

The background of the slide is a black and white aerial photograph of Budapest, Hungary. It shows the city's layout along the Danube River, with the Buda Castle and its surrounding hills on the left and the Pest side on the right. A semi-transparent white rectangle is centered over the image, containing the title text.

Programozás

9. előadás

Tartalom

- Programozási tételek általánosítása
 - Összegzés
 - Megszámolás
 - Maximum-kiválasztás
 - Feltételes Maximum-keresés
 - Kiválasztás
 - Keresés
- Multihalmaz típus
 - Multihalmaz típus elemek felsorolásával
 - Multihalmaz típus darabszám vektorral



Programozási tételek általánosítása₁



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

Módszer: függvények célszerű paraméterezése és behelyettesítése.

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

Például annak eldöntése, hogy egy tömb monoton növekvő-e, a $[2,N]$ 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=valmilyen
számhalmaz



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

helyett

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

Kimenet: $S \in H$

Előfeltétel: –

Utófeltétel: $S = \sum_{i=1}^N X_i$

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

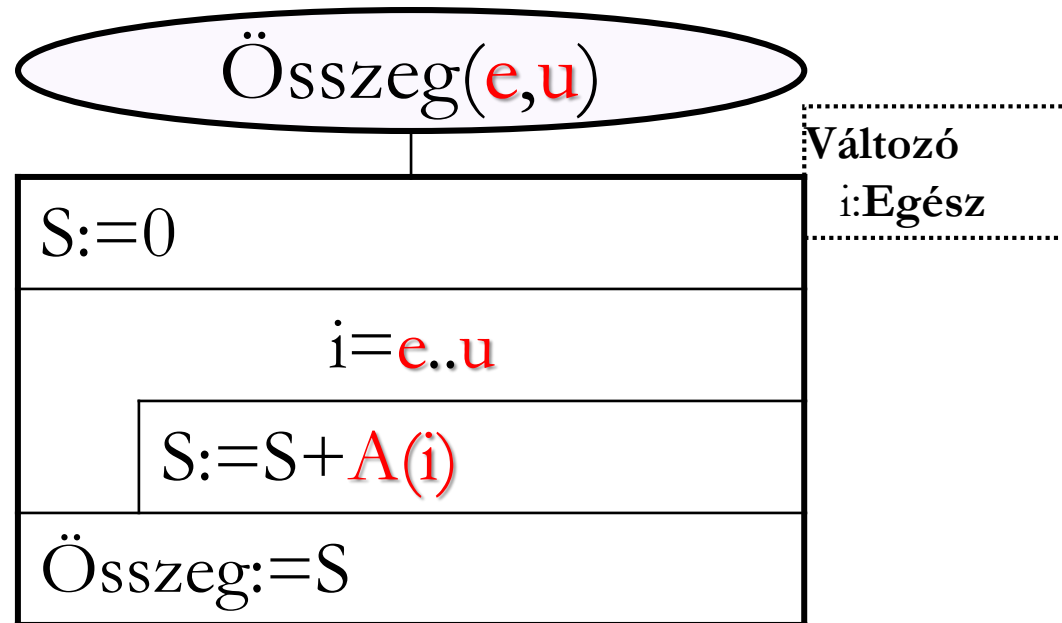
Az alapváltozatban:

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



Összegzés – intervallumon

Algoritmus:



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



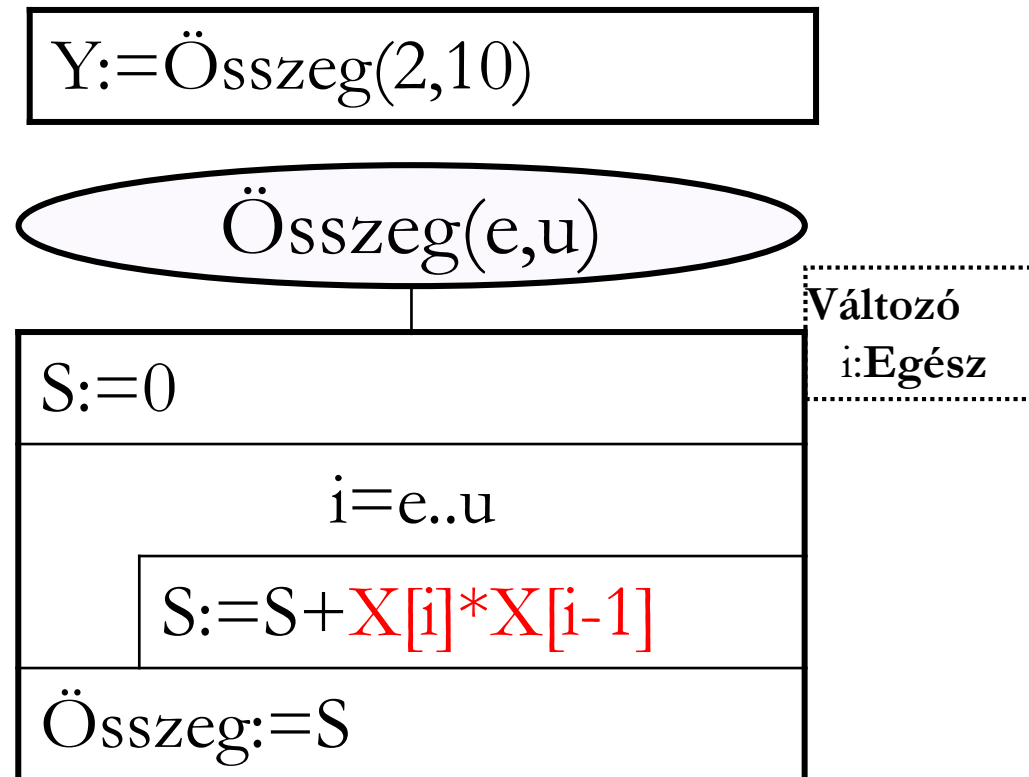
Összegzés – intervallumon

Példa₁:

Utófeltétel:

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

A behelyettesített X tömb globális változó.



Összegzés – intervallumon

Példa₂:

Utófeltétel:

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

Feltéve, hogy az X tömb
-11-től indexelhető.

Y:=Összeg(-10,10)

Összeg(e,u)

S:=0

i=e..u

S:=S+abs(X[i]-X[i-1])

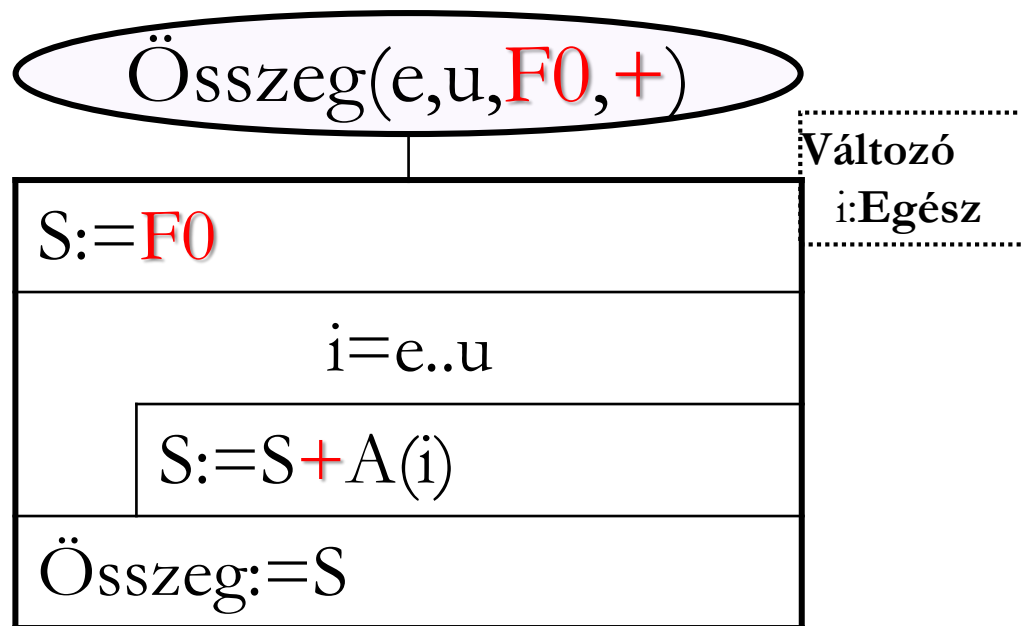
Összeg:=S

Változó
i:Egész



Sorozatszámítás – intervallumon

Algoritmus (általánosan):



Új paraméterek:

- a nullelem (**F0**) és a
- kapcsolódó 2-változós műveleti operátor (**+**).



