Programozás
9. előadás

Tartalom

- Programozási tételek általánosítása
  - Összegzés / Feltételes összegzés
  - Megszámolás
  - Maximum-kiválasztás / Feltételes maximum-keresés
  - Kiválasztás
  - Keresés / eldöntés
- Halmaz általánosítása: Multihalmaz
  - Multihalmaz típus elemek felsorolásával
  - Multihalmaz típus darabszám vektorral

Programozási tételek általánosítása

Cél: a programozási tételeket átalakítás nélkül lehessen használni feladatok megoldására.

Módszer: függvények célszerű paraméterezése és behelyettesítése.

Elv: N elemű tömb helyett egy $[e,u]\subset\mathbb{Z}$ intervallumon értelmezett függvény adja a sorozat elemeit, ahol a függvény általában egy tömbre épít (de akár több elemre is): $A:\mathbb{Z}\to H$

Például annak eldöntése, hogy egy tömb monoton növekvő-e, a [2,N] index-intervallumon vizsgálja, hogy a tömb minden i-edik eleme nagyobb-e az előzőnél (azaz a tulajdonság nem egy elemre vonatkozik).

A továbbiakban nem teljes programokat, hanem alprogramokat (függvény/eljárás) specifikálunk!

Összegzés – intervallumon

H=valamilyen számhalmaz

helyett

$e,u \in \mathbb{Z}$

$A:\mathbb{Z} \rightarrow H$

Bemenet: $N \in \mathbb{N}$,

$X_{1..N} \in H^N$

Kimenet: $S \in H$

Előfeltétel: –

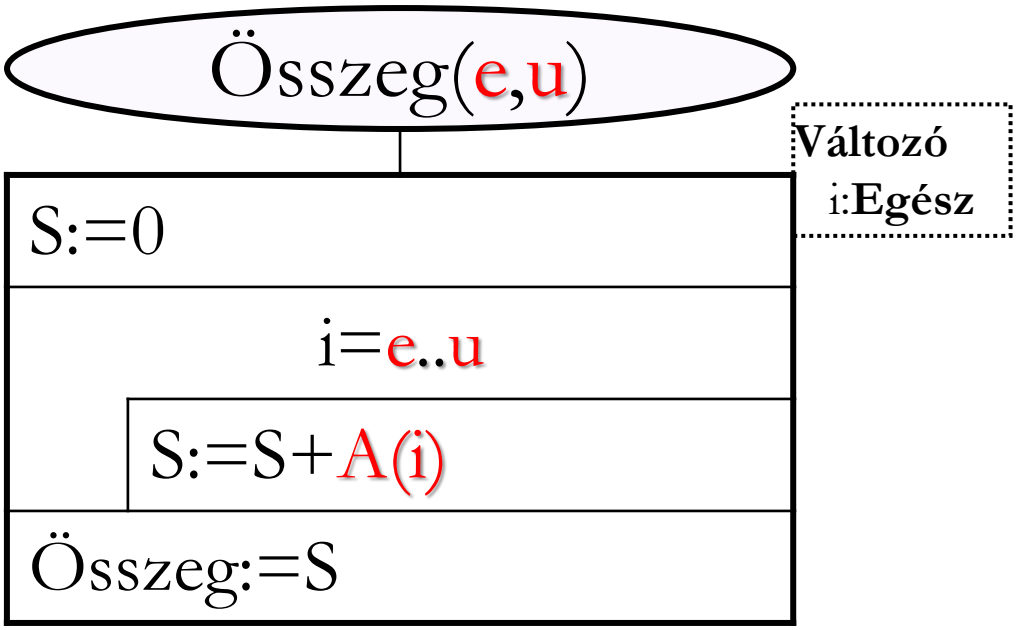
Utófeltétel: $S = \sum_{i=1}^{N} X_i$ → $S = \sum_{i=e}^{u} A(i)$

Az alapváltozatban:

$e=1$, $u=N$, $A(i)=X_i$

Összegzés – intervallumon

Algoritmus:



Az i. elemet megadó A(i) helyére minden esetben behelyettesítendő a megfelelő képlet, az e és az u pedig paraméter.

Összegzés – intervallumon

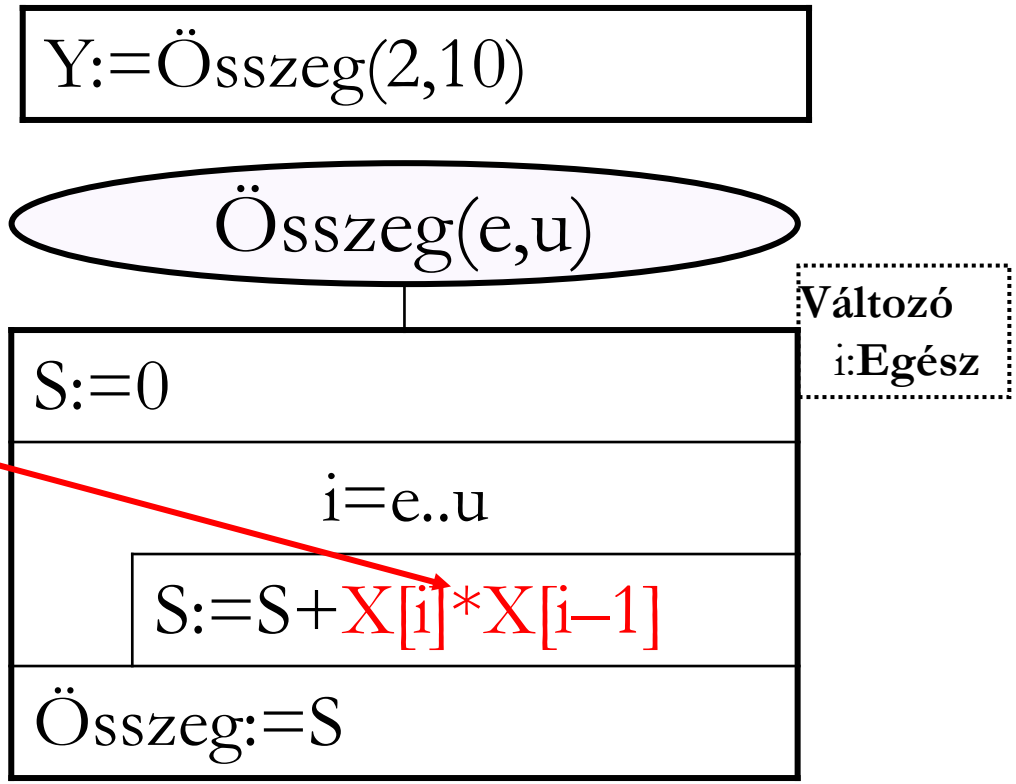
Példa$_1$:

Utófeltétel:

$$Y = \sum_{i=2}^{10} X_i * X_{i-1}$$

A(i)

Utófeltétel | Algoritmusok



A behelyettesített X tömb globális változó.

Összegzés – intervallumon

Példa$_2$:

Utófeltétel

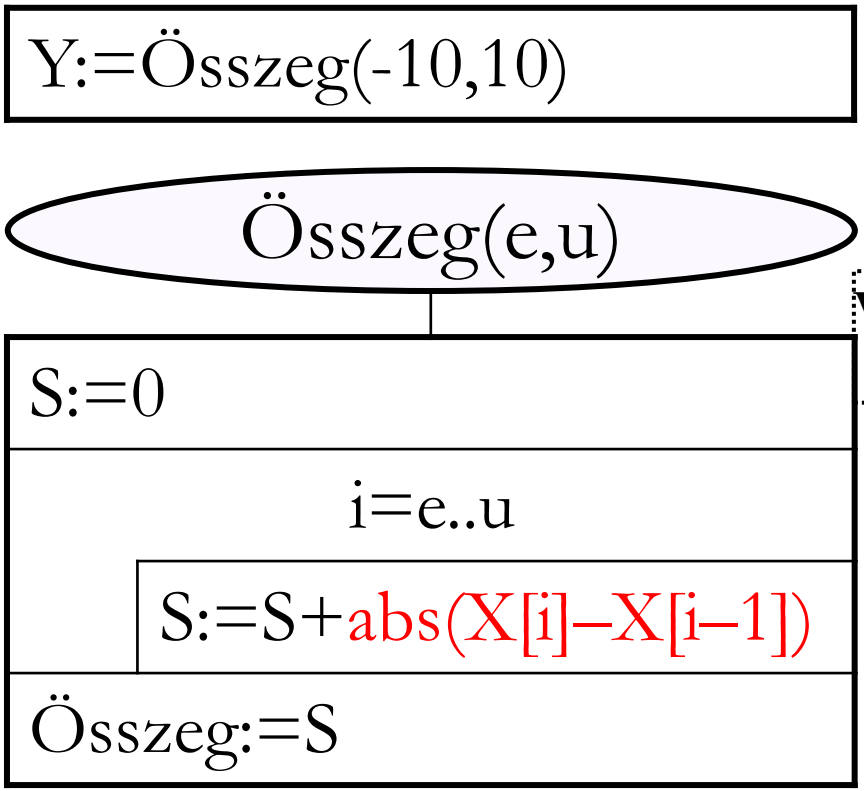
Algoritmusok

Utófeltétel:

$$Y = \sum_{i=-10}^{10} |X_i - X_{i-1}|$$

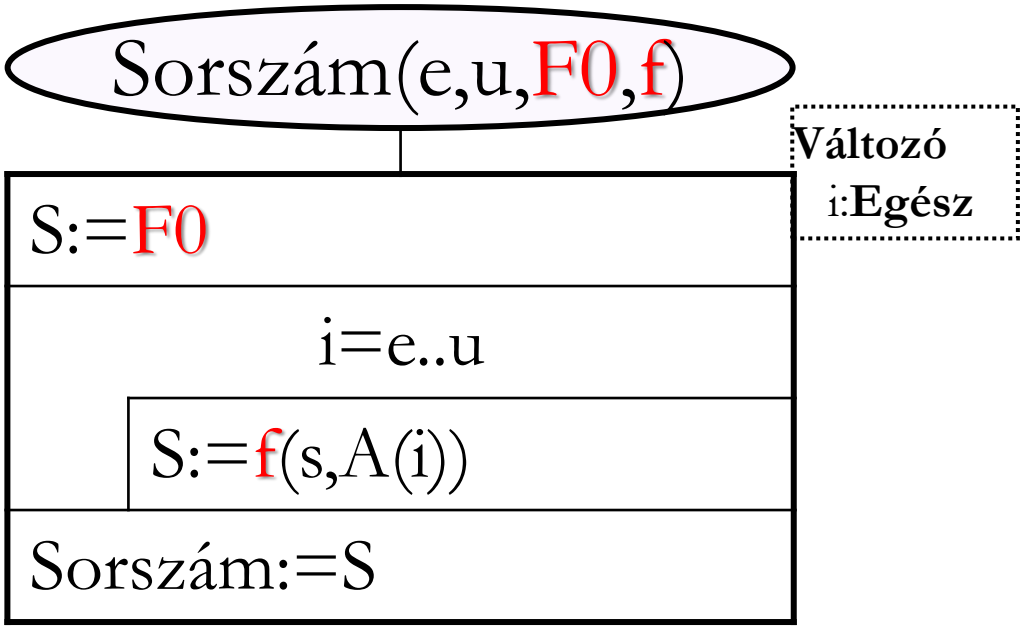
Feltettük:

az X sorozat –11-től indexelhető.



Sorozatszámítás – intervallumon

Algoritmus (általánosan):



Új paraméterek:

- a nullelem (F0) és a
- f a kapcsolódó 2-változós függvény (műveleti operátor).

Feltételes összegzés – intervallumon

helyett

Bemenet: $N \in \mathbb{N}$, $X_{1..N} \in H^N$, $T: H \to L$

$e, u \in \mathbb{Z}$, $A: \mathbb{Z} \to H$, $T: \mathbb{Z} \to L$

Kimenet: $S \in H$

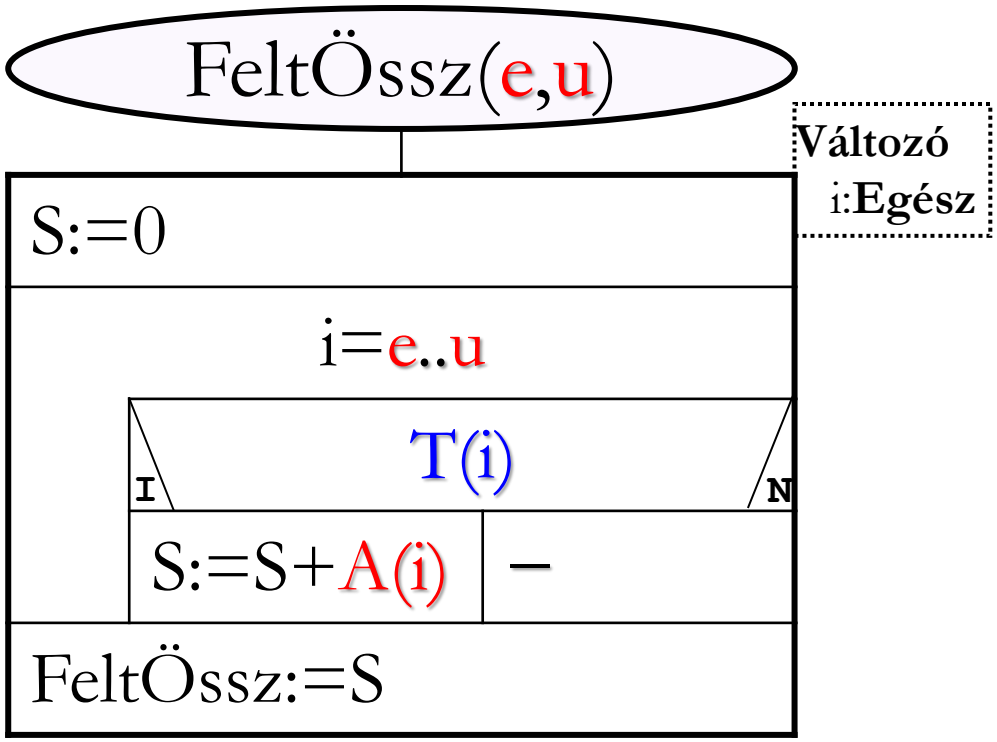
Előfeltétel: –

Utófeltétel: $S = \sum_{i=1}^{N} X_i$ $\sum_{T(X_i)}$

$S = \sum_{i=e}^{u} A(i)$ $\sum_{T(i)}$

Feltételes összegzés – intervallumon

Algoritmus:



Az i. elemet megadó A(i), az i. elem T-tulajdonság teljesülését megadó T(i) helyére minden esetben behelyettesítendő a megfelelő képlet.

