idő a multihalmaz elemeinek maximális számával arányos.



Multihalmaz típus



Műveletigény számítása:

A ciklus a multihalmaz elemtípusának számosságaszor fut le, azaz a futási idő a multihalmaz elemeinek maximális számával arányos – hacsak nincs tömb 0-val feltöltésére művelet.



Multihalmaz típus

ÜresE(a) : **Logikai**

i := Min 'Elemtípus

i ≤ Max 'Elemtípus és a[i] = 0

i := i + 1

ÜresE := i > Max 'Elemtípus

Változó

i : **Elemtípus**

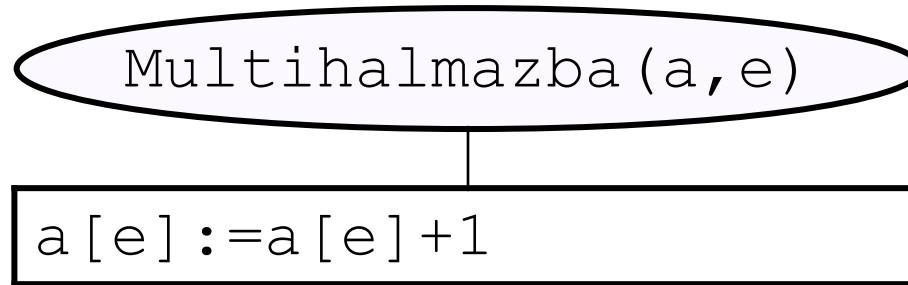
Műveletigény számítása:

A futási idő a multihalmaz elemtípusa számosságával arányos (eldöntés tétel).

Ha a multihalmazban lévő elemek számát is tárolnánk, akkor nem kellene ciklus.



Multihalmaz típus

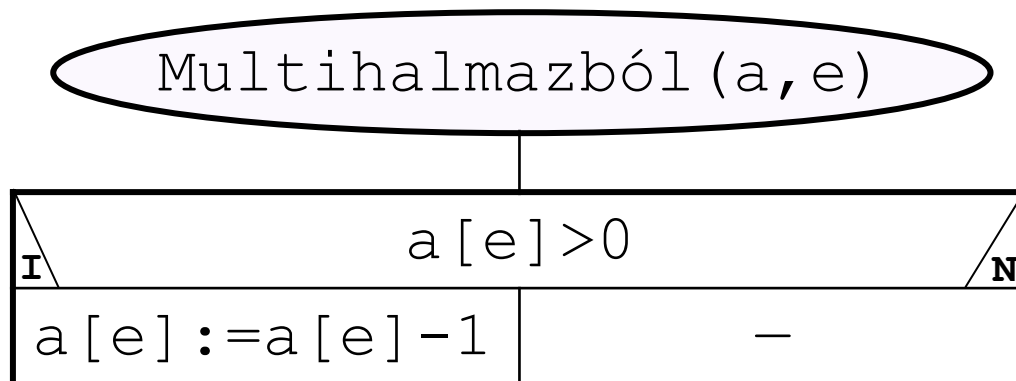


Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus



Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

$\text{ElemE}(e, a) : \text{Logikai}$

$\text{ElemE} := a[e] > 0$

Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

Multiplicitás (e, a) : **Egész**

Multiplicitás $:= a[e]$

Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

RészeE (a, b) : **Logikai**

i := Min 'Elemtípus

i ≤ Max 'Elemtípus és a[i] ≤ b[i]

i := i + 1

RészeE := i > Max 'Elemtípus

Változó

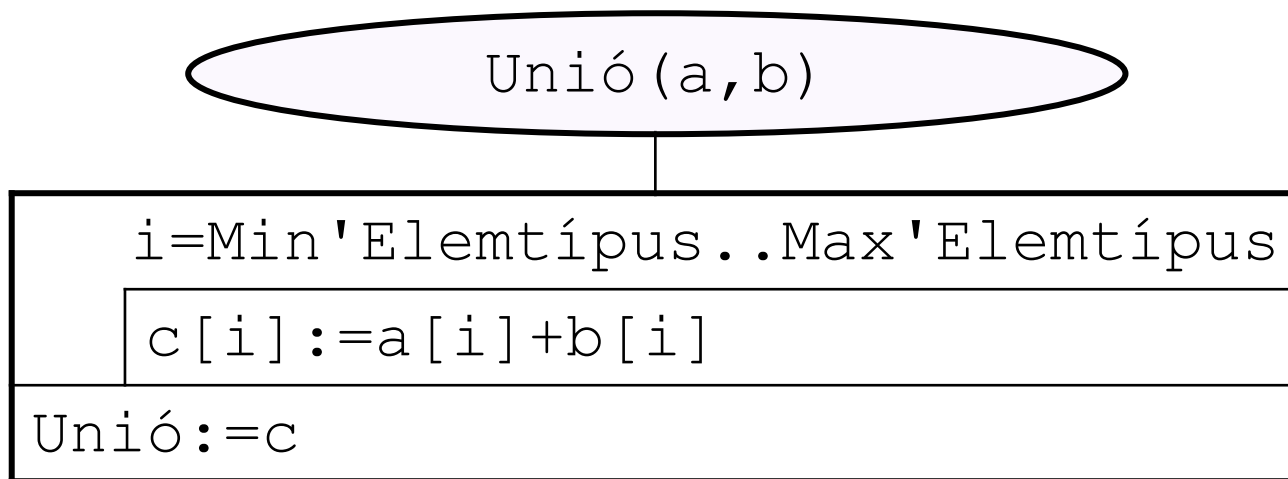
i : **Elemtípus**

Műveletigény számítása:

A futási idő a multihalmaz elemtípusa számosságával arányos (eldöntés tétel).