Feltételes összegzés – intervallumon

Bemenet: $N \in \mathbb{N}$, helyett $e, u \in \mathbb{Z}$
 $X_{1..N} \in H^N \rightarrow A: \mathbb{Z} \rightarrow H$
 $T: H \rightarrow L$ $T: \mathbb{Z} \rightarrow L$

Kimenet: $S \in H$

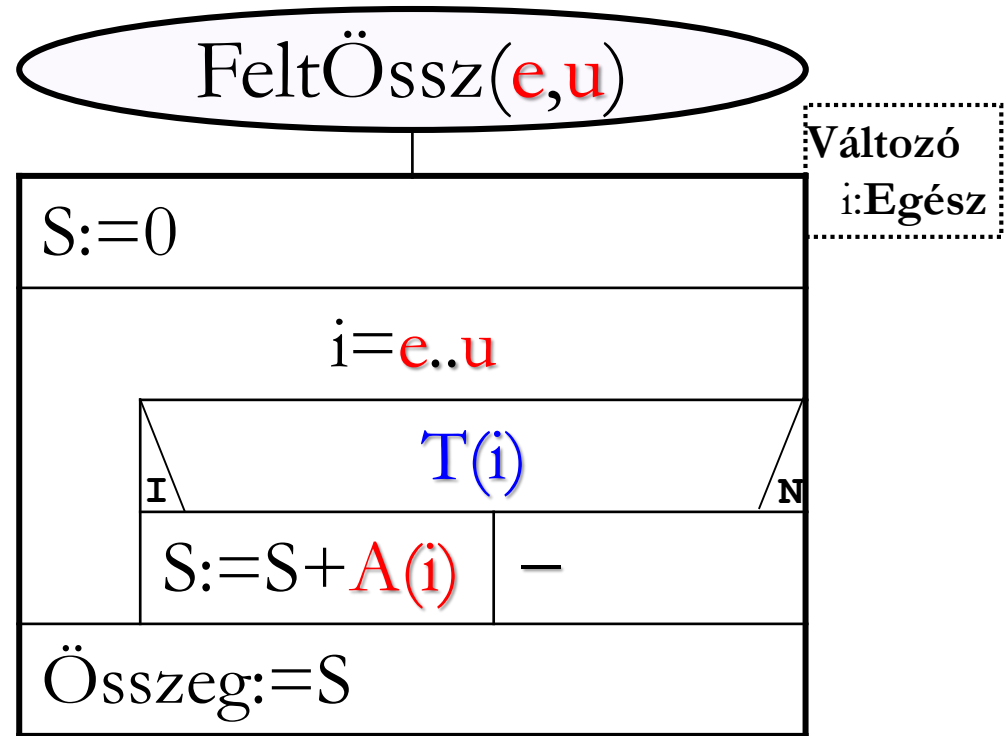
Előfeltétel: –

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



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

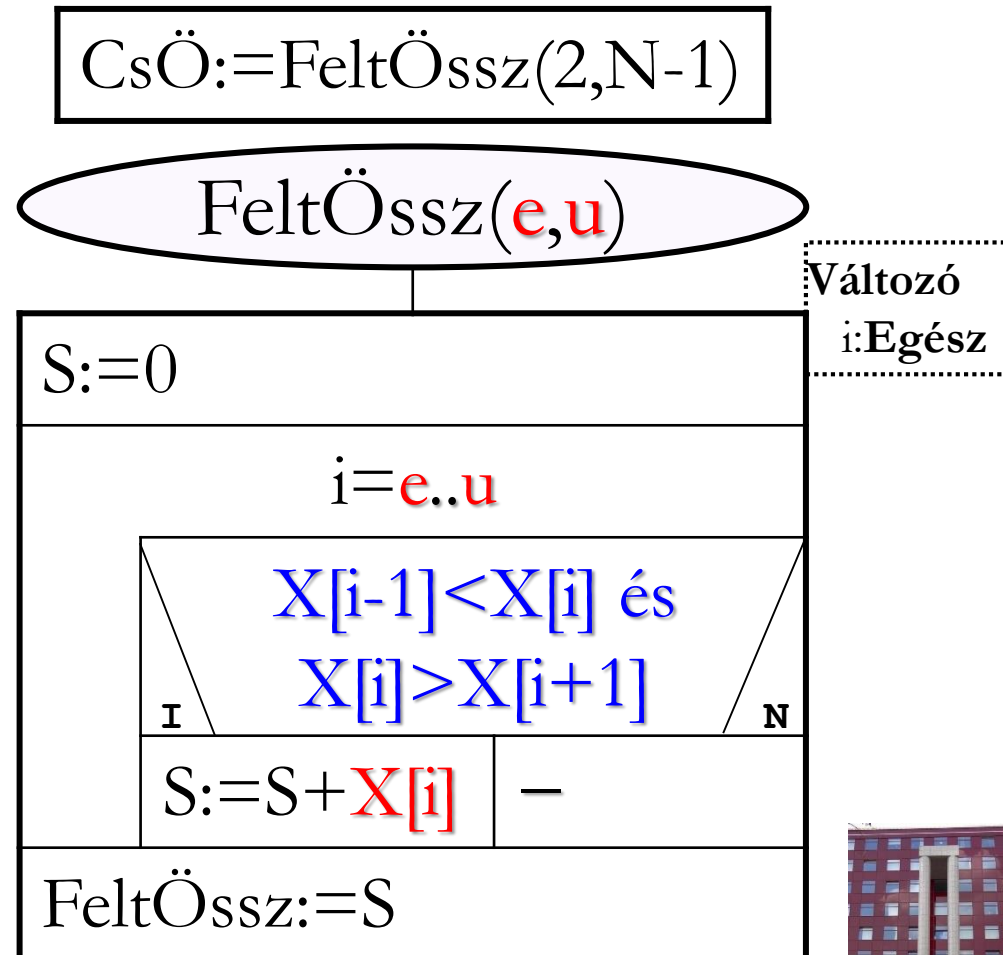
Példa:

Feladat: csúcsok összege...

Utófeltétel:

$$C_s \ddot{O} = \sum_{i=2}^{N-1} X_i$$

$X_{i-1} < X_i$ és $X_i > X_{i+1}$



Megszámolás – intervallumon

Specifikáció:

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

helyett

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

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

Előfeltétel: –

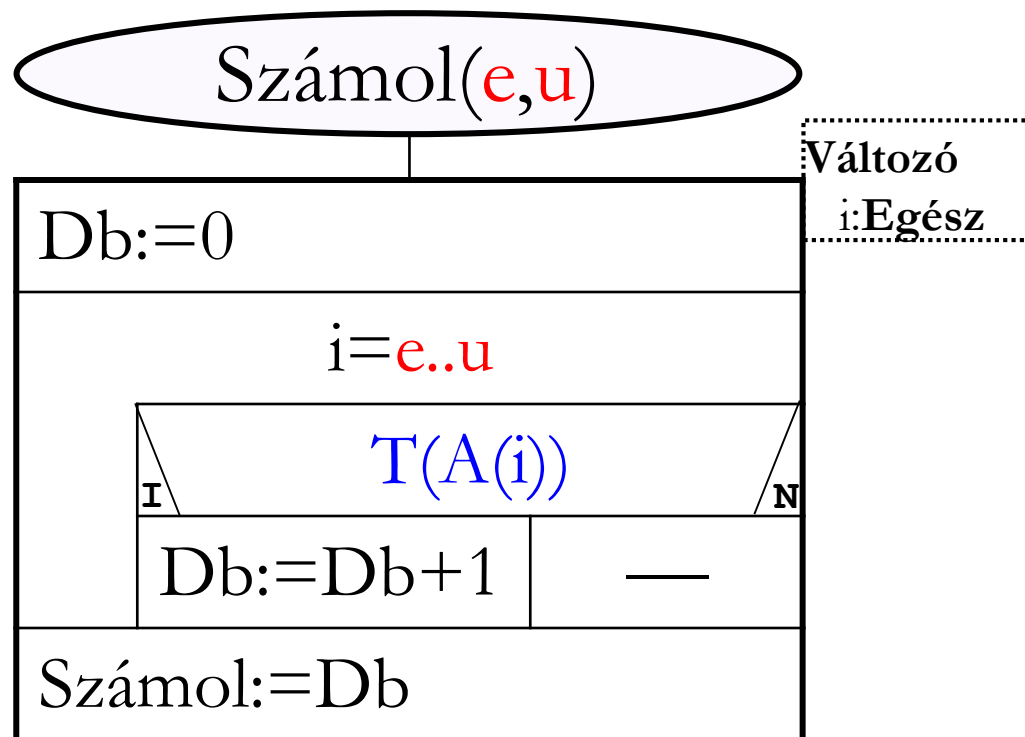
Utófeltétel: $D_b = \sum_{\substack{i=1 \\ T(X_i)}}^N 1$

$D_b = \sum_{\substack{i=e \\ T(A(i))}}^u 1$



Megszámolás – intervallumon

Algoritmus:



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



Megszámolás – intervallumon

Specifikáció₂:

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

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

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

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

Előfeltétel: –

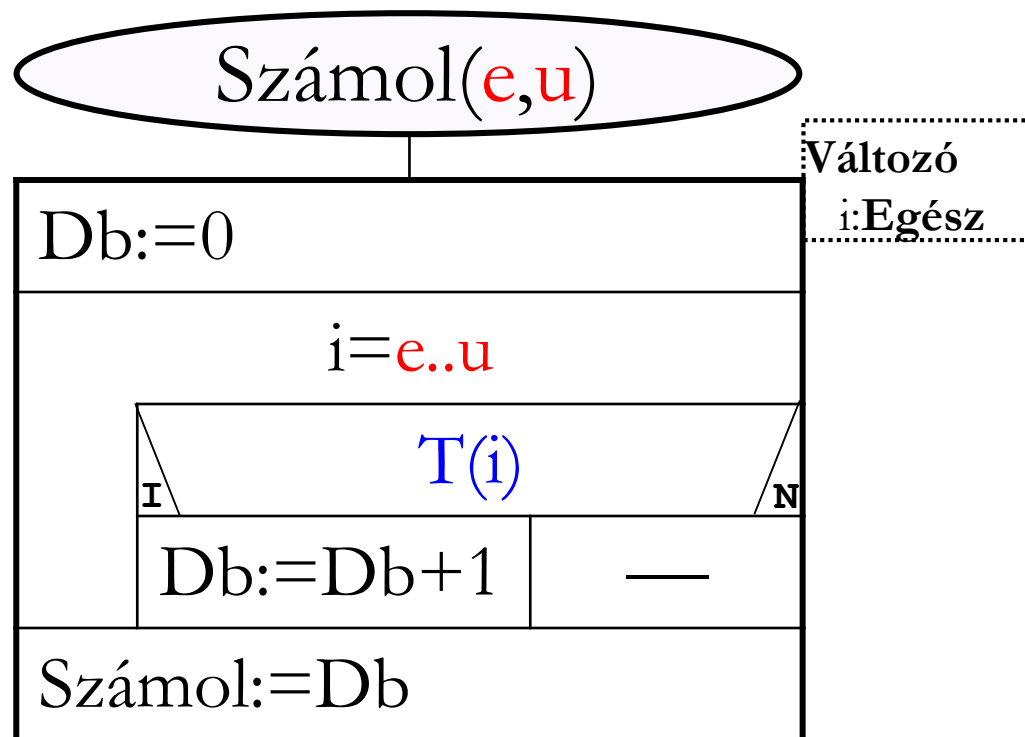
Utófeltétel: $D_b = \sum_{\substack{i=1 \\ T(X_i)}}^N 1$