Feltételes összegzés – intervallumon

Példa:

Utófeltétel

Feladat: csúcsok összege…

Utófeltétel:

$$CsÖ = \sum_{i=2}^{N-1} X_i$$

$X_{i-1} < X_i$ és $X_i > X_{i+1}$

Algoritmusok

CsÖ:=FeltÖssz(2,N–1)

FeltÖssz(e,u)

Változó
i:Egész

S:=0

i=e..u

X[i–1]<X[i] és
X[i]>X[i+1]

I N

S:=S+X[i] | –

FeltÖssz:=S

Megszámolás – intervallumon

Specifikáció:

Bemenet: $N \in \mathbb{N}$,
$X_{1..N} \in \mathbb{H}^N$
$T: \mathbb{H} \to \mathbb{L}$

helyett

$e,u \in \mathbb{Z}$

$T: \mathbb{Z} \to \mathbb{L}$

Kimenet: $Db \in \mathbb{N}$

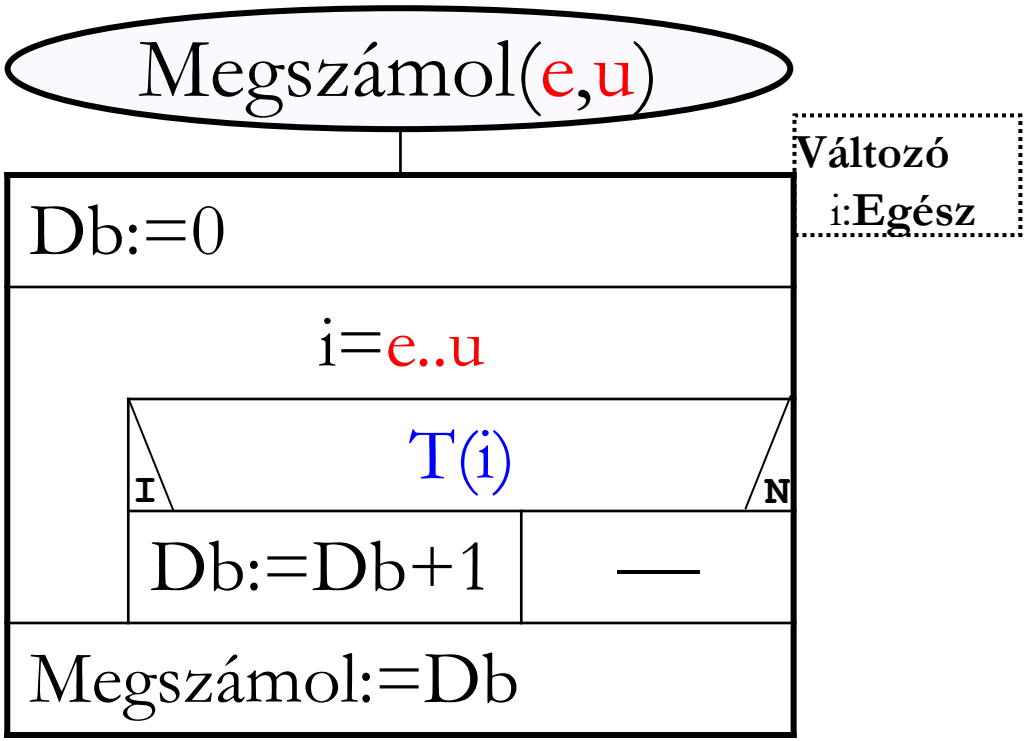
Előfeltétel: –

Utófeltétel: $Db = \sum_{\substack{i=1 \\ T(X_i)}}^{N} 1$ → $Db = \sum_{\substack{i=e \\ T(i)}}^{u} 1$

Megjegyzés: az X helyetti A függvényre explicite nincs szükség a $T(X_i)$-nek megfelelő T(A(i)) helyett írható T(i).

Megszámolás – intervallumon

Algoritmus:



Az i. elemet megadó T(i) helyére minden esetben behelyettesítendő a megfelelő képlet.

Megszámolás – intervallumon

Példa:

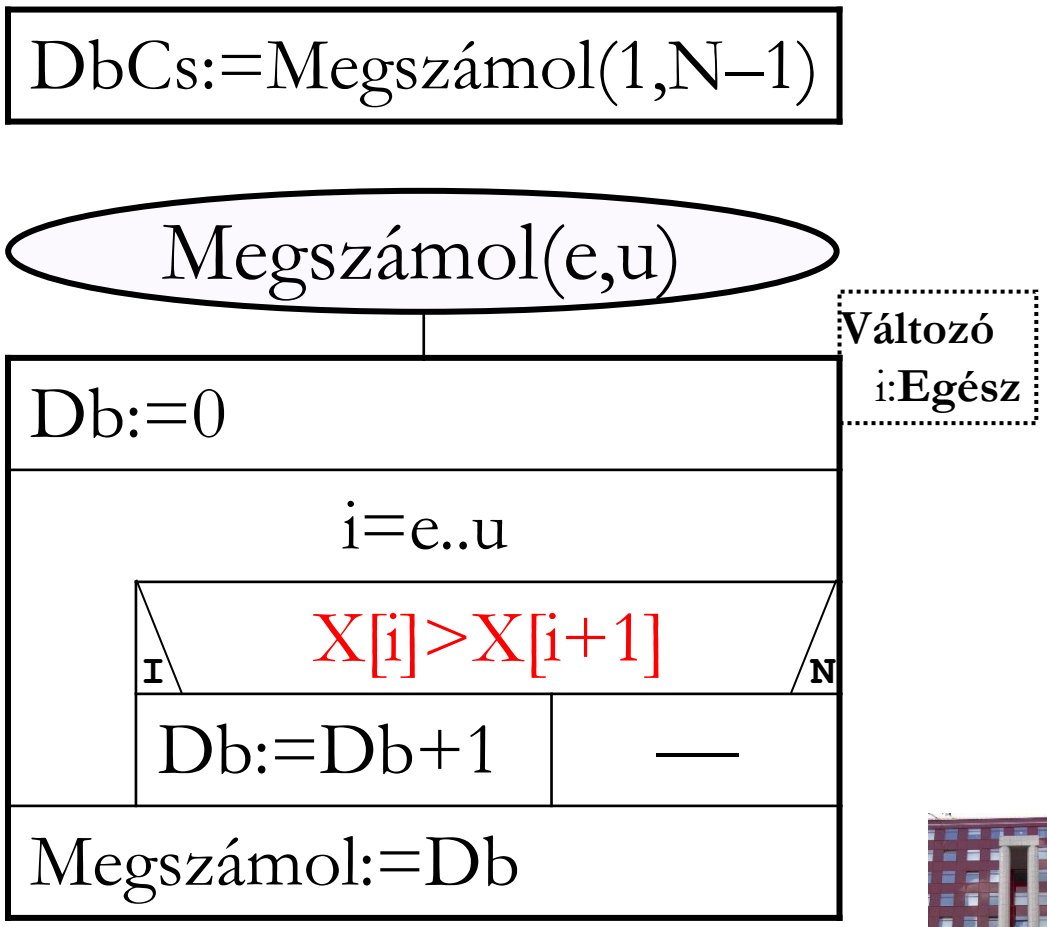
Utófeltétel

Utófeltétel:

$$DbCs = \sum_{i=1}^{N-1} 1$$

$X_i > X_{i+1}$

Algoritmusok



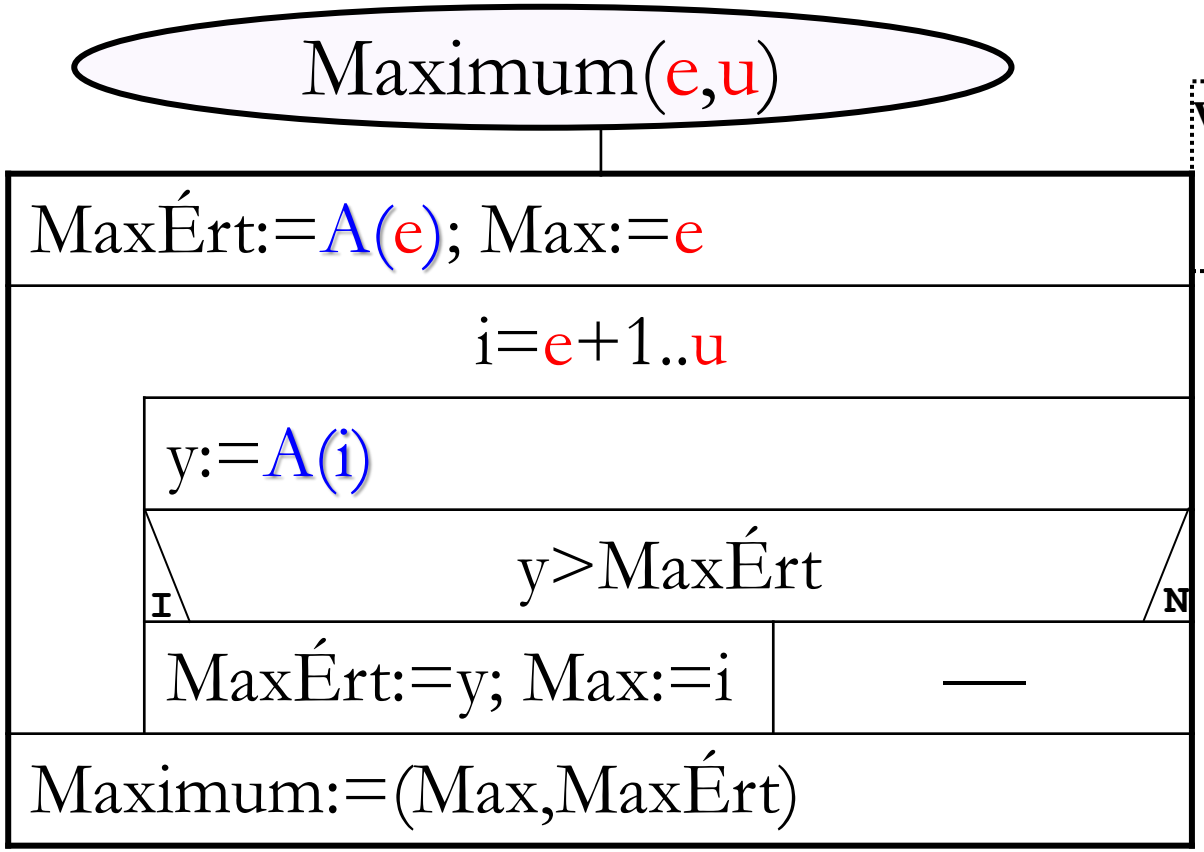
Maximum-kiválasztás – intervallumon

Specifikáció:

		helyett
Bemenet:	$N \in \mathbb{N}$,	$e,u \in \mathbb{Z}$,
	$X_{1..N} \in H^N$	$A:\mathbb{Z} \rightarrow H$
Kimenet:	$MaxÉrt \in H$,	
	$Max \in \mathbb{Z}$	
Előfeltétel:	$N>0$	$e \leq u$
Utófeltétel:	$1 \leq Max \leq N$ és	$e \leq Max \leq u$ és
	$MaxÉrt = X_{Max}$	$MaxÉrt = A(Max)$
	$\forall i(1 \leq i \leq N)$:	$\forall i(e \leq i \leq u)$:
	$MaxÉrt \geq X_i$	$MaxÉrt \geq A(i)$

Maximum-kiválasztás – intervallumon

Algoritmus:



Változó
i:**Egész**
y:TH

A(i) helyébe az adott képlet behelyettesítendő (2 helyen).

Maximum-kiválasztás

Implementálási kitérő:

Sok esetben csak a maximális értékű elem indexére, vagy az értékére van szükség.