Multihalmaz típus



Változó

i : Elemtípus

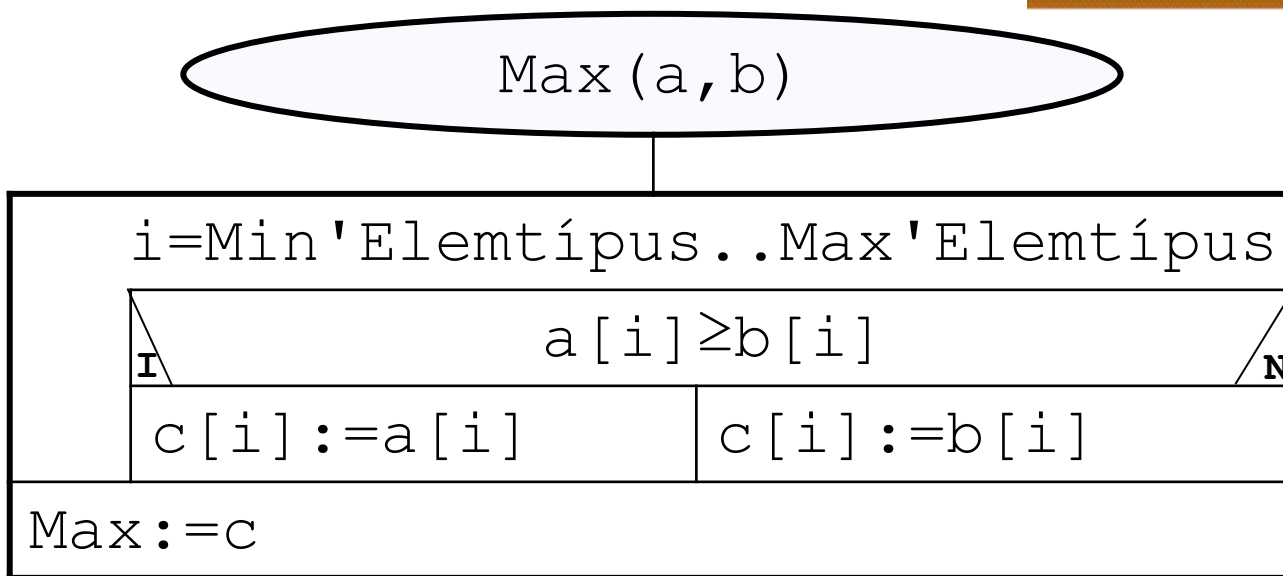
c : MultiHalmaz

Műveletigény számítása:

A futási idő a multihalmaz elemtípusa számosságával arányos (másolás tétel).



Multihalmaz típus



Változó

i: Elemtípus

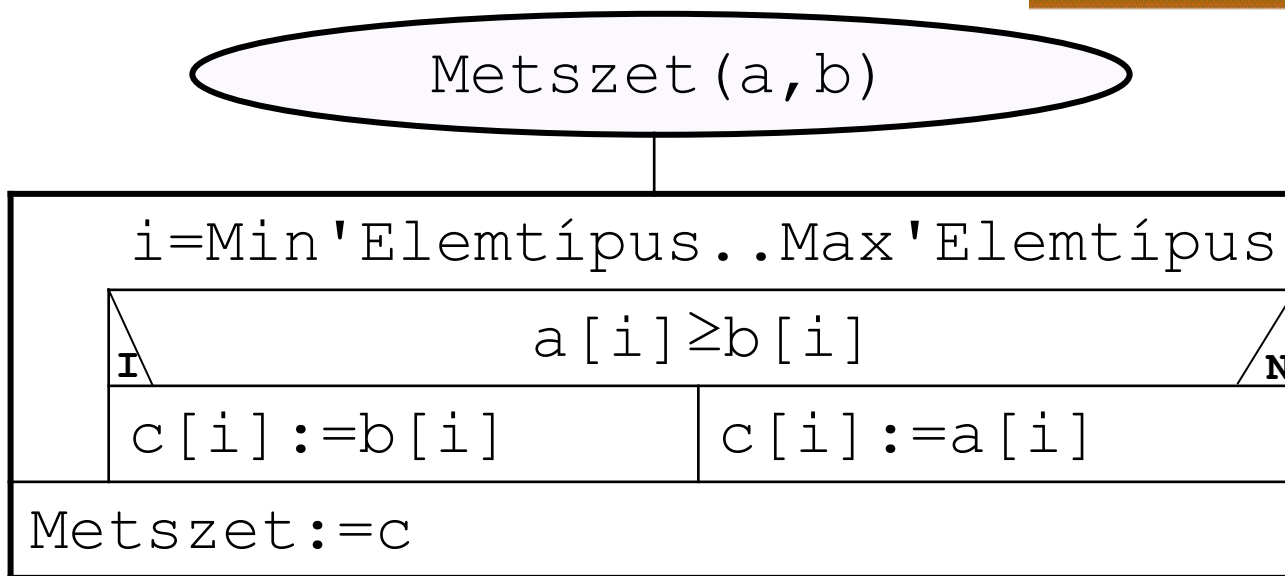
c: MultiHalma

Műveletigény számítása:

A futási idő a multihalmaz elemtípusa számosságával arányos (másolás tétel).



Multihalmaz típus



Változó

i: Elemtípus

c: MultiHalma

Műveletigény számítása:

A futási idő a multihalmaz elemtípusa számosságával arányos (másolás tétel).



Visszatekintés

➤ Halmazos tételek

➤ Metszet

➤ Unió

➤ Halmaz

➤ Halmaz típus elemek felsorolásával

➤ Halmaz típus darabszám vektorral

➤ Halmaz általánosítása: Multihalmaz

➤ Multihalmaz típus elemek felsorolásával

➤ Multihalmaz típus darabszám vektorral