$D_b = \sum_{\substack{i=e \\ T(i)}}^u 1$



Megszámolás – intervallumon

Algoritmus₂:



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



Megszámolás – intervallumon

Példa:

Utófeltétel:

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

$X_i > X_{i+1}$

DbCs:=Számol(1,N-1)

Számol(e,u)

Változó
i:Egész

Db:=0

i=e..u

$X[i] > X[i+1]$

Db:=Db+1

—

Számol:=Db



Maximum-kiválasztás – intervallumon

Specifikáció:

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

helyett

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

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

Előfeltétel: $N > 0$

$e \leq u$

Utófeltétel: $1 \leq \text{Max} \leq N$ és

$e \leq \text{Max} \leq u$ és

$\text{MaxÉrt} = X_{\text{Max}}$

$\text{MaxÉrt} = A(\text{Max})$

$\forall i (1 \leq i \leq N)$:

$\forall i (e \leq i \leq u)$:

$\text{MaxÉrt} \geq X_i$

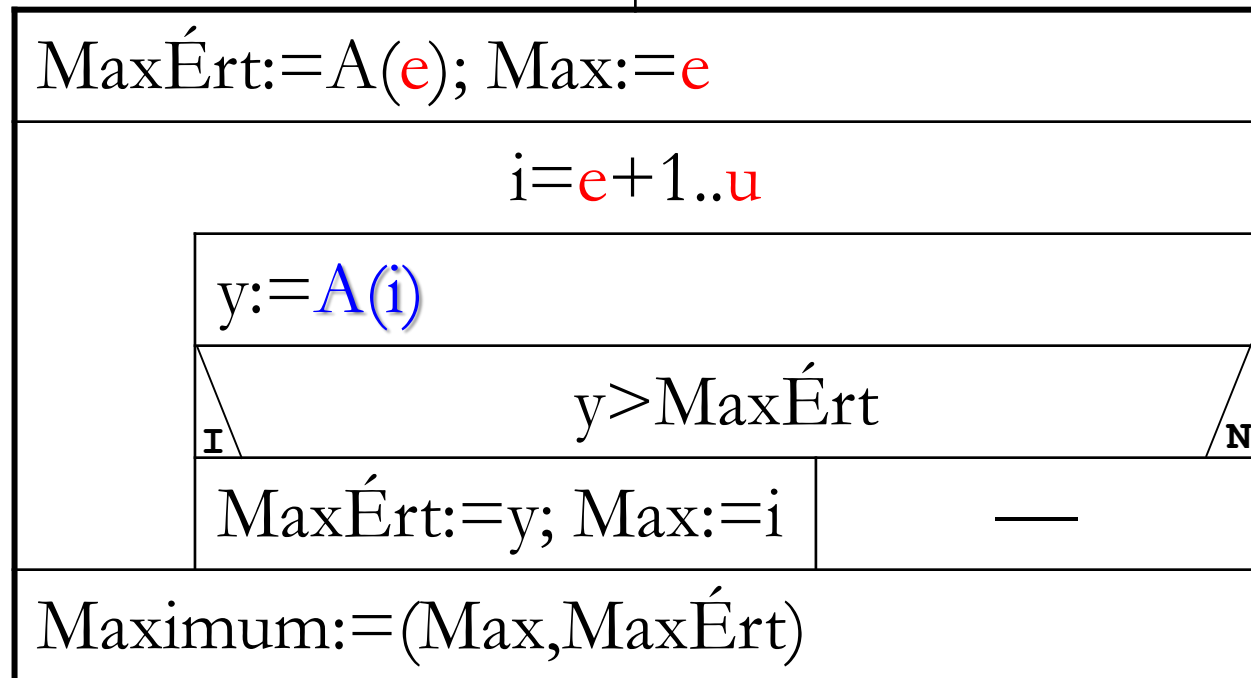
$\text{MaxÉrt} \geq A(i)$



Maximum-kiválasztás – intervallumon

Algoritmus:

Maximum(**e**,**u**)



Változó
i:Egész
y:TH

A(**i**) helyébe
az adott képlet
behelyettesítendő (egyetlen helyen).



Maximum-kiválasztás

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

Ilyenkor az utolsó értékadás

$\text{Maximum} := (\text{Max}, \text{MaxÉrt})$

helyett a

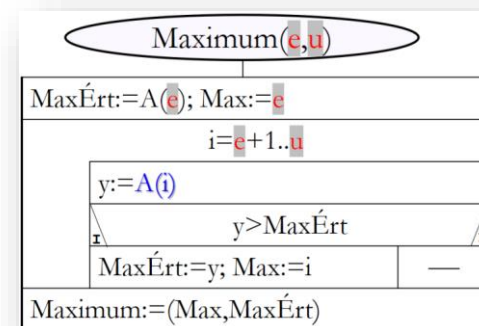
$\text{Maximum} := \text{Max}$

vagy

$\text{Maximum} := \text{MaxÉrt}$

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(e,u,MaxÉrt)

MaxÉrt:=A(e); Max:=e

i=e+1..u

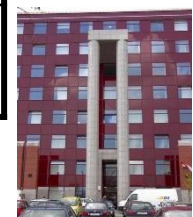
y:=A(i)

y>MaxÉrt

MaxÉrt:=y; Max:=i

Maximum:=Max

Változó
i:Egész
y:TH

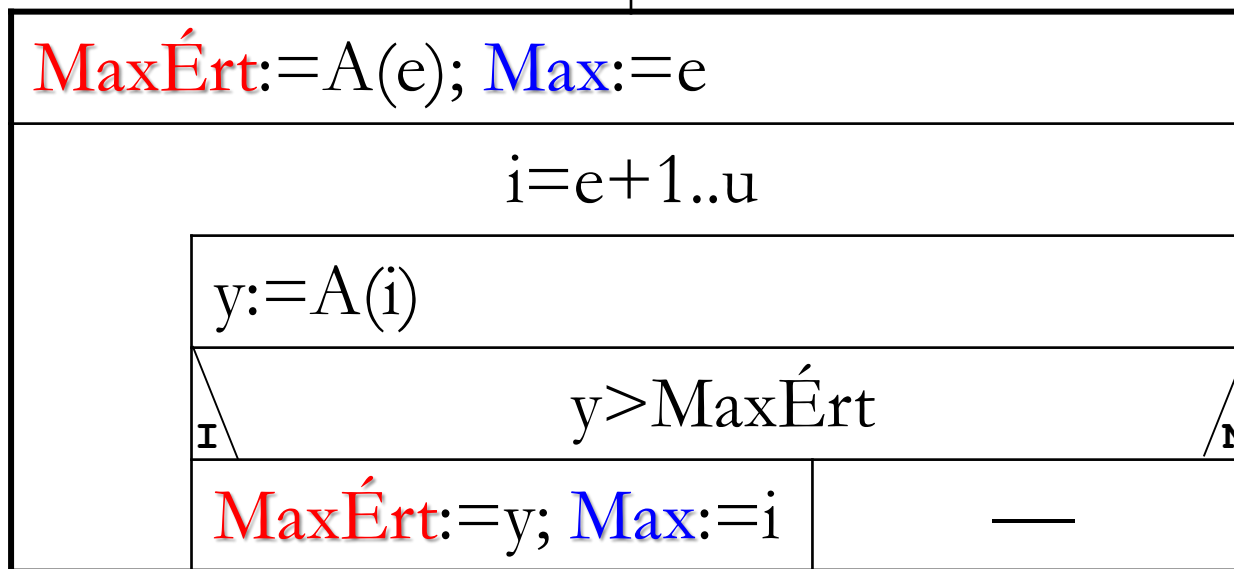


Maximum-kiválasztás – intervallumon

Algoritmus: (függvény két kimenő paraméterrel)

Maximum($e, u, \underline{Max}, \underline{MaxÉrt}$)

Változó
i: Egész
y: TH

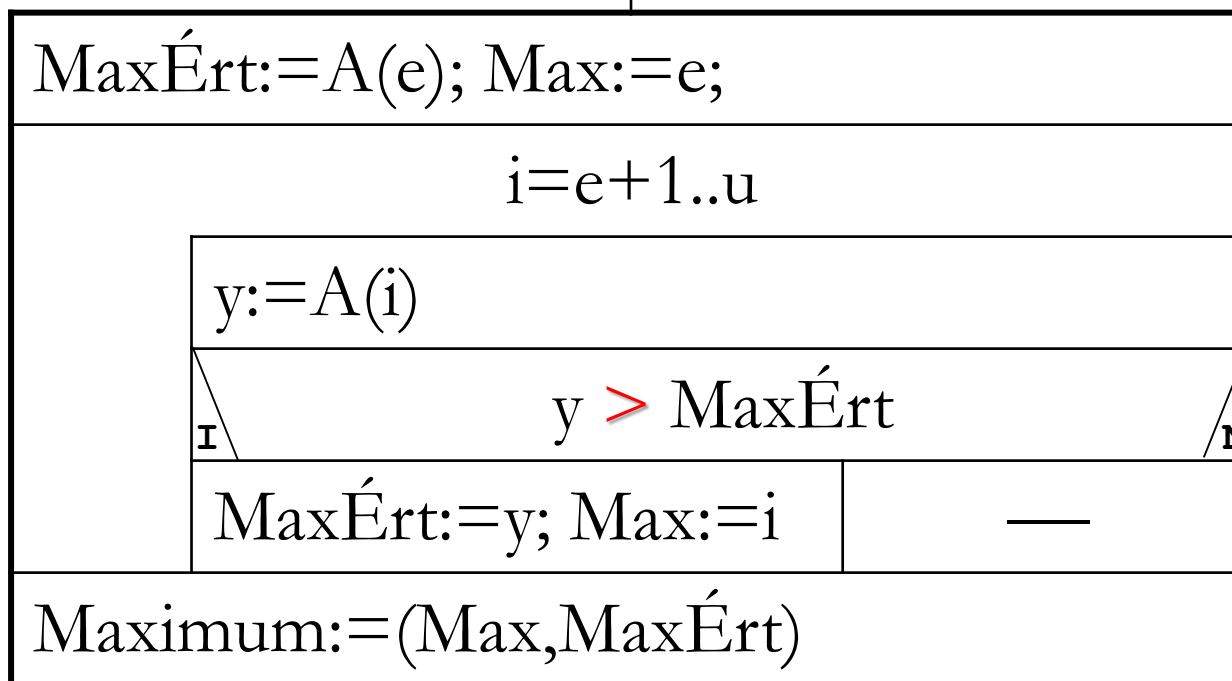


Maximum-kiválasztás – intervallumon

Algoritmus (általános relációval):