Ilyenkor az utolsó értékadás

Maximum:=(Max,MaxÉrt)

helyett a

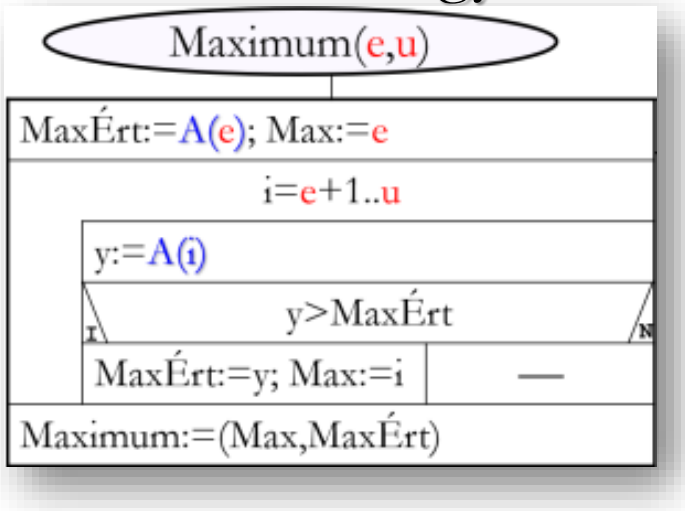
Maximum:=Max

vagy

Maximum:=MaxÉrt

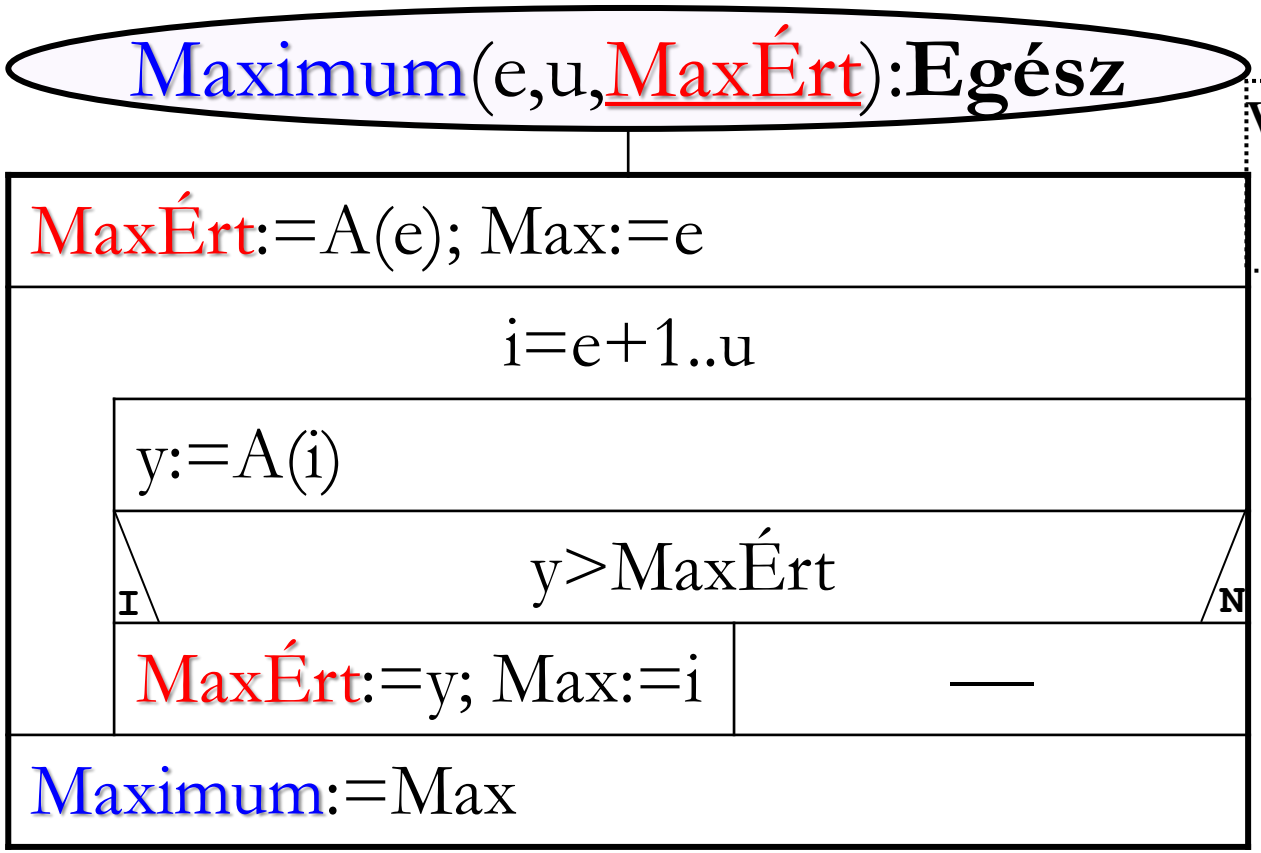
függvény értékadás szerepel a függvény végén.

Ha mindkettő kell, akkor kérdéses, hogy mi legyen a függvény (egyetlen) értéke? Lehet mindkettő?



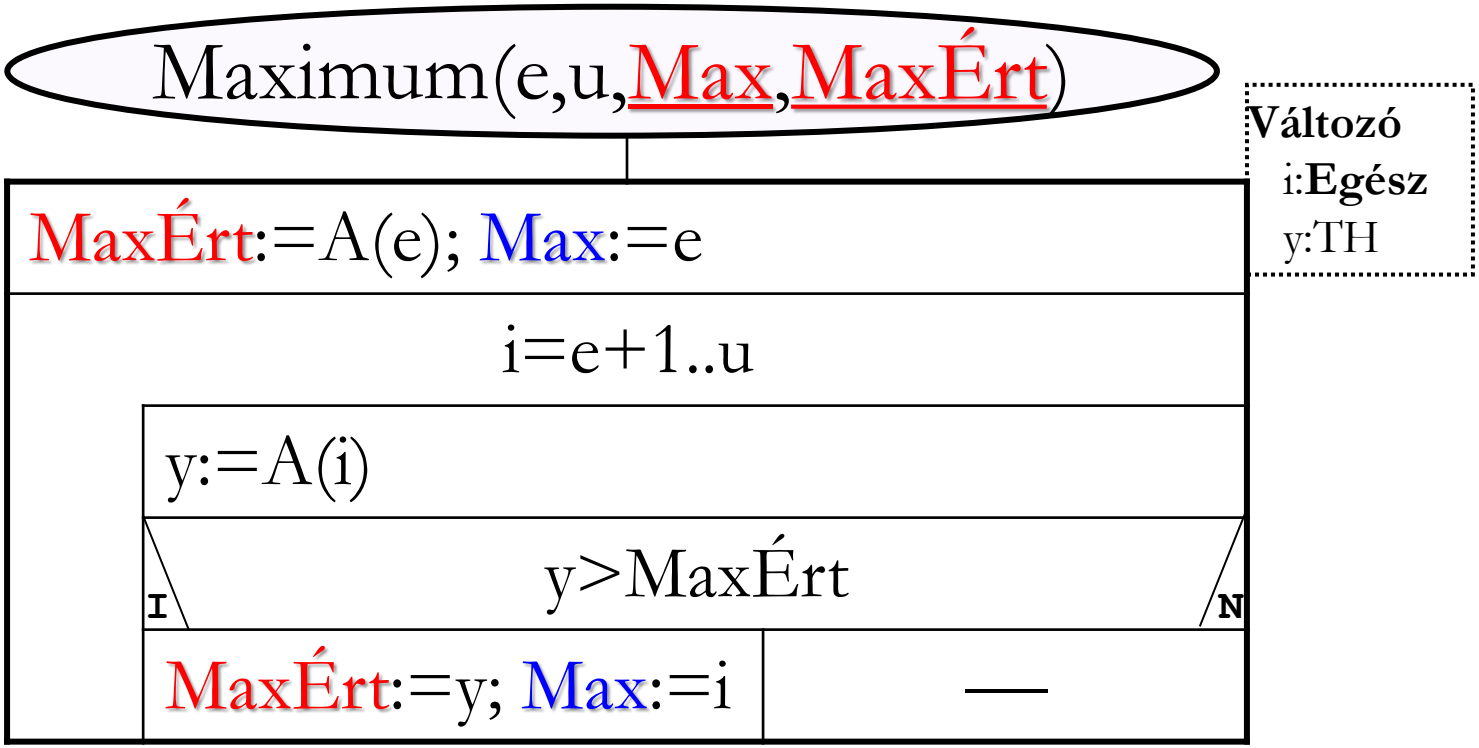
Maximum-kiválasztás – intervallumon

Algoritmus: (kimenő értékek szétválasztása: függvény kimenő paraméterrel)



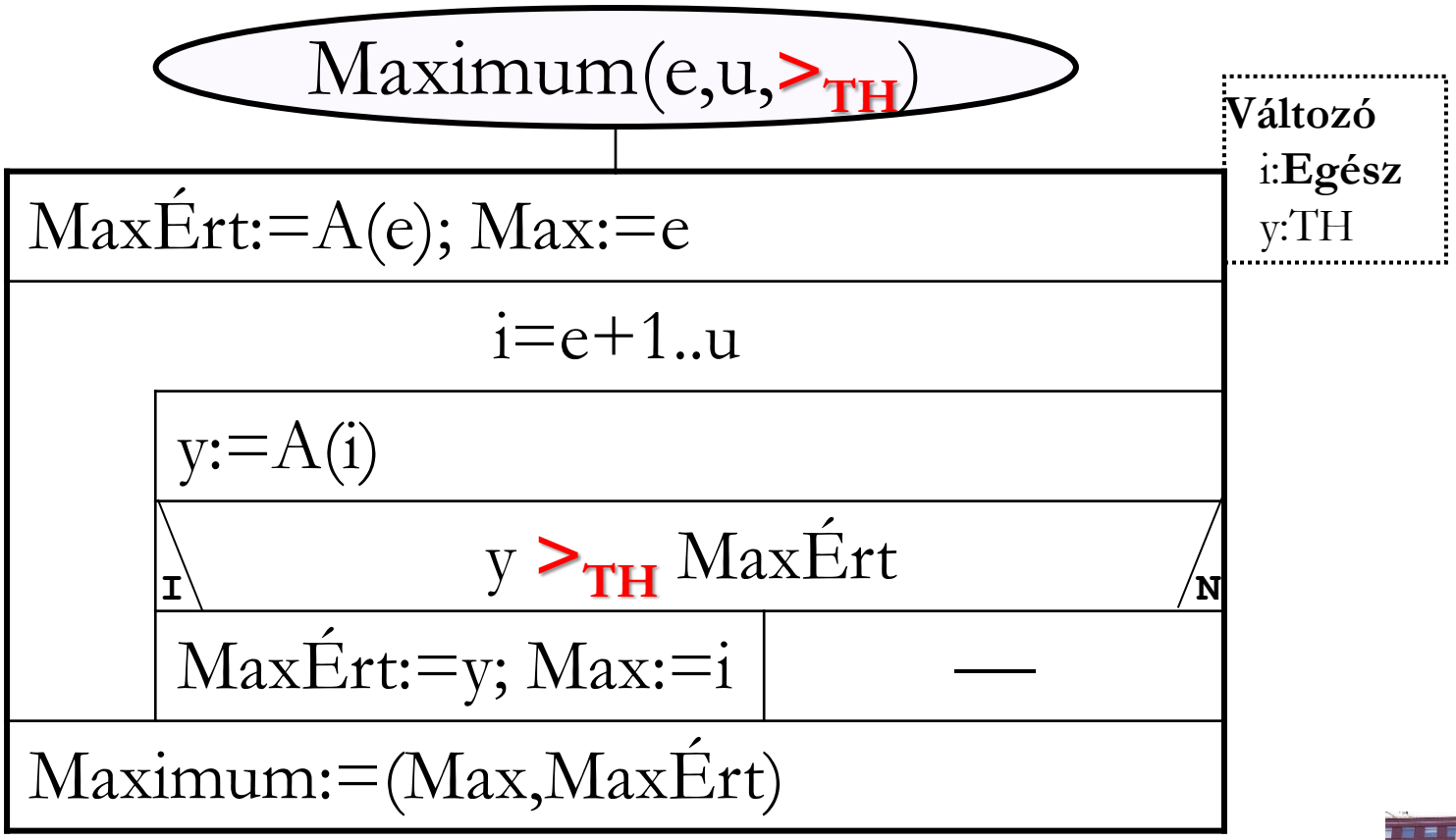
Maximum-kiválasztás – intervallumon

Algoritmus: (eljárás két kimenő paraméterrel)



Maximum-kiválasztás – intervallumon

Algoritmus (általános relációval):



Maximum-kiválasztás – intervallumon

Példa:

Első születésnapos …

Specifikáció$_2$:
- Bemenet: $N \in \mathbb{N}$, $Szül_{1..N} \in Dátum^N$, $Dátum = hó \times nap$, hó,nap=N
- Kimenet: Első$\in \mathbb{N}$, Mikor$\in$Dátum
- Előfeltétel: N>0 …
- Utófeltétel: $(Első,Mikor) = \min_{i=1}^{N} Szül_i$
- Definíció:
  $\leq$:Dátum×Dátum→L
  d1≤d2 ↔ d1.hó<d2.hó vagy
  d1.hó=d2.hó és d1.nap≤d2.nap

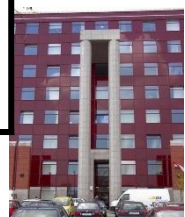
(Első,Mikor):=Maximum(1,N,Előbb)

A Maximum függvényben az A(i) törzse egyszerűen Szül[i] (azaz az i. ember születésnapja).

Az $<_{Dátum}$ implementálása függvényként:

Előbb(x,y:TDátum)
Operátor x<y

Előbb:=x.hó<y.hó vagy
x.hó=y.hó és x.nap≤y.nap



Feltételes maximumkeresés – intervallumon

Specifikáció:

Bemenet: $e,u \in \mathbb{Z}$,
$A:\mathbb{Z} \to \mathbb{H}$,
$T:\mathbb{Z} \to \mathbb{L}$

Példa: lokális csúcsok minimuma; itt a tulajdonság indextől függ!