Maximum($e, u, >$)

Változó
i: Egész
y: TH



Maximum-kiválasztás – intervallumon

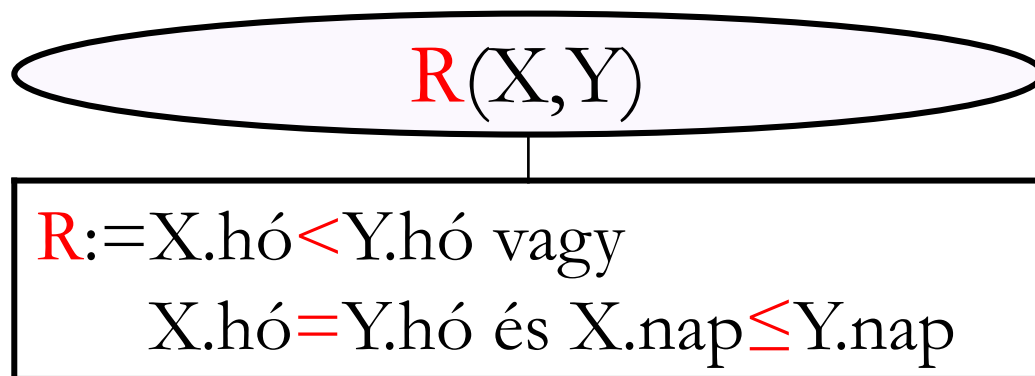


Példa:

Első születésnapos

$$(S, \text{Dátum}) := \text{Maximum}(1, N, \mathbf{R})$$

A Maximum függvényben az $A(i)$ törzse egyszerűen $Sz[i]$, azaz az i . ember születésnapja.



Feltételes maximumkeresés – intervallumon



Specifikáció:

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

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

$T: H \rightarrow L$

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

Előfeltétel: $e \leq u$

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



Feltételes maximumkeresés – intervallumon



Algoritmus:

FeltMax(**e**,**u**,**T**)

MaxÉrt:=-∞

i=**e..u**

I	T (A(i)) és A(i)>MaxÉrt	N
MaxÉrt:=A(i)	—	
Max:=i		

Van:=MaxÉrt≠-∞

FeltMax:=(Van,Max,MaxÉrt)

Változó
i:Egész



Feltételes maximumkeresés – intervallumon



Specifikáció₂:

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

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

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

Példa: lokális
csúcsok minimuma

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

Előfeltétel: $e \leq u$

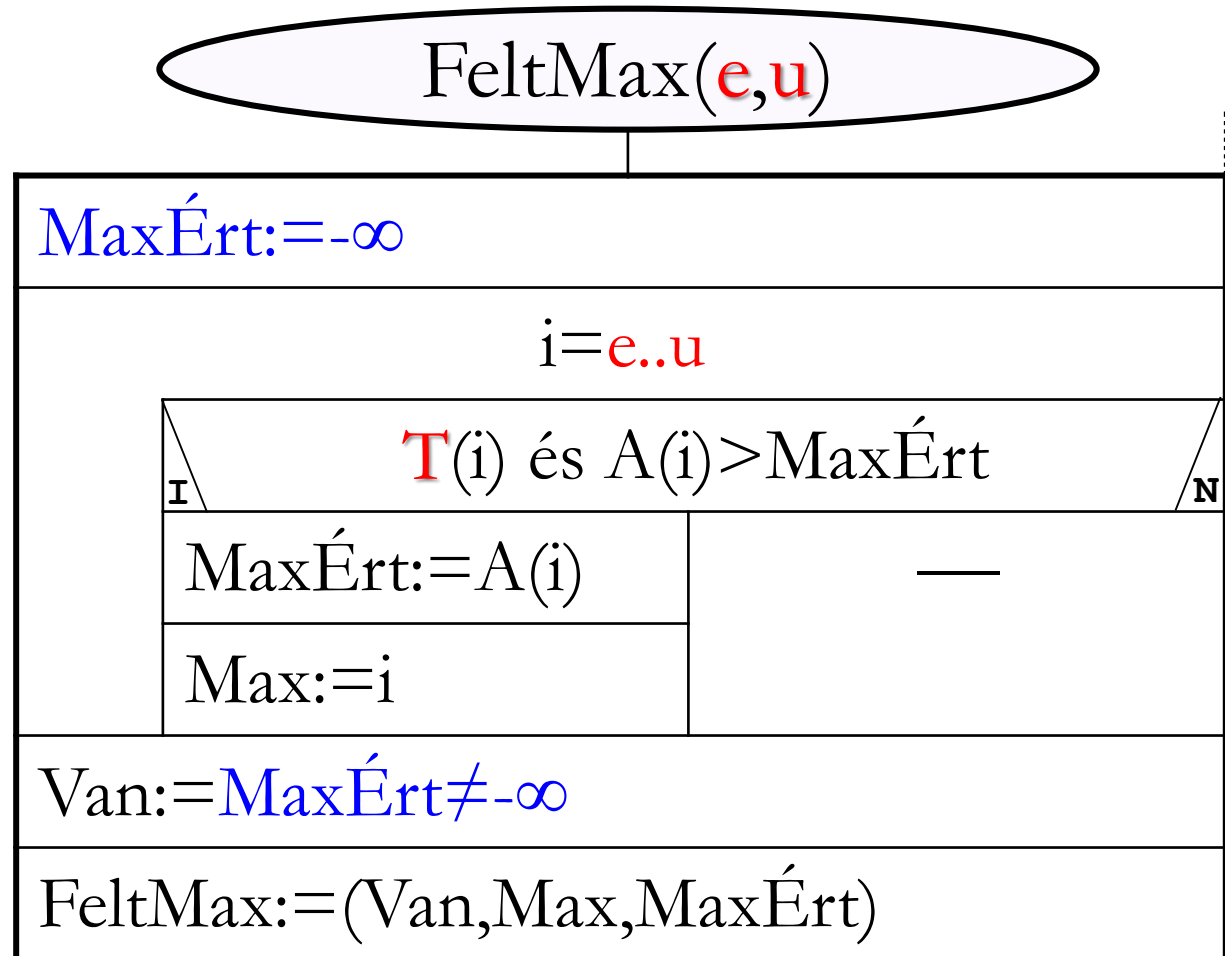
Utófeltétel: $Van = \exists i (e \leq i \leq u) : T(i)$ és
 $Van \rightarrow (e \leq Max \leq u \text{ és } MaxÉrt = A(Max) \text{ és}$
 $T(Max) \text{ és}$
 $\forall i (e \leq i \leq u) : T(i) \rightarrow MaxÉrt \geq A(i))$



Feltételes maximumkeresés – intervallumon



Algoritmus₂:



Változó
i:Egész



Feltételes maximumkeresés

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

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

$T: H \rightarrow L$

Kimenet: $Ind \in \mathbb{Z}, Ért \in H$

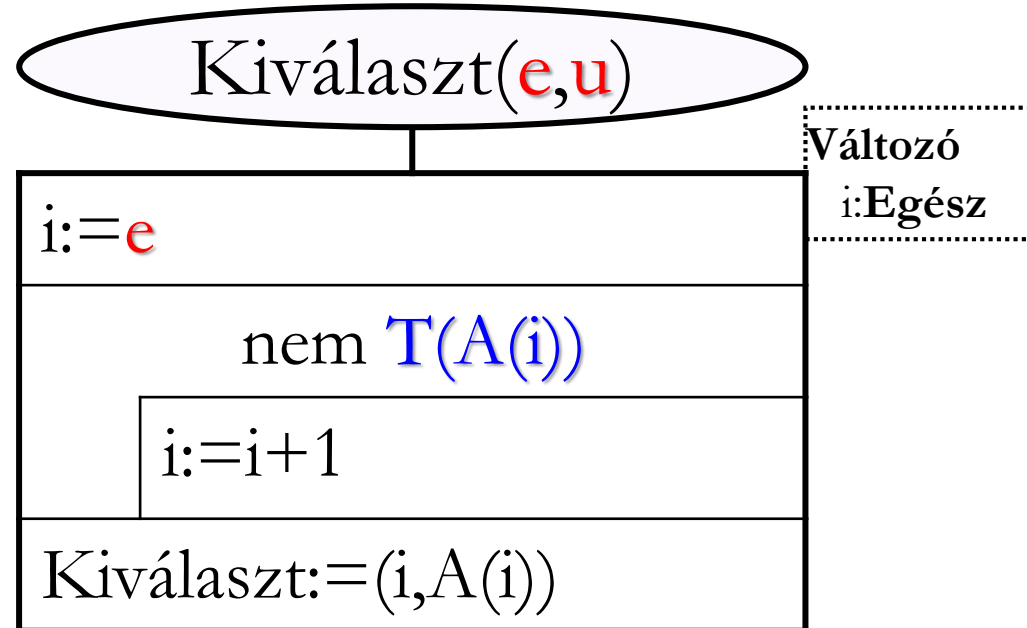
Előfeltétel: $e \leq u$ és $\exists i (e \leq i \leq u): T(A(i))$

Utófeltétel: $e \leq Ind \leq u$ és $T(A(Ind))$ és $Ért = A(Ind)$



Kiválasztás – intervallumon

Algoritmus:



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

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



Keresés – intervallumon

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

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

$T: H \rightarrow L$

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

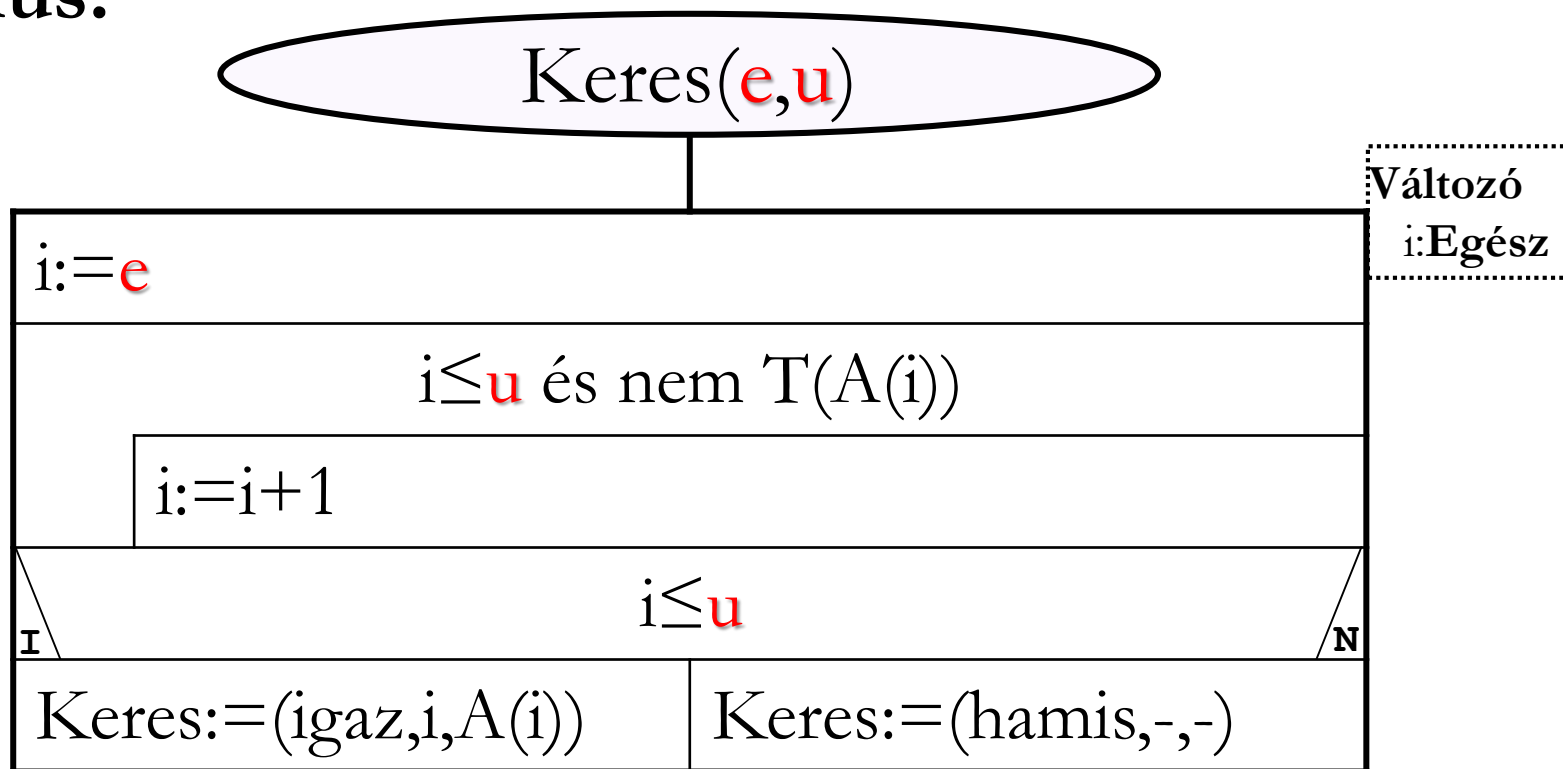
Előfeltétel: –

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



Keresés – intervallumon

Algoritmus:



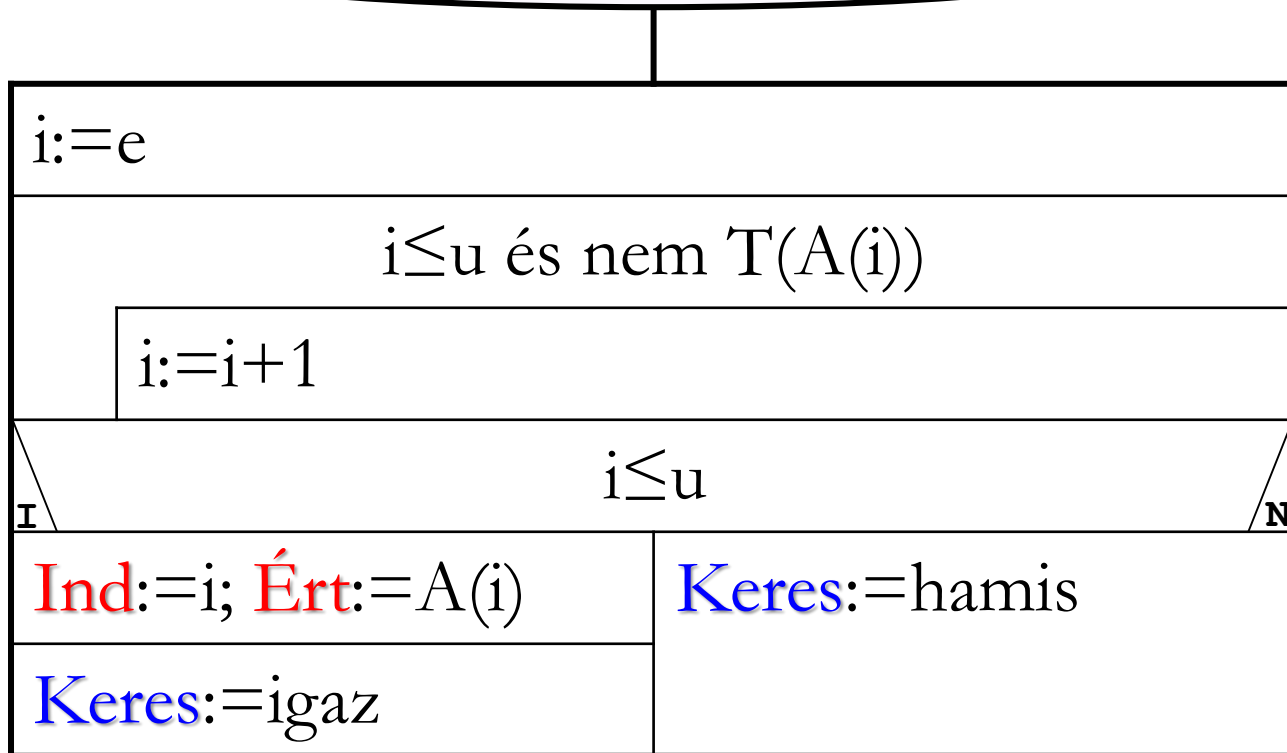
A keresésből eldöntés lesz, ha csak az első, logikai érték a függvény értéke.



Keresés – intervallumon

Algoritmus (függvény két kimenő paraméterrel):

Keres(e,u,Ind,Ért)



Változó
i:Egész



Sorozat \rightarrow multihalmaz transzformáció



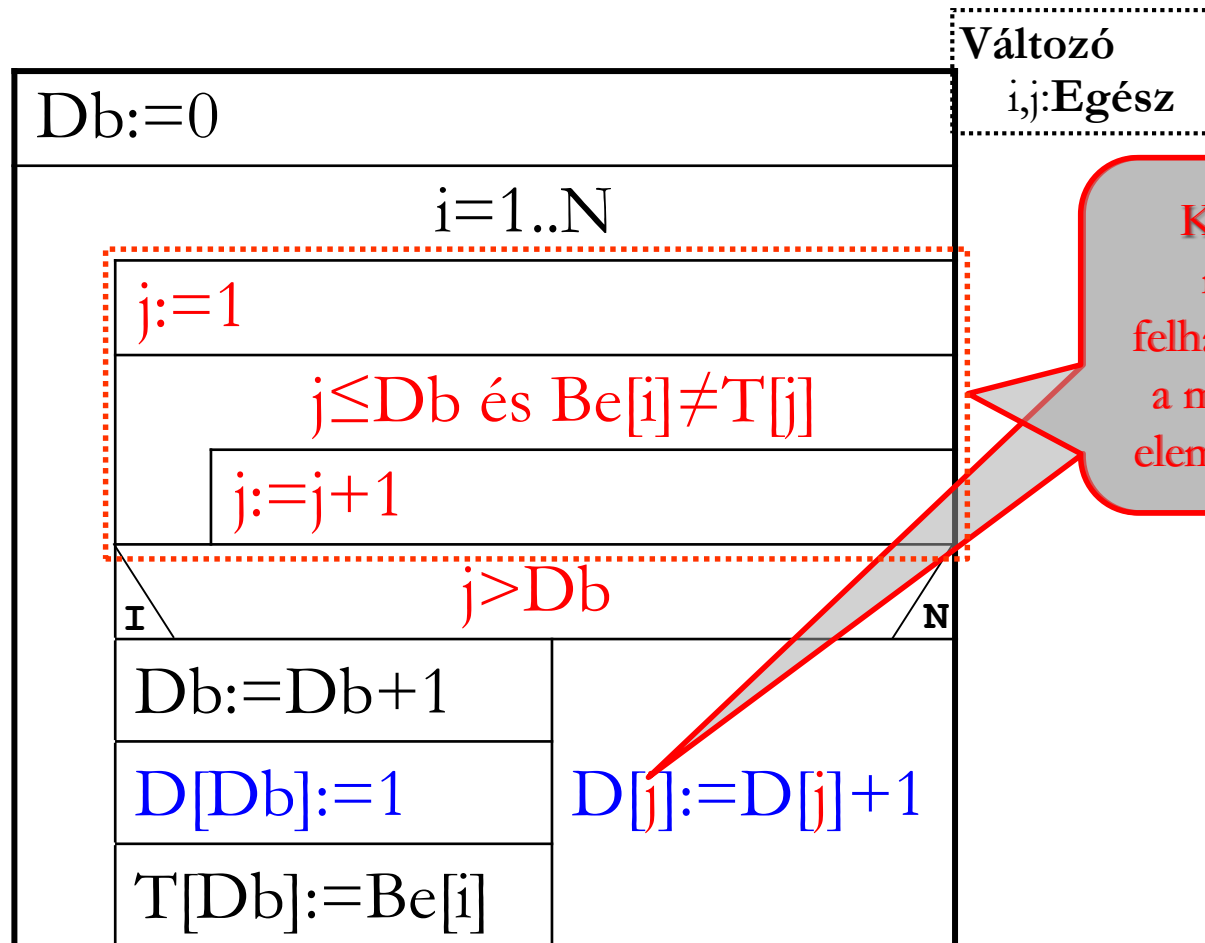
Egyes esetekben a bemenetbeli sorozatból multihalmazt kell készítenünk, ahol az elemek értéke mellett a számosságukat is tároljuk.