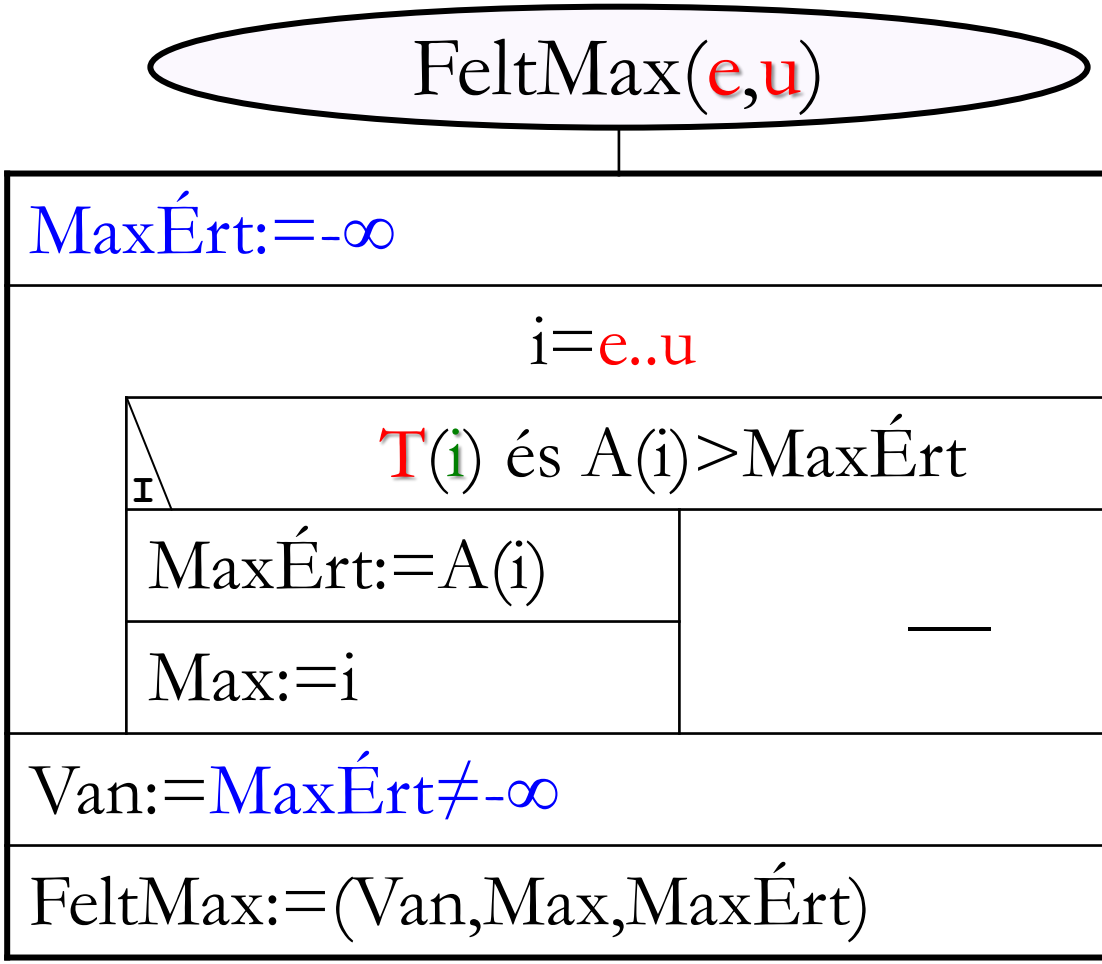
Kimenet: $Van \in \mathbb{L}$, $Max \in \mathbb{Z}$, $MaxÉrt \in \mathbb{H}$

Előfeltétel: $e \leq u$

Utófeltétel: $Van = \exists i(e \leq i \leq u): T(i)$ és
$Van \to (e \leq Max \leq u$ és $MaxÉrt = A(Max)$ és
$T(Max)$ és
$\forall i(e \leq i \leq u): T(i) \to MaxÉrt \geq A(i))$

Feltételes maximumkeresés – intervallumon

Algoritmus:



Változó
i:Egész

Feltételes maximumkeresés

Implementálási kitérő:

A feltételes maximumkeresést sokszor elágazás feltételében szeretnénk használni, ahol az elágazás attól függ, hogy van-e feltételes maximum.

A három értékből álló függvényérték nem szerepelhet feltételben!

Egy lehetséges megoldás az, hogy

- csak a logikai érték a feltételes maximumot meghatározó függvény értéke,
- a másik kettő pedig kimenő paraméter.

Kiválasztás – intervallumon

Bemenet: e,u ∈ℤ,
A:ℤ→ℍ,
T:ℤ→𝕃

Kimenet: Hol∈ℤ, Ért∈ℍ

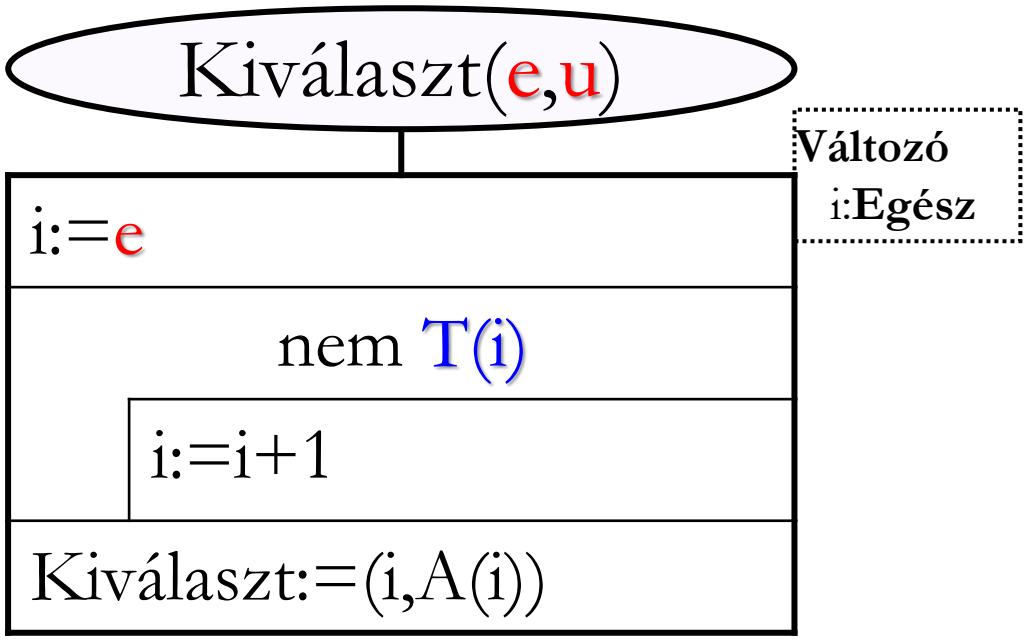
Előfeltétel: e≤u és ∃i (e≤i≤u): T(i)

Utófeltétel: e≤Hol≤u és T(Hol) és Ért=A(Hol)

Az eredeti Ind paraméter helyett, kifejezőbb a Hol azonosító

Kiválasztás – intervallumon

Algoritmus:



Az T(i) helyébe mindig az aktuális elemet megadó képlet behelyettesítendő.

Implementálási kitérő:

Az utolsó sor lehet: Kiválaszt:=i vagy Kiválaszt:=A(i) is.

Keresés – intervallumon

Bemenet: $e,u \in \mathbb{Z}$
$A:\mathbb{Z}\rightarrow\mathbb{H}$
$T:\mathbb{Z}\rightarrow\mathbb{L}$

Kimenet: $Van\in\mathbb{L}$, $Hol\in\mathbb{Z}$, $Ért\in\mathbb{H}$

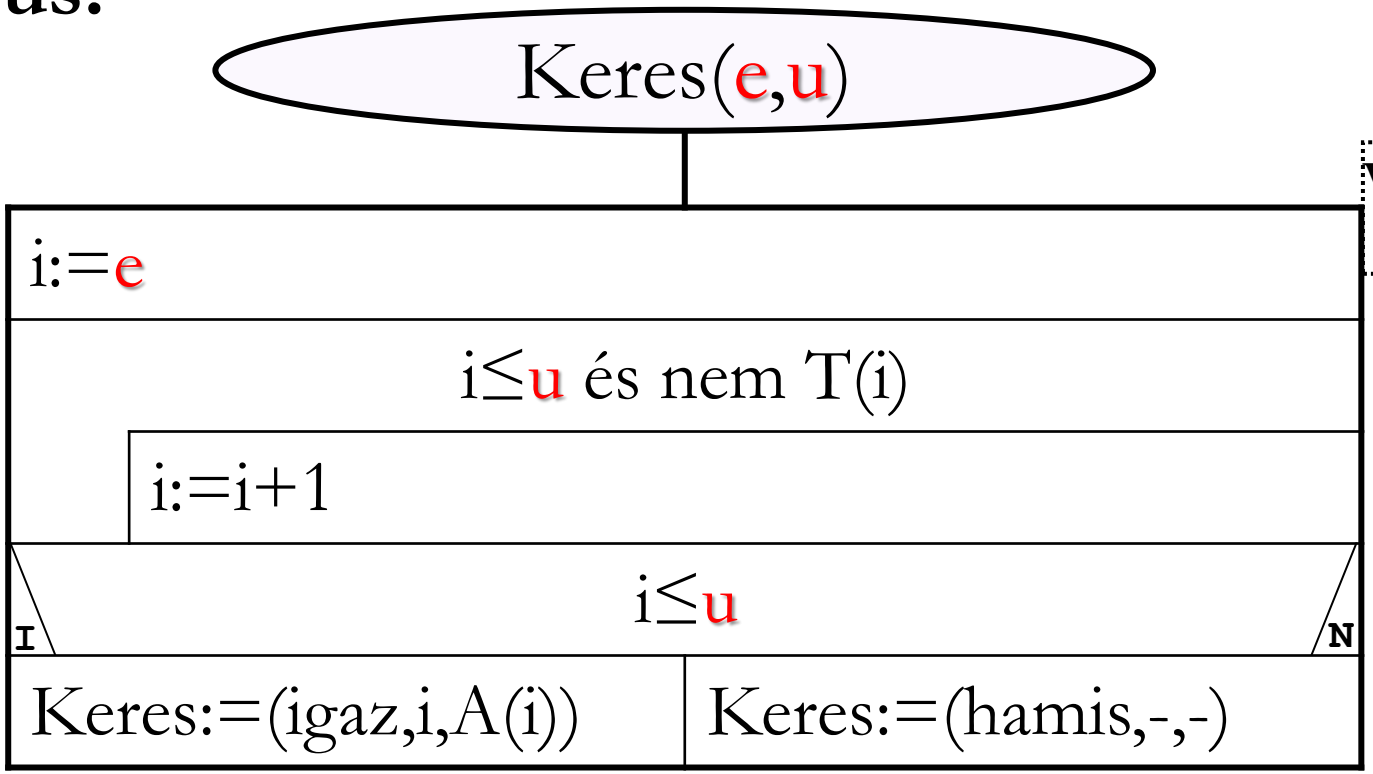
Előfeltétel: –

Utófeltétel: $Van=\exists i(e\leq i\leq u)$: $T(i)$ és
$Van\rightarrow e\leq Hol\leq u$ és $T(Hol)$ és $Ért=A(Hol)$

Az eredeti Ind paraméter helyett, kifejezőbb a Hol azonosító

Keresés – intervallumon

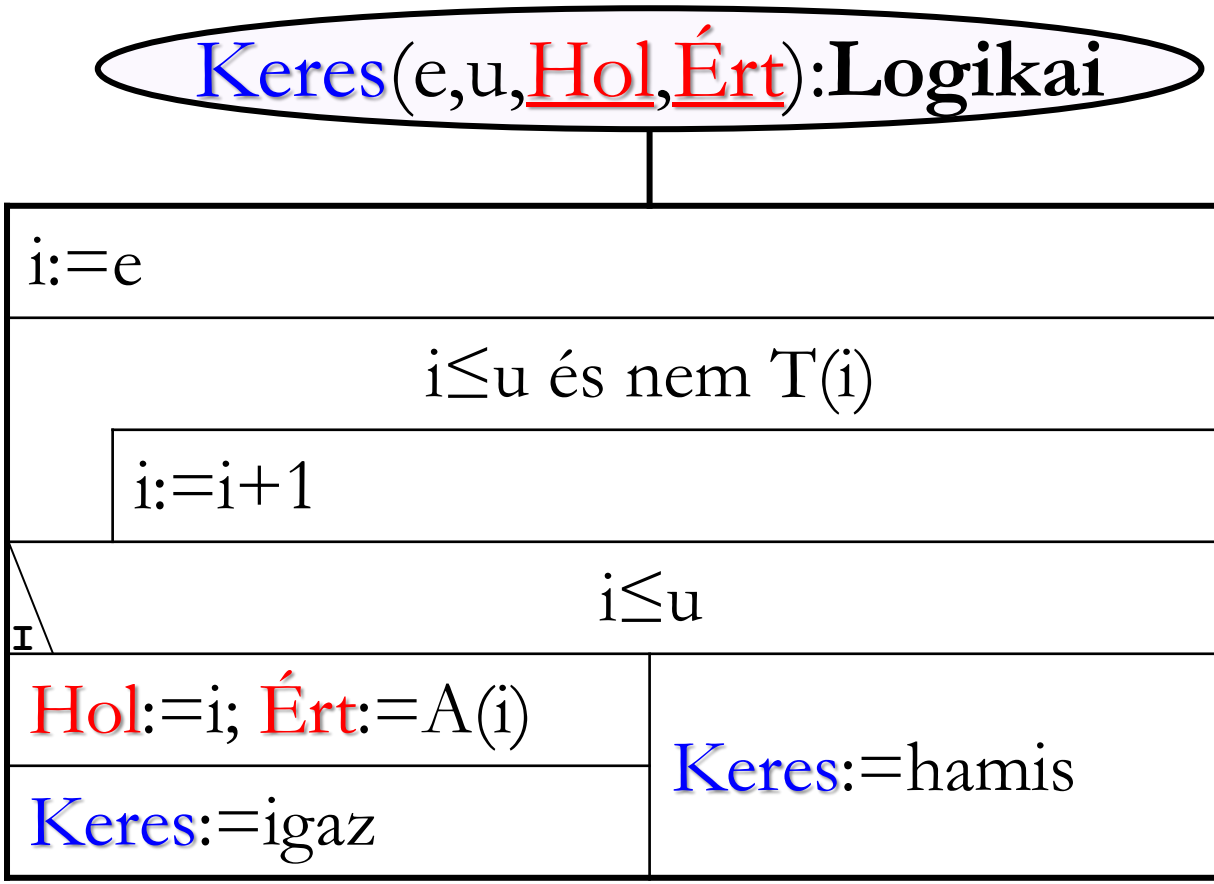
Algoritmus:



A keresésből eldöntés lesz, ha a függvény értéke csak a logikai érték.