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



Sorozat \rightarrow multihalmaz transzformáció



Keresés,
mivel
felhasználjuk
a megtalált
elem indexét



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):
 $H := H - \{e, 1\}$
- 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

Halmaz \times Halmaz műveletek:

- metszet (\cap) (értékek metszete, multiplicitások minimuma)
- unió (\cup) (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) (\subset, \subseteq)
- mindközös? (a két multihalmaz az elemek multiplicitásától eltekintve azonos-e)



Multihalmaz típus

Példa:

Típus

ÁllatFajta=Szöveg

Állatok=Multihalmaz (Fajta)

Változó

A:Állatok; sok:Egész(10)

A:=Állatok(("lúd",13),("disznó",1))

Ha "disznó" \in A és "lúd" \in A **akkor**

Ha Multiplicitás(A,"lúd")>sok és

Multiplicitás(A,"disznó")=1 **akkor**

Ki: "Sok lúd disznót győz!"

Elágazás vége

Elágazás vége



Multihalmaz típus ábrázolása₁

Elemek felsorolása:

Típus

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

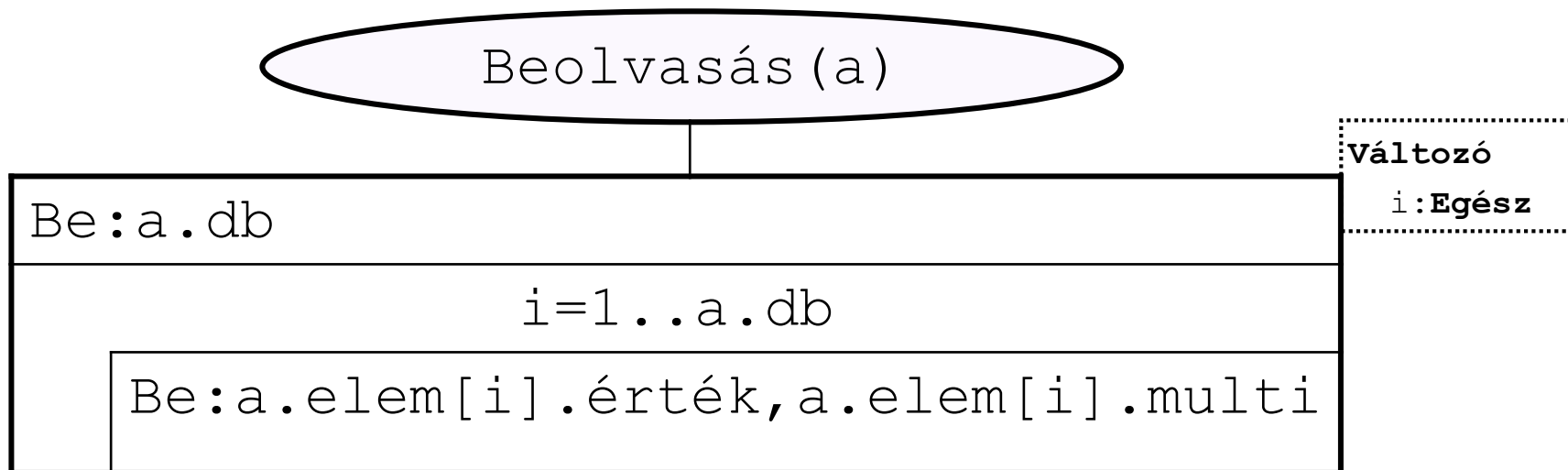
Multihalmaz (Elemtípus)=**Rekord** (
db:Egész,
elem:**Tömb** [1..MaxDb:Halmazelem])

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

Csak a legalább 1 multiplicitású elemeket tároljuk!



Multihalmaz típus

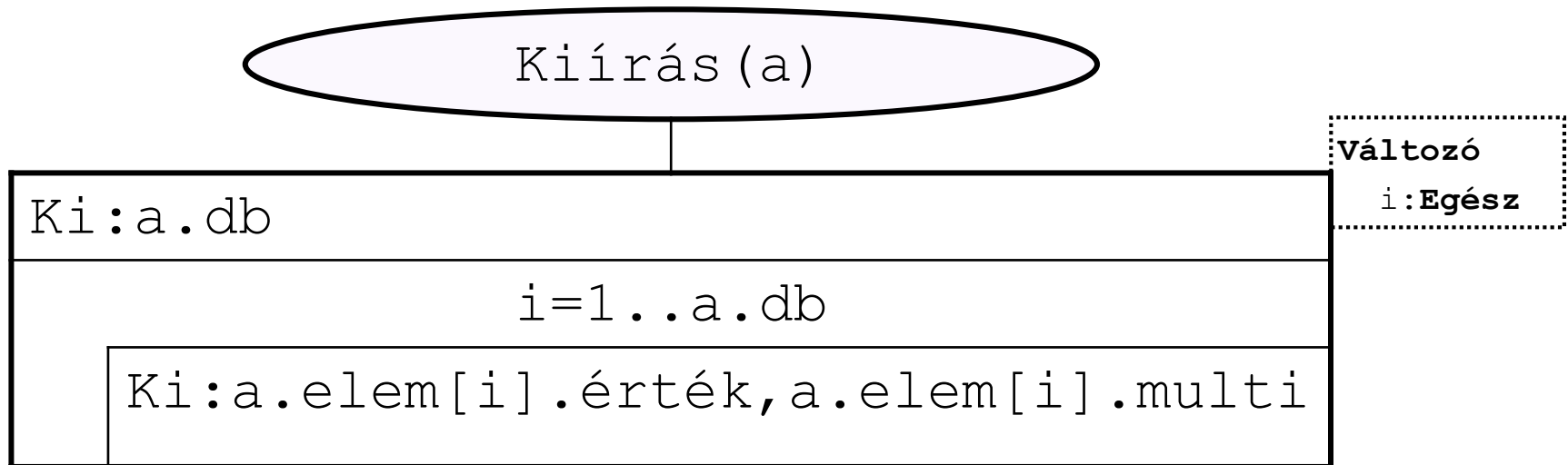


Műveletigény számítása:

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.

Üres? (a) : Logikai

Üres? := a.db = 0

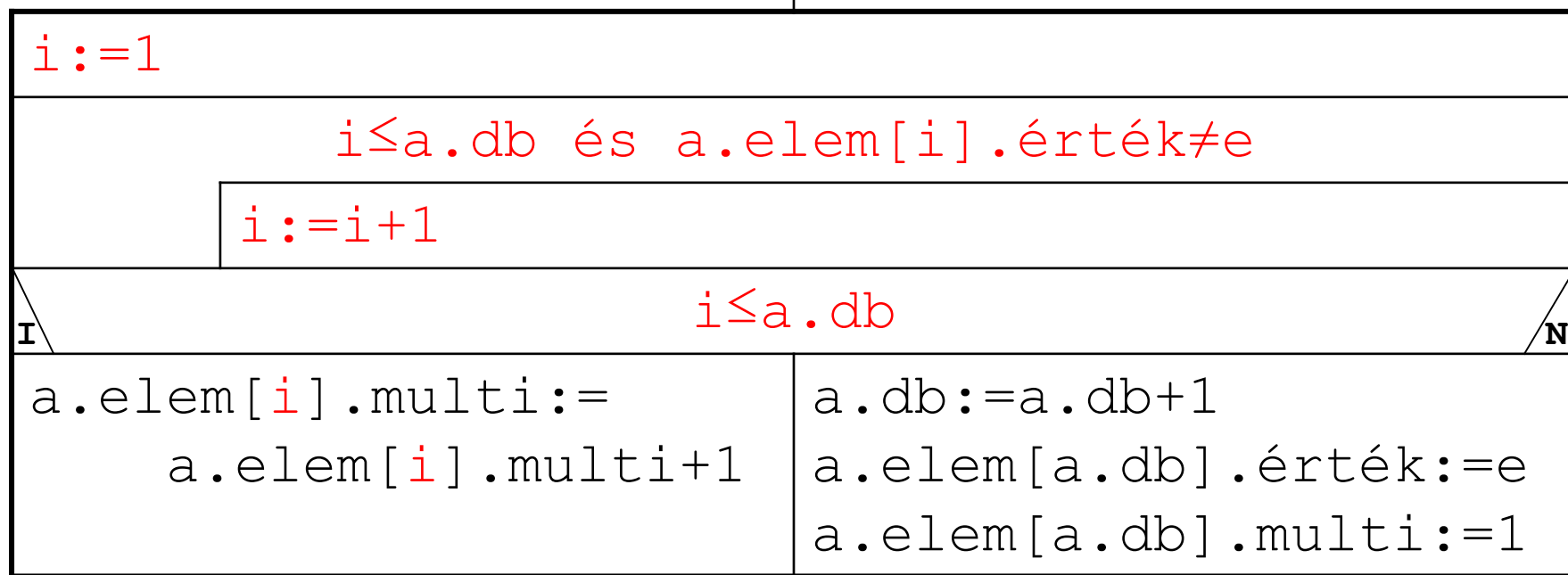
Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

Multihalmazba (a, e)



vált

i :

Műveletigény számítása:

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



Multihalmaz típus

Multihalmazból (a, e)

$i := 1$		
$i \leq a.db$ és $a.elem[i].érték \neq e$		
$i := i + 1$		
$i \leq a.db$		
$a.elem[i].multi = 1$		N
$a.elem[i] :=$ $a.elem[a.db]$ $a.db := a.db - 1$	$a.elem[i].multi :=$ $a.elem[i].multi - 1$	—

Válto

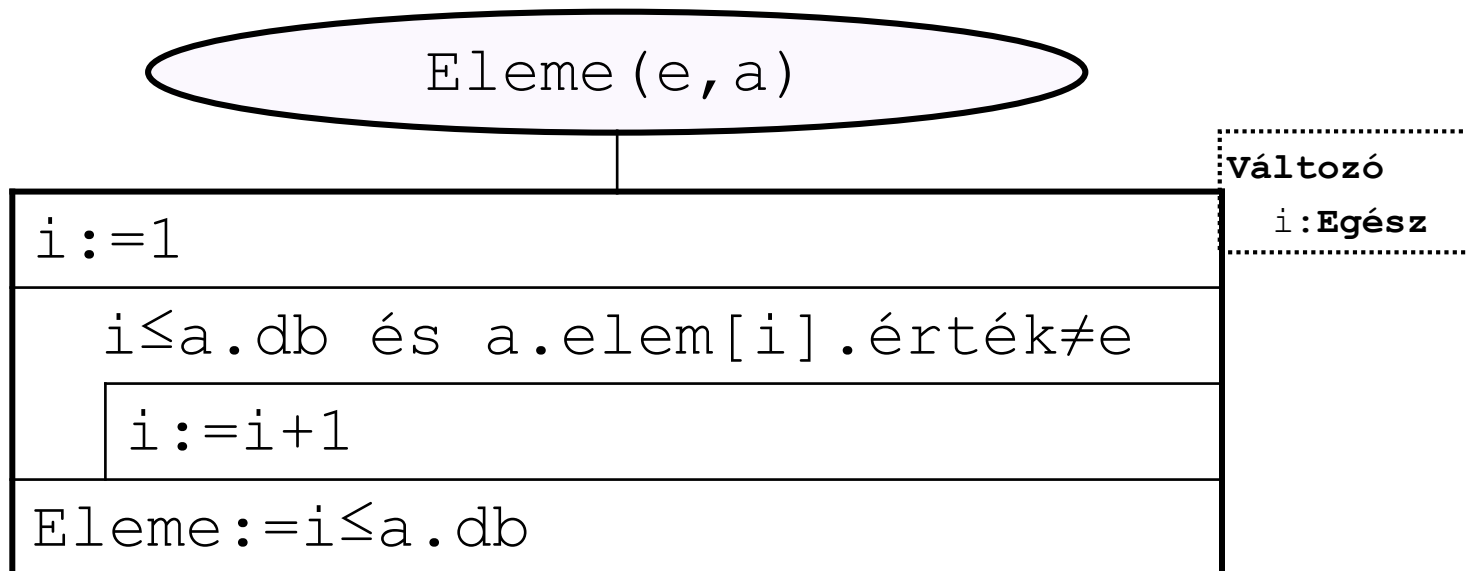
i: E

Műveletigény számítása:

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



Multihalmaz típus



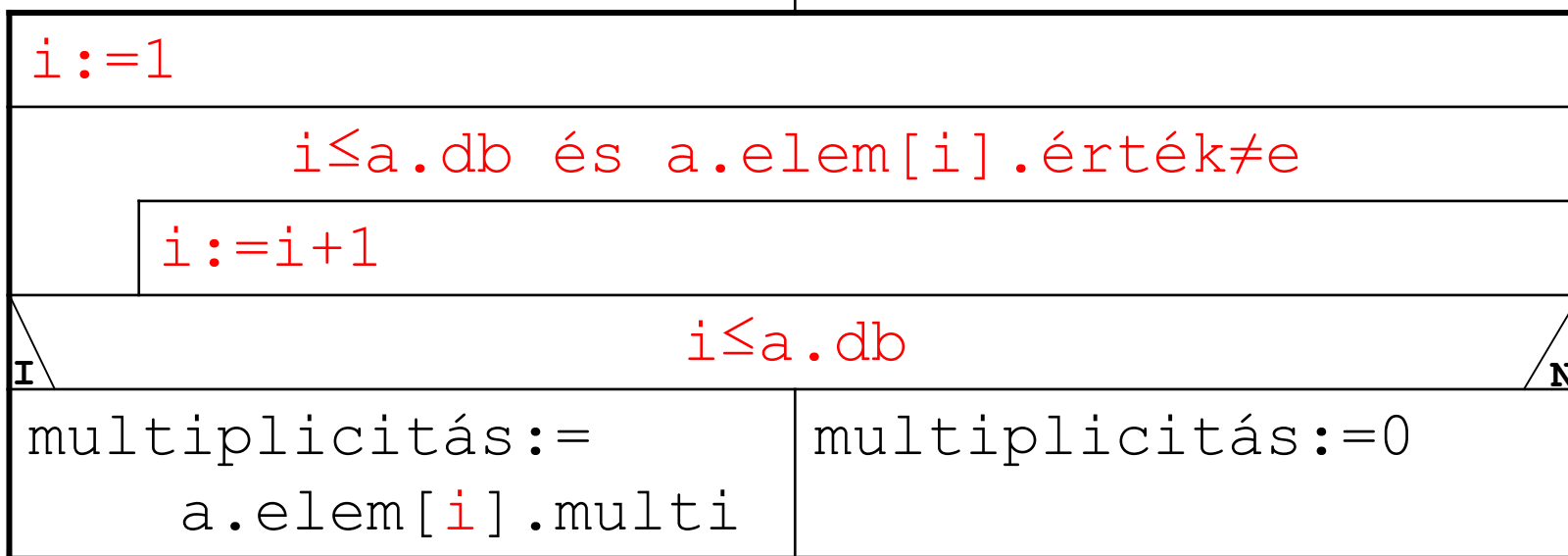
Műveletigény számítása:

Arányos a multihalmaz elemszámával (eldöntés tétel).



Multihalmaz típus

Multiplicitás (a, e)



Változó

i : Egér

Műveletigény számítása:

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



Multihalmaz típus

Halmazelem
típusú:
(érték, multi)

Benne (e, a)

```
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
```

Változó

i : Egész

Műveletigény számítása:

Arányos a multihalmaz elemszámával (keresés tétel).



Multihalmaz típus

Halmazelem

típusú:
(érték, multi)

Része(a, b)

```
i := 1
```

```
i ≤ a.db és Benne(a.elem[i], b)
```

```
  i := i + 1
```

```
Része := i ≤ a.db
```

Változó

i : Egész

Műveletigény számítása:

A külső ciklus az A, a benne műveletben levő belső ciklus a B multihalmaz elemszámánszor 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]		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

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

vál.

i, j

c.db



Multihalmaz típus **ábrázolása₂**

Darabszám vektor:

Típus

Multihalmaz (Elemtípus) =

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

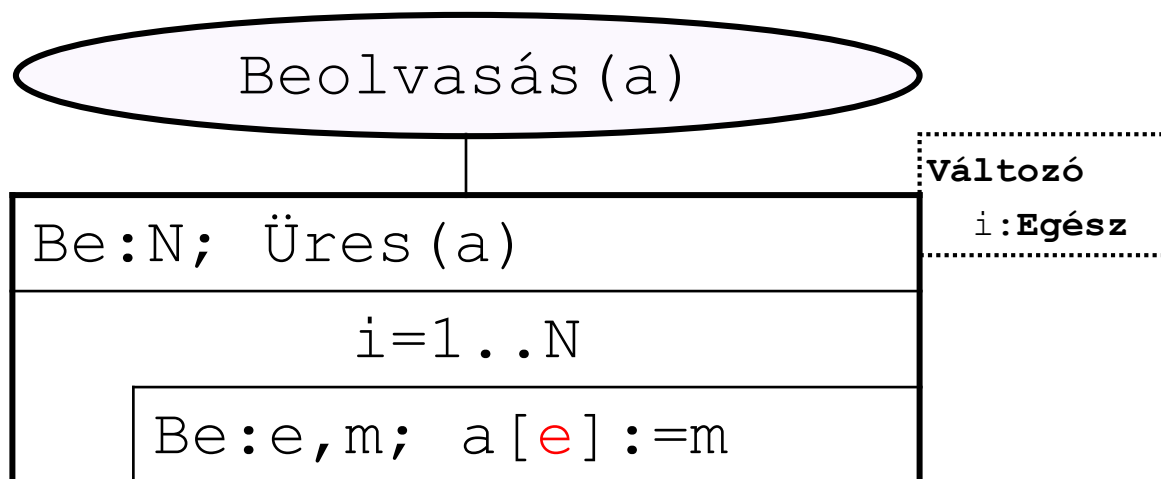
Vegyünk fel egy annyi elemből álló sorozatot, amennyi a multihalmaz lehetséges elem fajtáinak száma!