Keresés – intervallumon

Algoritmus (függvény két kimenő paraméterrel):



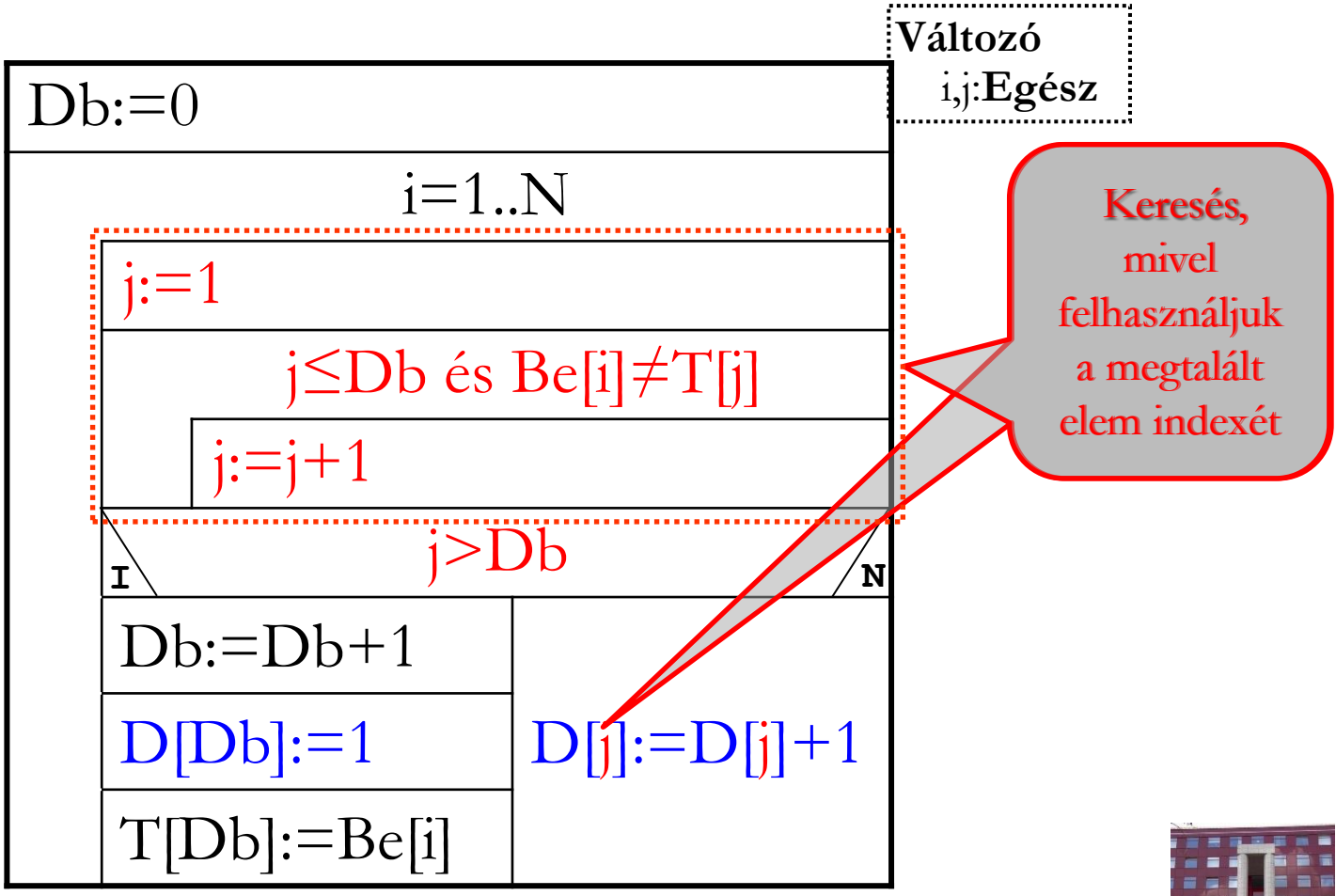
Sorozat → multihalmaz transzformáció

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→keresés**), s e közben számláljunk is (**megszámolás**)!

Sorozat → multihalmaz transzformáció



Multihalmaz típus

Értékhalmaz:

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
- Üres? (logikai értékű függvény).

Multihalmaz típus

Alapműveletek:

- eleme (egy elem benne van-e a multihalmazban) ($\in$)
- benne (egy elem legalább adott multiplicitással benne van-e a multihalmazban)
- multiplicitás (egy elem hányszoros multiplicitással van benne a multihalmazban)

Multihalmaz típus

Multihalmaz×Multihalmaz műveletek:

- metszet (∩) (értékek metszete, multiplicitások minimuma)
- unió (∪) (értékek uniója, multiplicitások összege)
- különbség (\ ) (értékek különbsége, multiplicitások különbsége; nincs benne egy elem, ha a multiplicitások különbsége 1-nél kisebb)
- max (multiplicitások maximuma),
- része (egyik multihalmaz részhalmaza-e a másiknak) (⊂ , ⊆)
- mindközös? (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 ábrázolása$_1$

Elemek felsorolása:

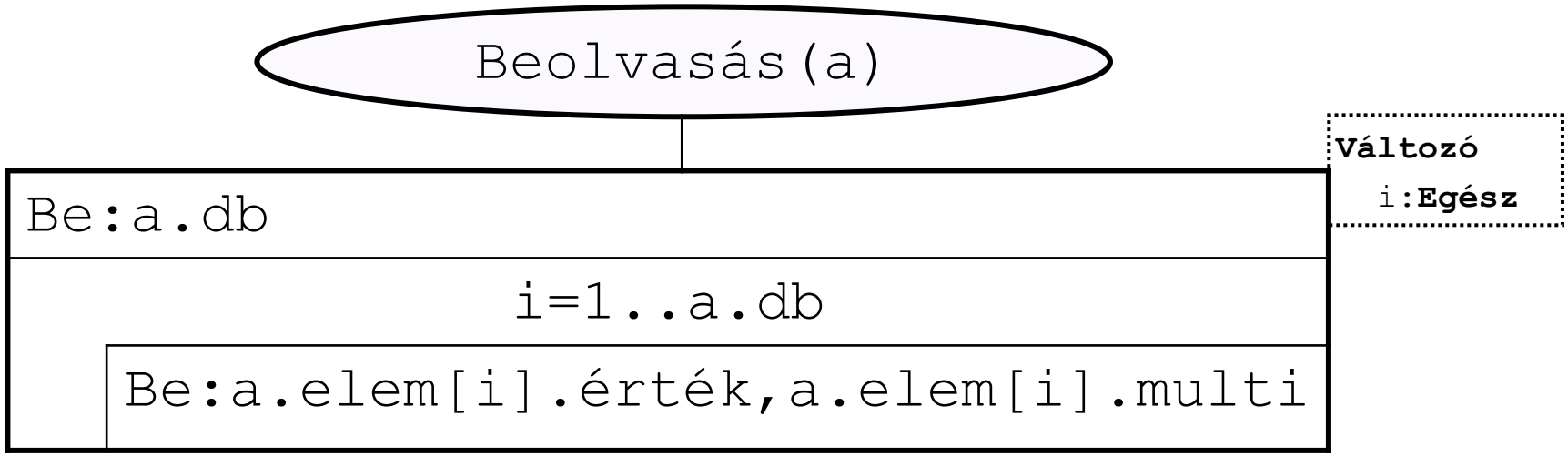
```
Típus
  Halmazelem(Elemtípus)=Rekord(érték:Elemtípus,
                               multi:Egész)

  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!

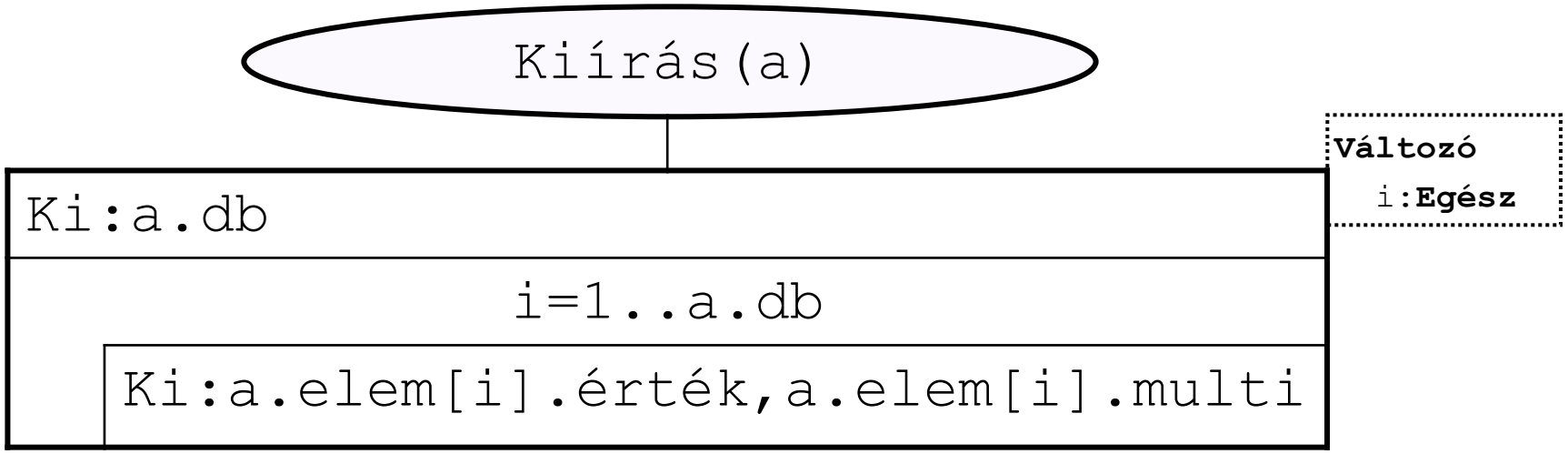
Multihalmaz típus



Műveletigény számítása:

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

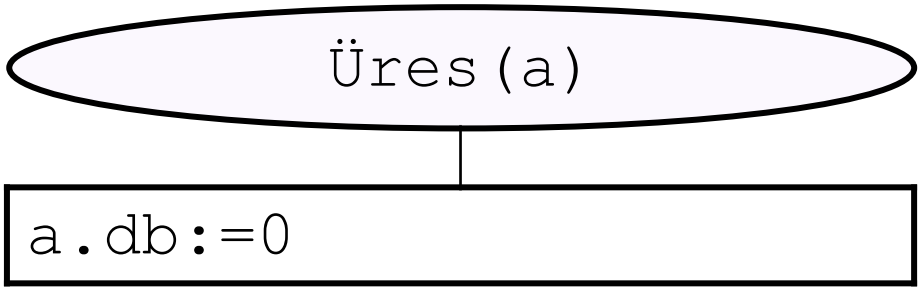
Multihalmaz típus



Műveletigény számítása:

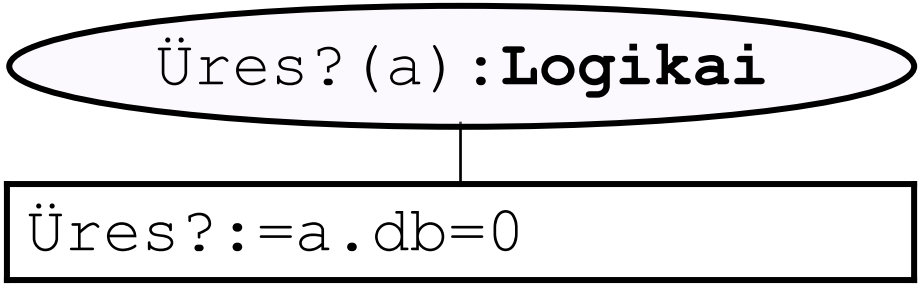
A ciklus a multihalmaz elemértékeinek számaszor fut le, azaz a futási idő a multihalmaz elemszámával arányos.

Multihalmaz típus



Műveletigény számítása:

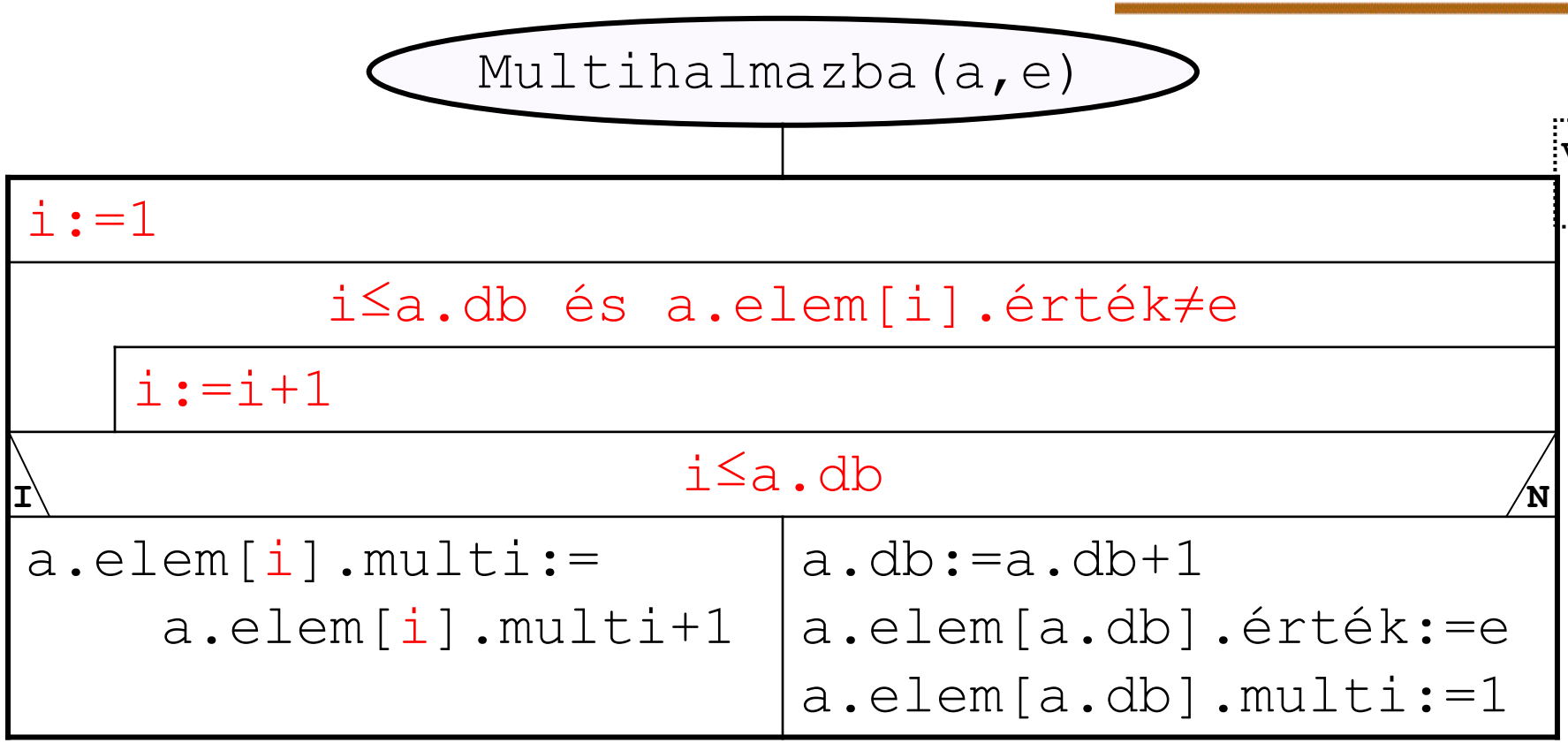
Nem függ a multihalmaz elemszámától.



Műveletigény számítása:

Nem függ a multihalmaz elemszámától.

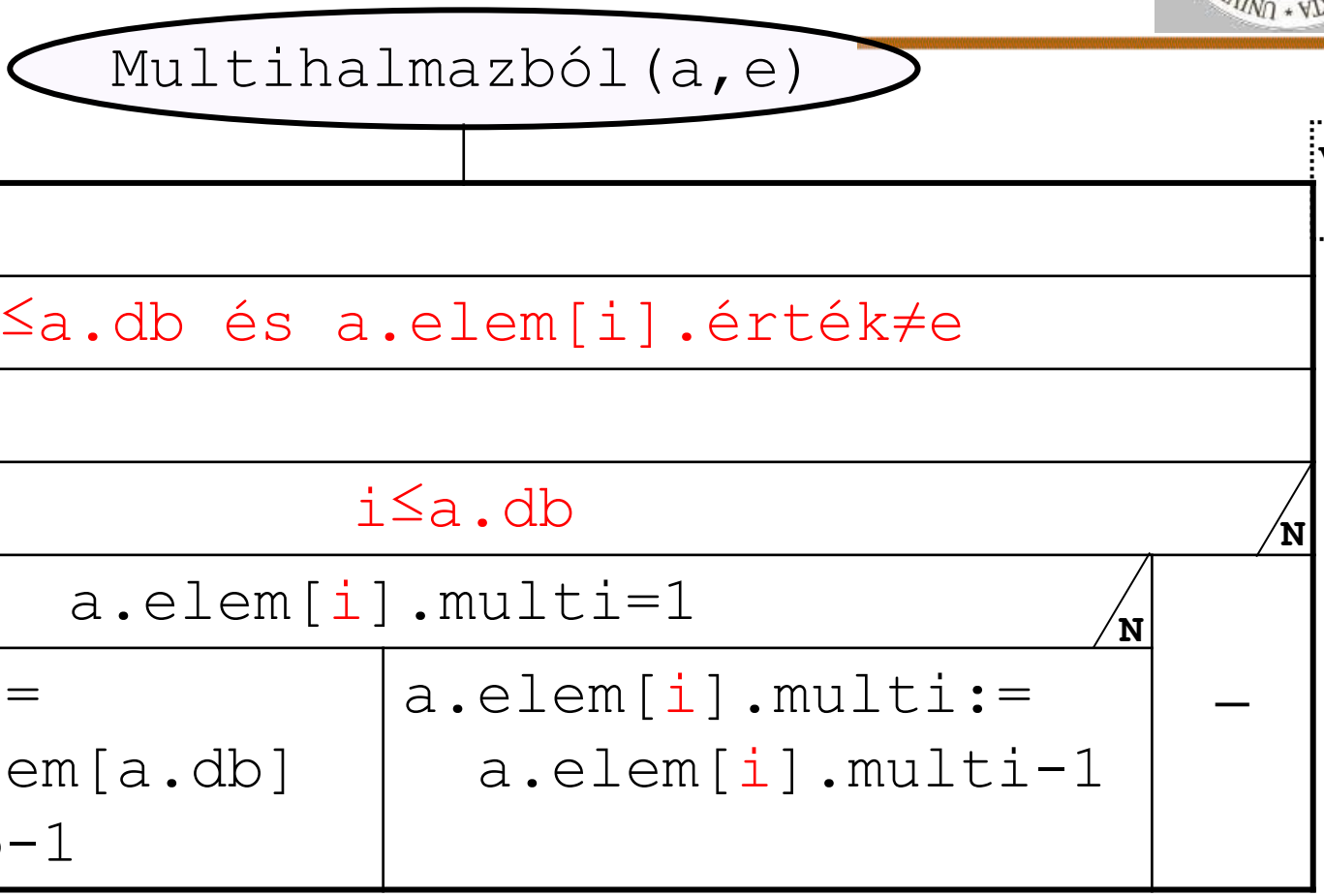
Multihalmaz típus



Műveletigény számítása:

Arányos a multihalmaz elemszámával (keresés tétel).

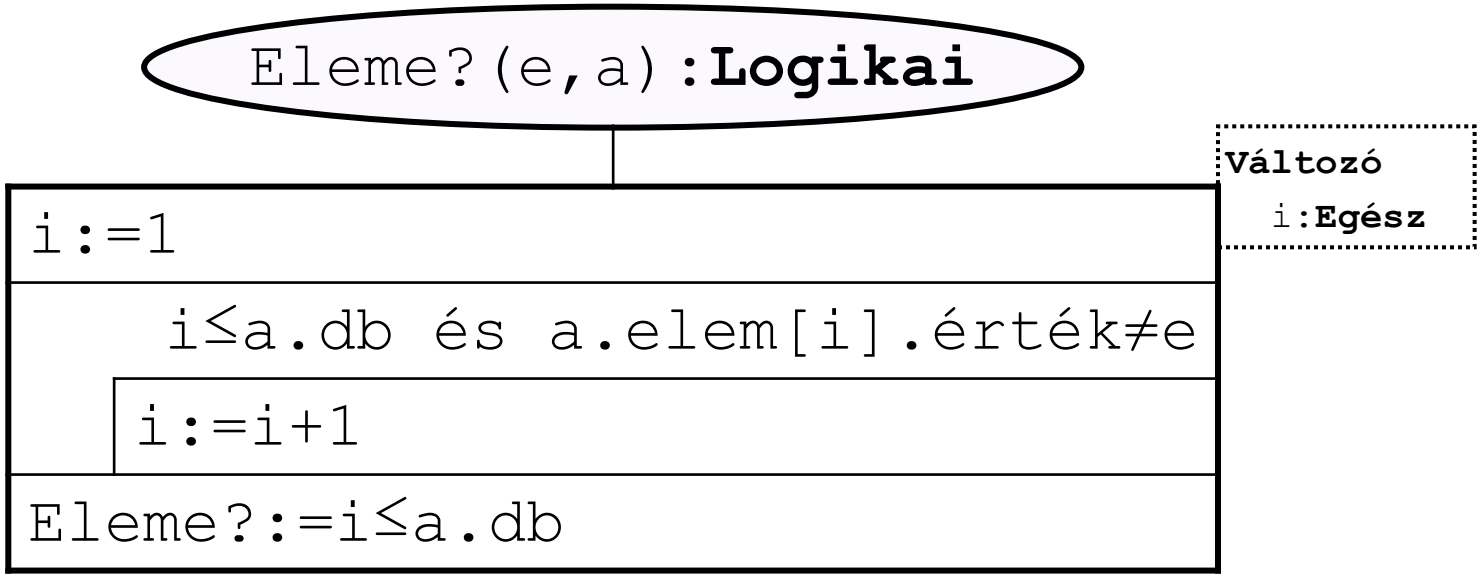
Multihalmaz típus



Műveletigény számítása:

Arányos a multihalmaz elemszámával (keresés tétel).

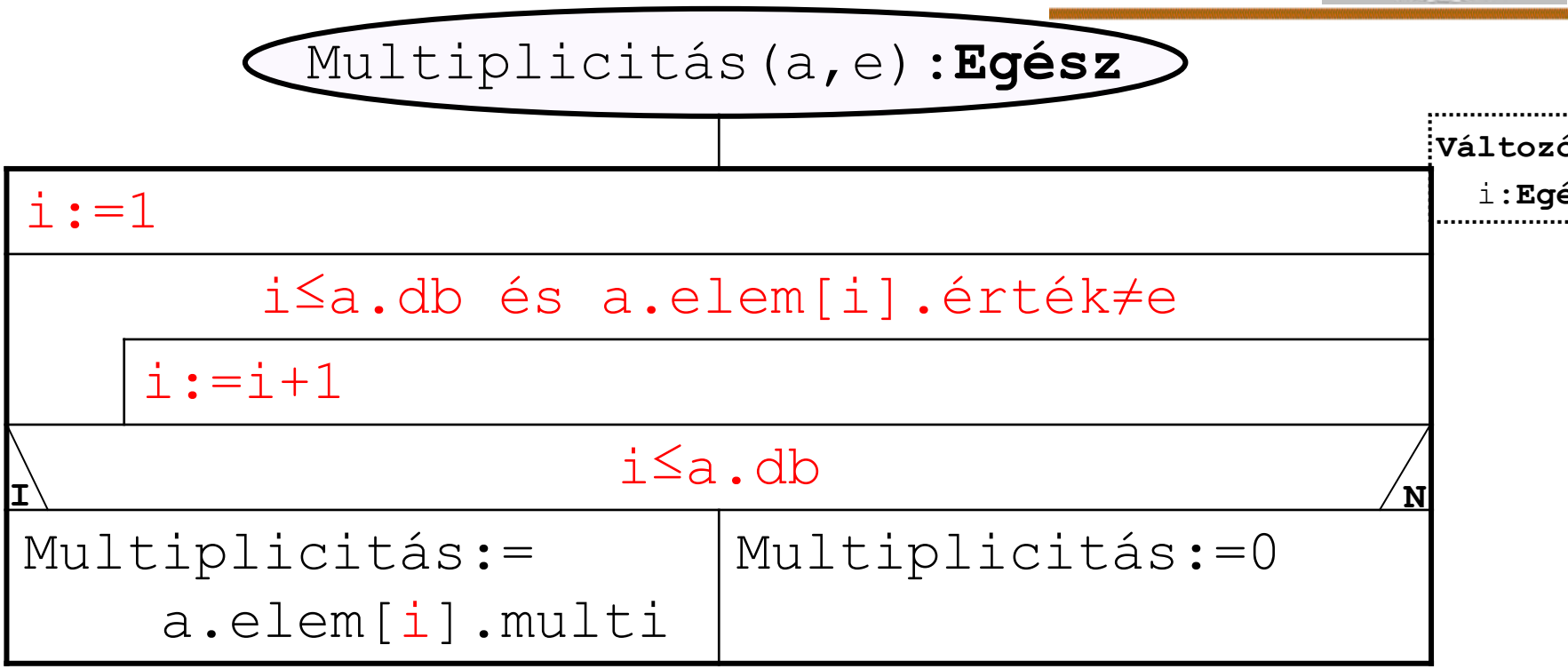
Multihalmaz típus



Műveletigény számítása:

Arányos a multihalmaz elemszámával (eldöntés tétel).

Multihalmaz típus



Műveletigény számítása:

Arányos a multihalmaz elemszámával (keresés tétel).

Multihalmaz típus

Halmazelem típusú: (érték,multi)

Benne?(e,a):**Logikai**

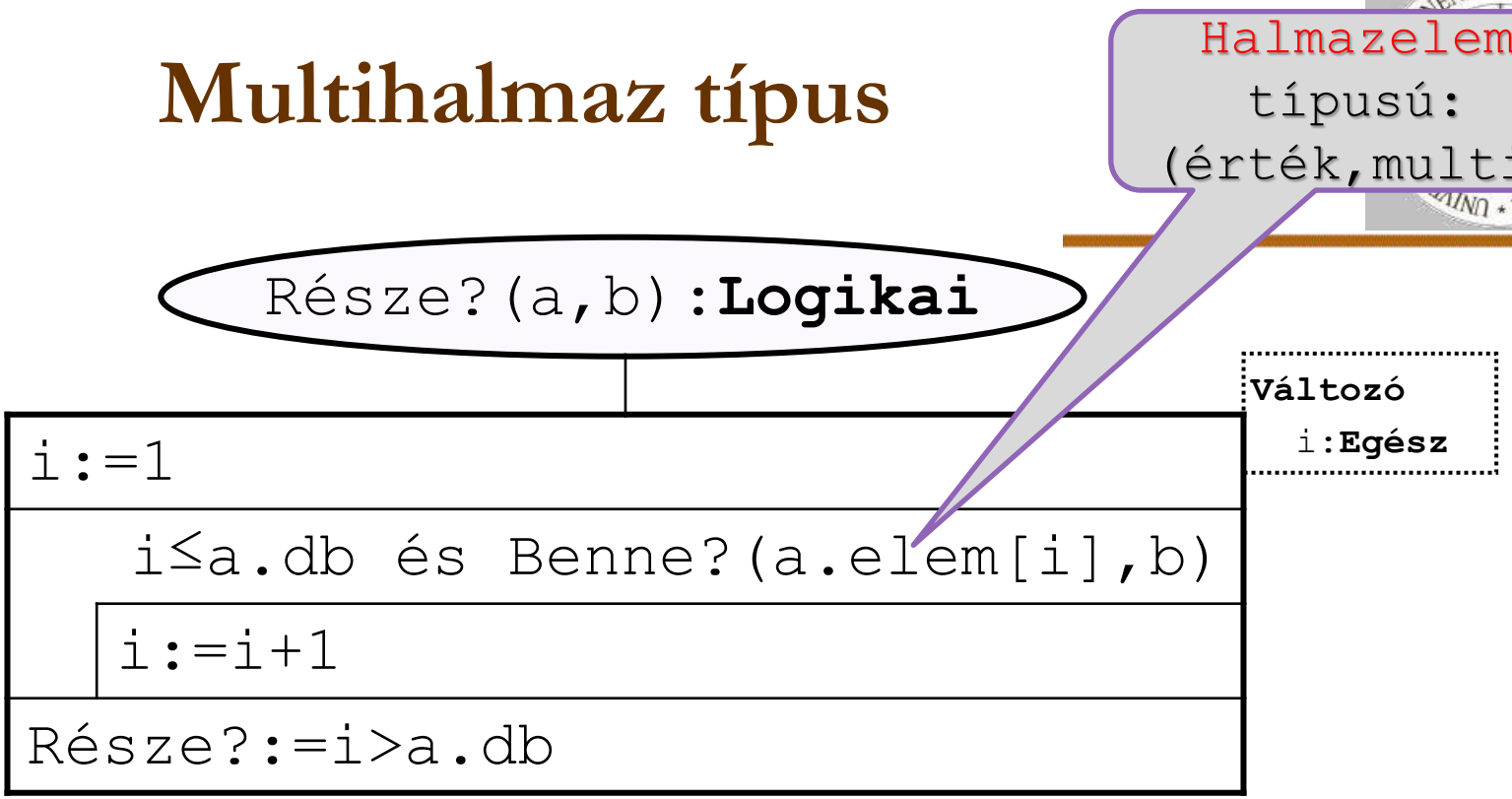
Változó
i:**Egész**

```
i:=1
    i≤a.db és a.elem[i].érték≠e.érték
    i:=i+1
Benne?:=i≤a.db és e.multi≤a.elem[i].multi
```

Műveletigény számítása:

Arányos a multihalmaz elemszámával (keresés tétel).

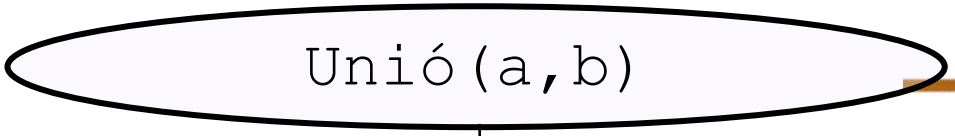
Multihalmaz típus



Műveletigény számítása:

A külső ciklus az ‚a', a `Benne` műveletben levő belső ciklus a ‚b' multihalmaz elemszámaszor fut le, azaz a futási idő a két multihalmaz elemszáma szorzatával arányos.

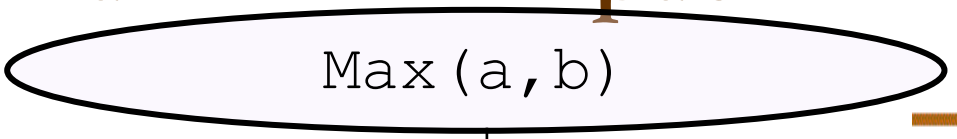
Multihalmaz típus




```
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
```

```
Válto
i,j:
c:Mu
```

Multihalmaz típus




```
c:=a
```

i=1..b.db

```
j:=1
```

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

```
j:=j+1
```

j>a.db

I

```
c.db:=c.db+1
c.elem[c.db]:=
          b.elem[i]
```

N

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

I

```
c.elem[j].multi:=
    b.elem[i].multi
```

N

```
_
```

```
Max:=c
```

Válto
i,j:
c:Mu

Multihalmaz típus


```
c.db:=0
i=1..a.db
    j:=1
    j≤b.db és b.elem[j].érték≠a.elem[i].érték
        j:=j+1
    I  j≤b.db  N
    c.db:=c.db+1; c.elem[c.db]:=a.elem[i]
    I  b.elem[j].multi<a.elem[i].multi  N
    c.elem[c.db].multi:=b.elem[j].multi  —
—
Metszet:=c
```

Multihalmaz típus ábrázolása$_2$

Darabszám vektor:

Típus

```
Multihalmaz(Elemtípus)=
        Tömb[Min'Elemtípus..Max'Elemtípus:Egész]
```

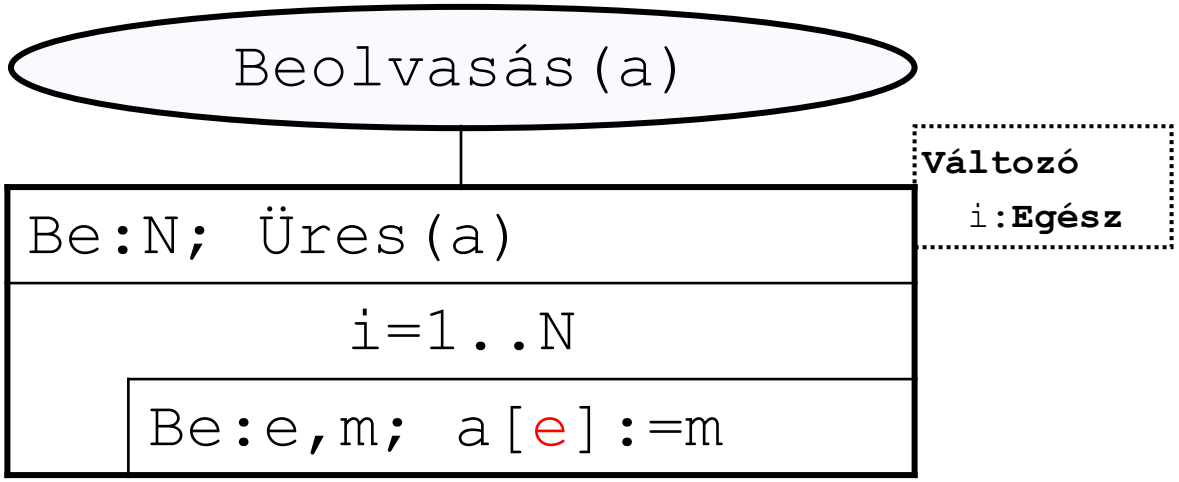
Vegyünk fel egy annyi elemből álló sorozatot, amennyi a multihalmaz lehetséges elem fajtáinak száma!

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

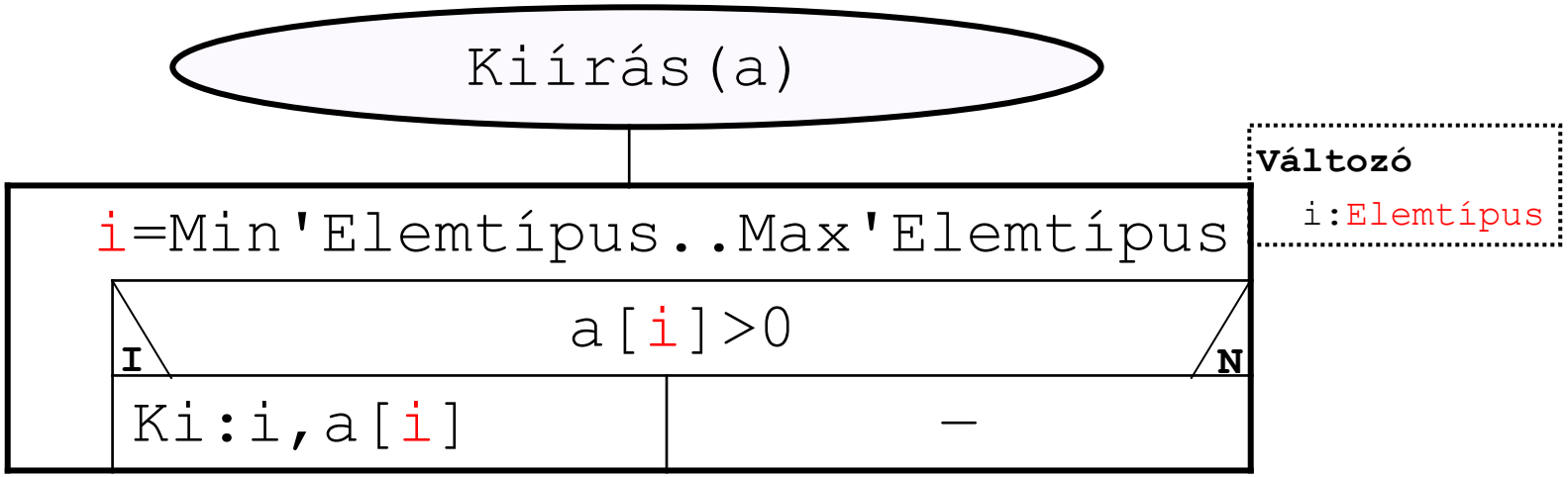


Műveletigény számítása:

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.

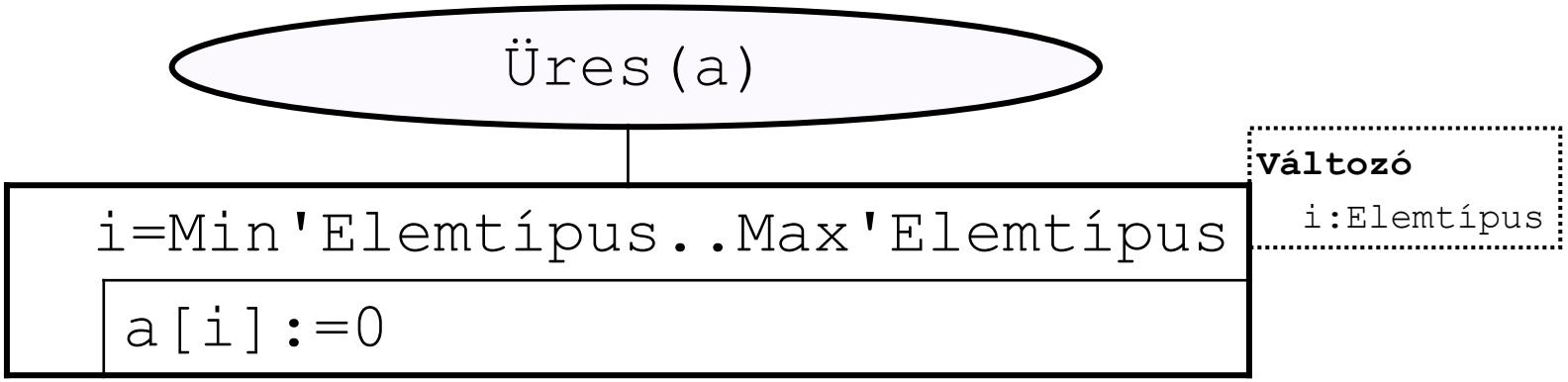
Multihalmaz típus



Műveletigény számítása:

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

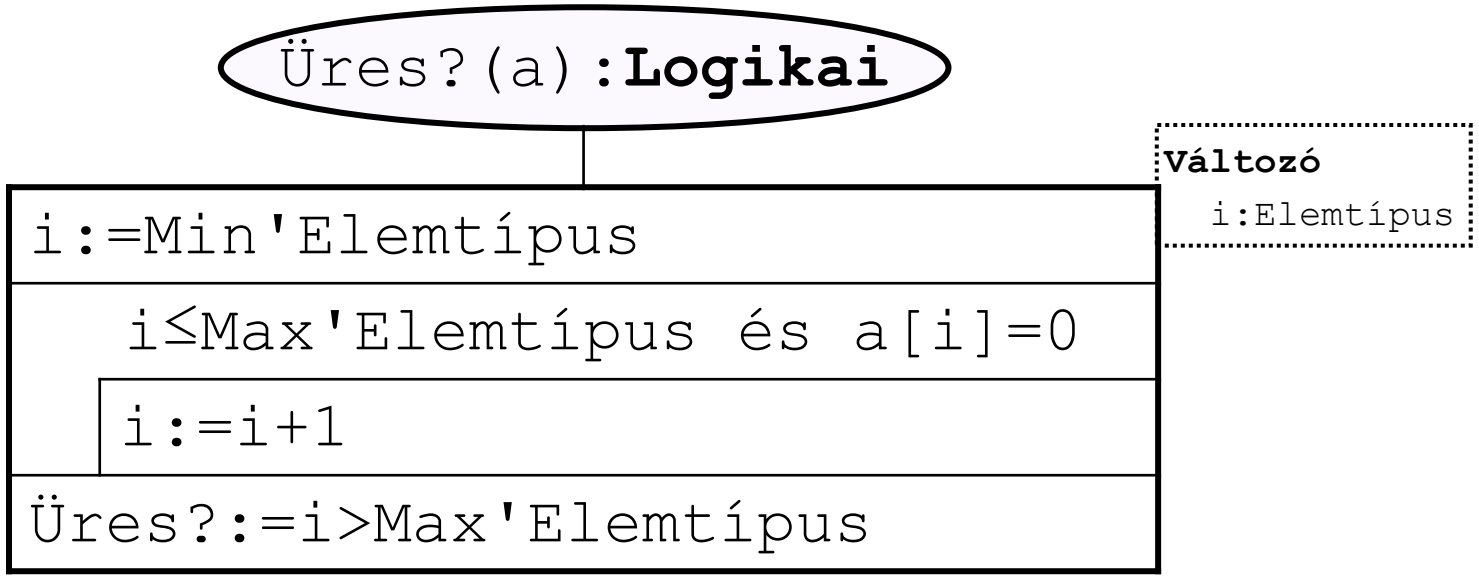
Multihalmaz típus



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

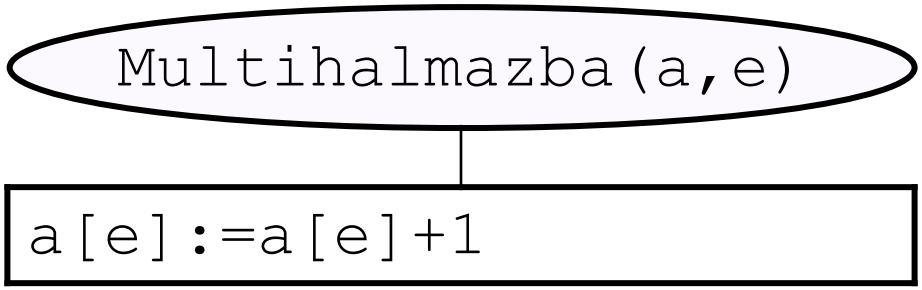


Műveletigény számítása:

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

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

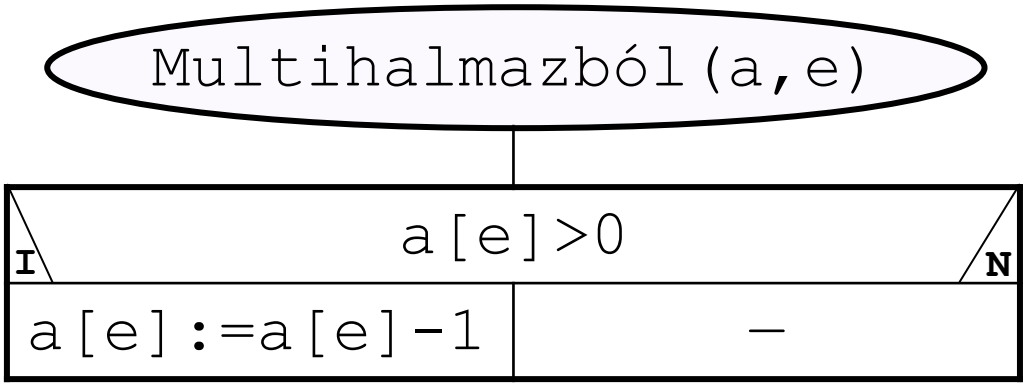
Multihalmaz típus



Műveletigény számítása:

Nem függ a multihalmaz elemszámától.

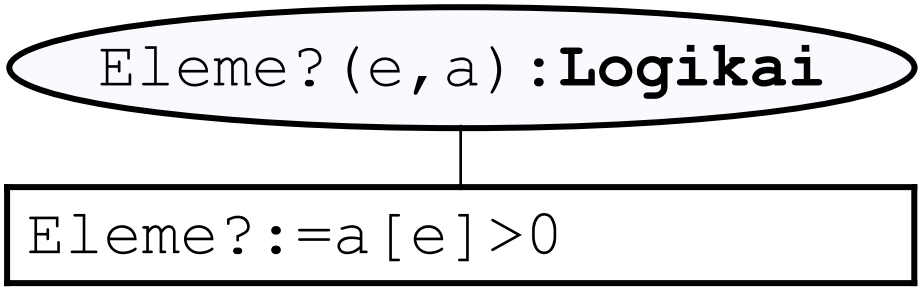
Multihalmaz típus



Műveletigény számítása:

Nem függ a multihalmaz elemszámától.

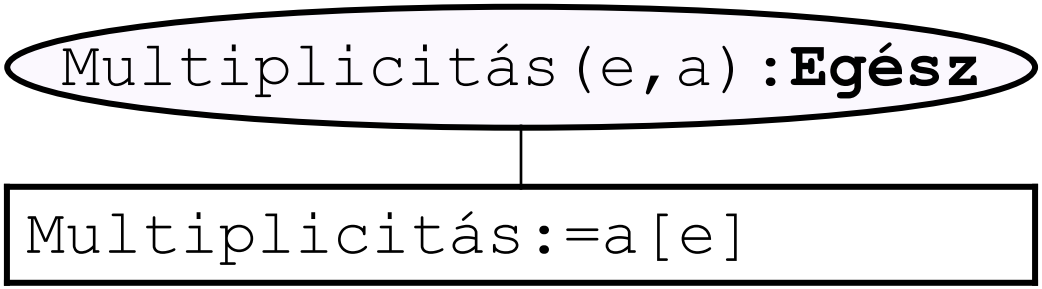
Multihalmaz típus



Műveletigény számítása:

Nem függ a multihalmaz elemszámától.

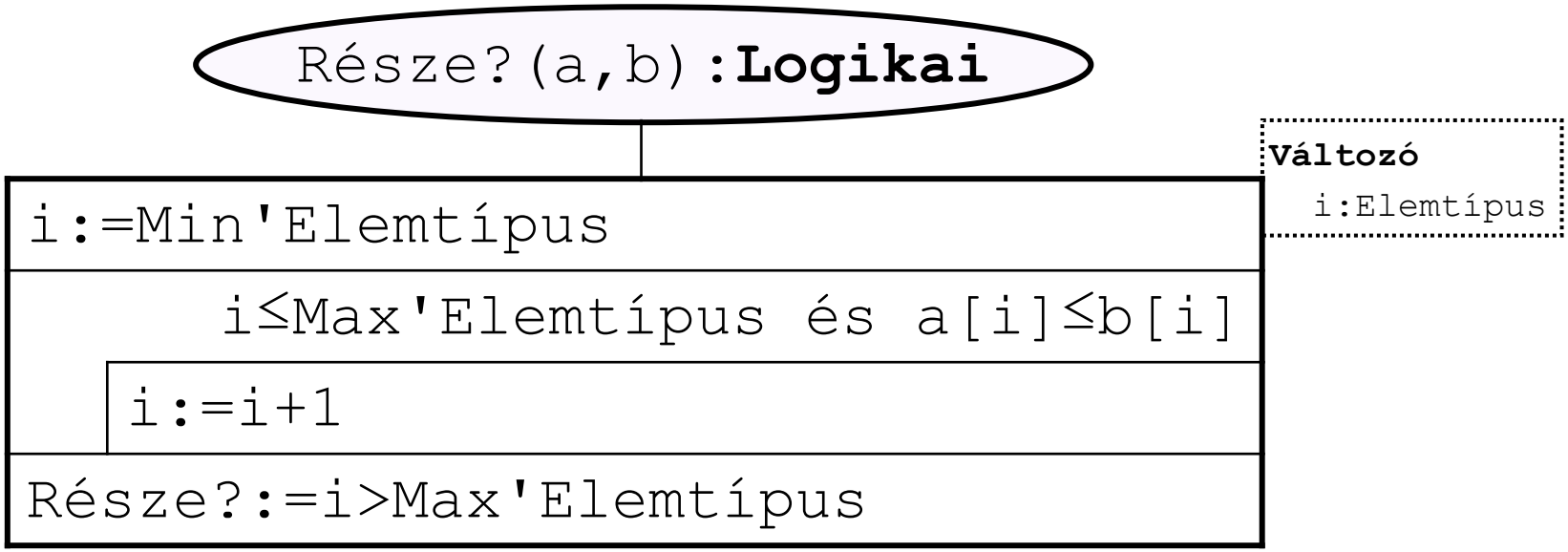
Multihalmaz típus



Műveletigény számítása:

Nem függ a multihalmaz elemszámától.

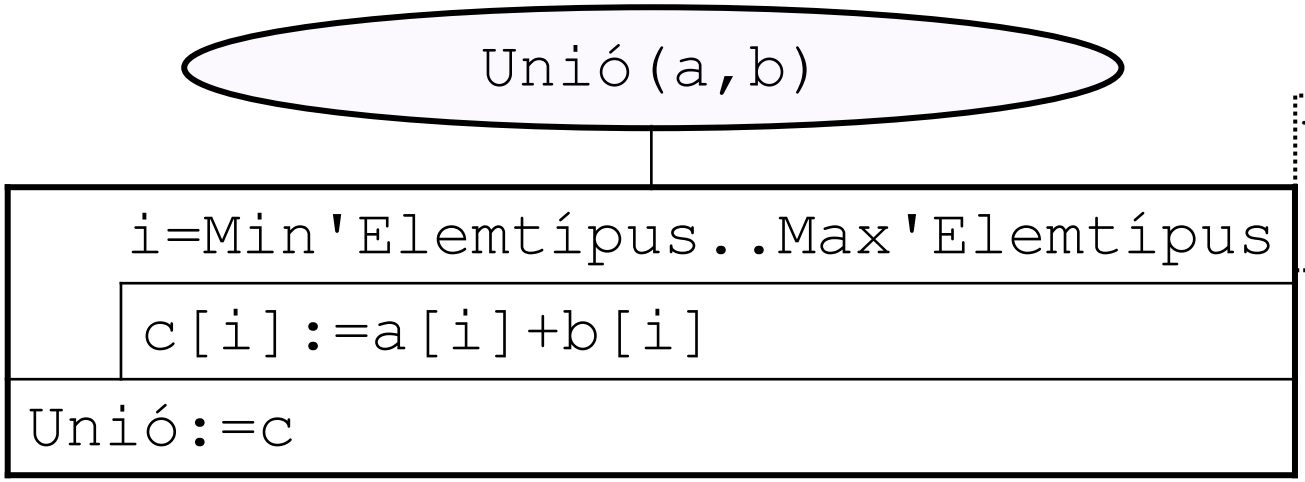
Multihalmaz típus



Műveletigény számítása:

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

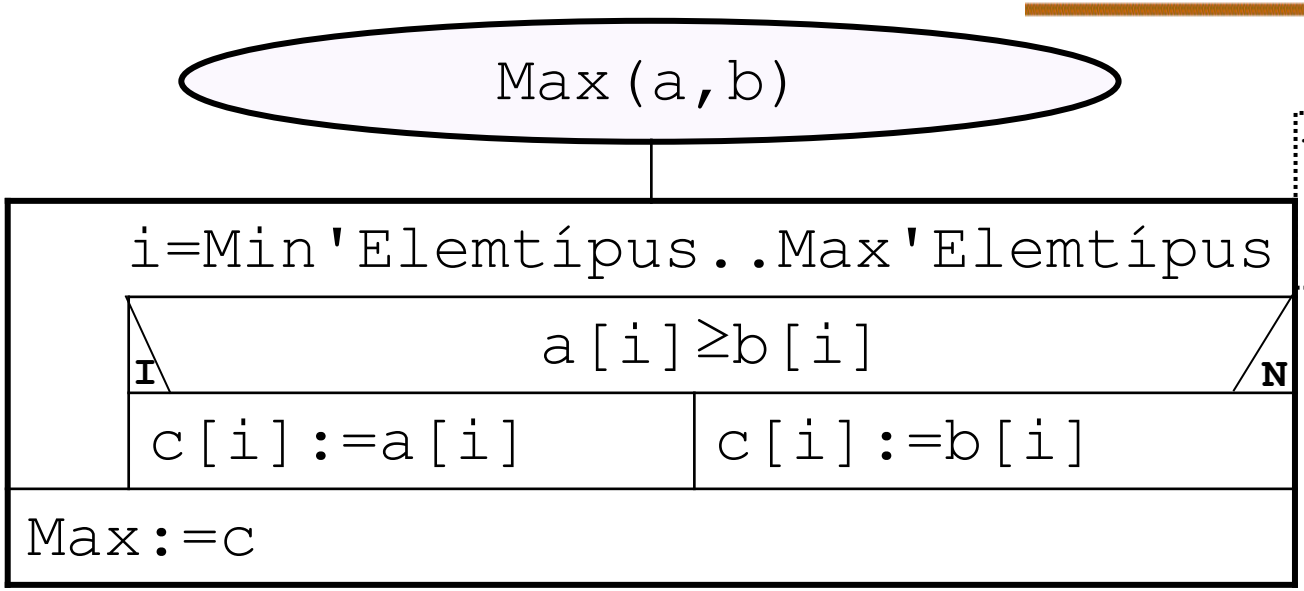
Multihalmaz típus



Műveletigény számítása:

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

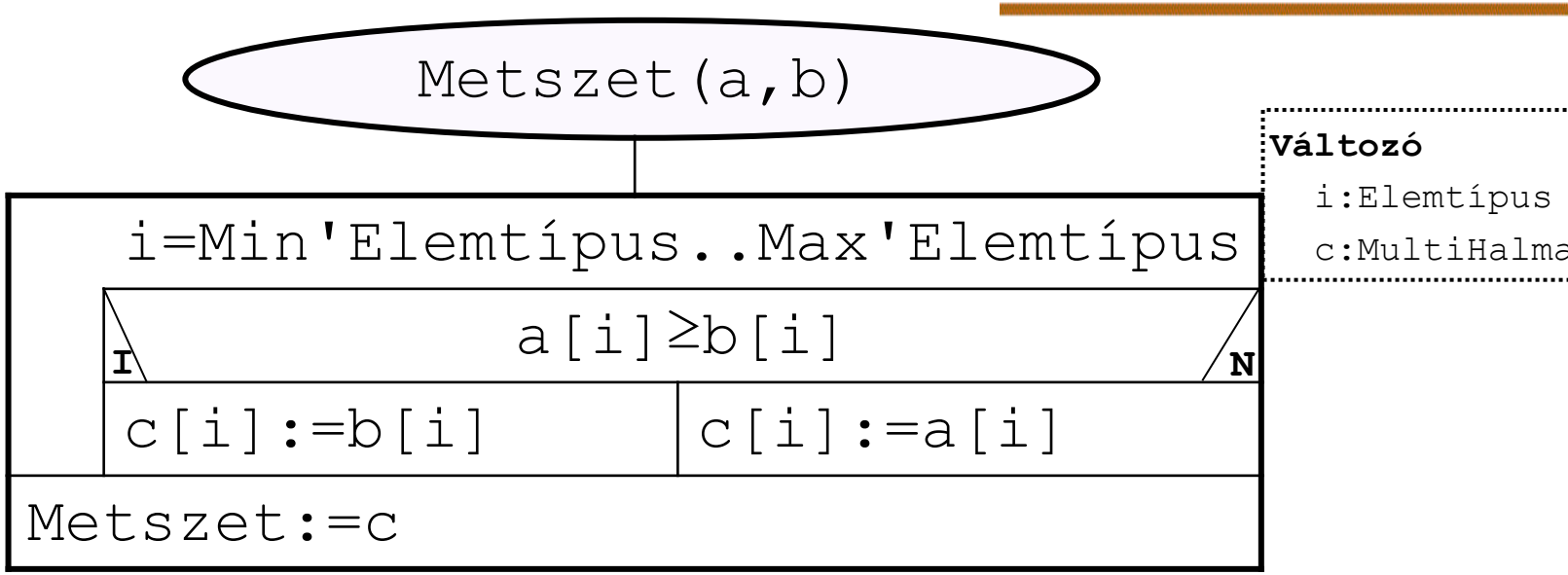
Multihalmaz típus



Műveletigény számítása:

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

Multihalmaz típus



Műveletigény számítása:

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

Tartalom

- Programozási tételek általánosítása
  - Összegzés / Feltételes összegzés
  - Megszámolás
  - Maximum-kiválasztás / Feltételes maximum-keresés
  - Kiválasztás
  - Keresés / eldöntés
- Halmaz általánosítása: Multihalmaz
  - Multihalmaz típus elemek felsorolásával
  - Multihalmaz típus darabszám vektorral