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 „felsorolható-nak” kell lennie! Ilyenekkel fogunk indexelni!



Multihalmaz típus



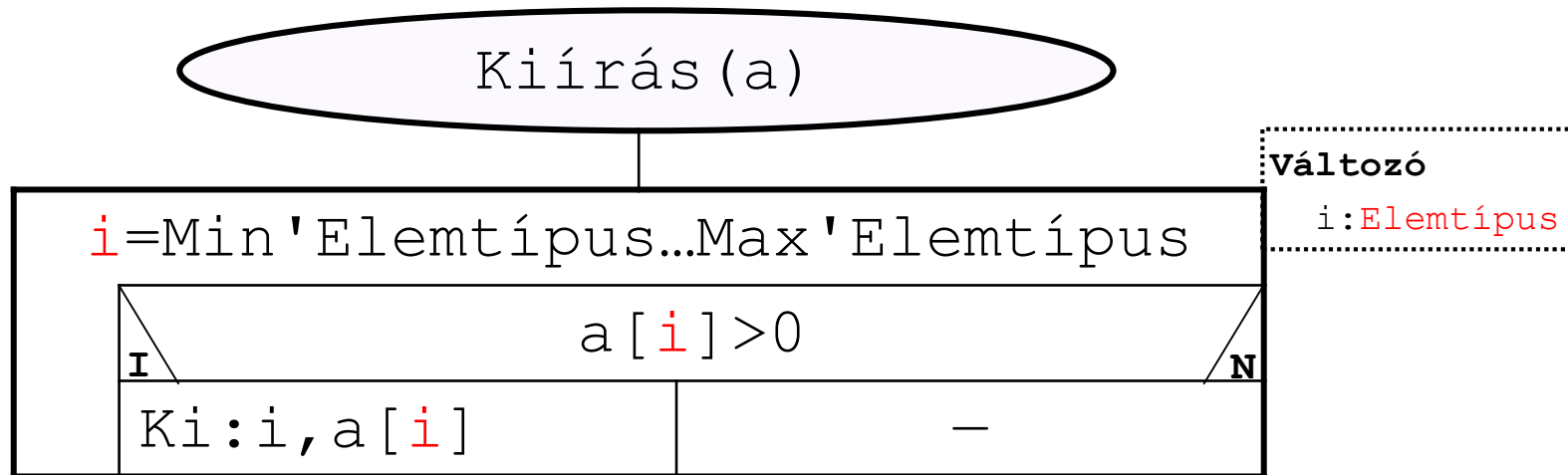
Műveletigény számítása:

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) !



Multihalmaz típus

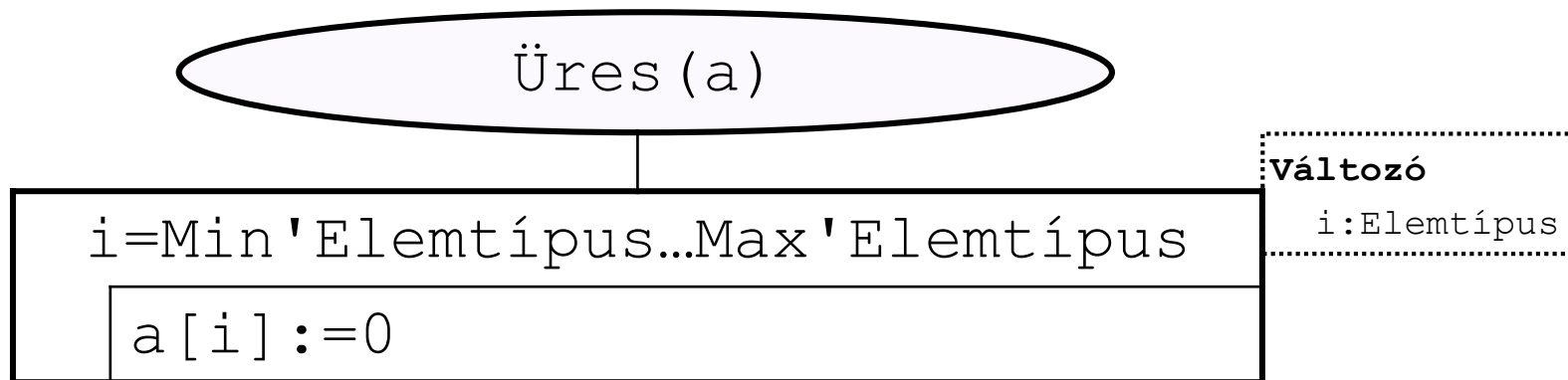


Műveletigény számítása:

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



Multihalmaz típus

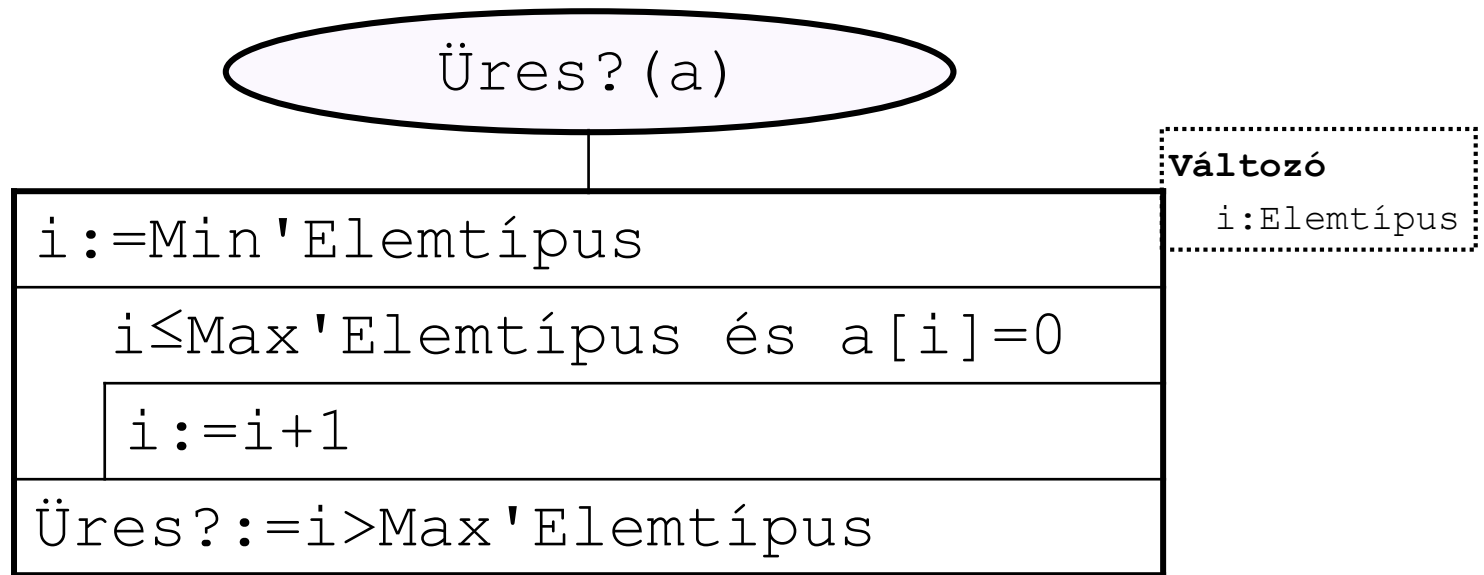


Műveletigény számítása:

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

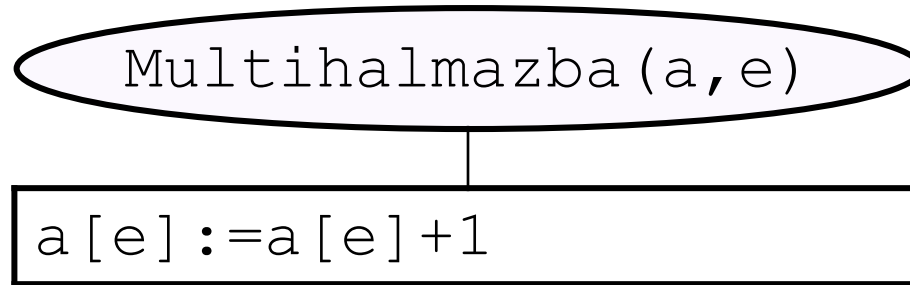


Műveletigény számítása:

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



Multihalmaz típus

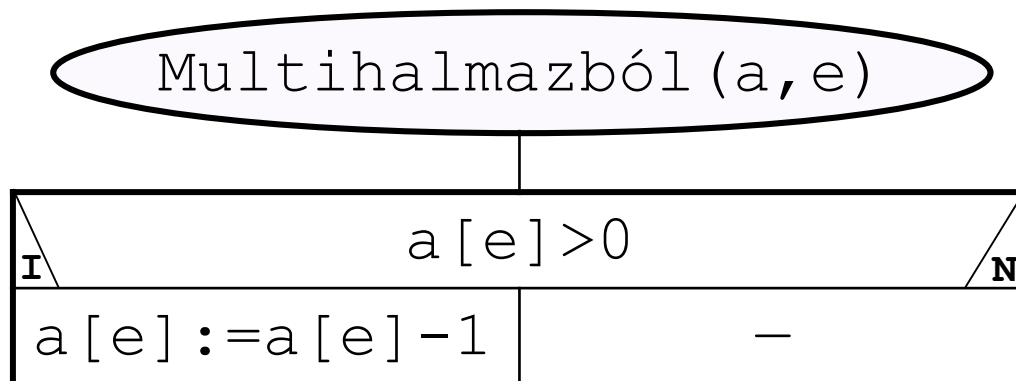


Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

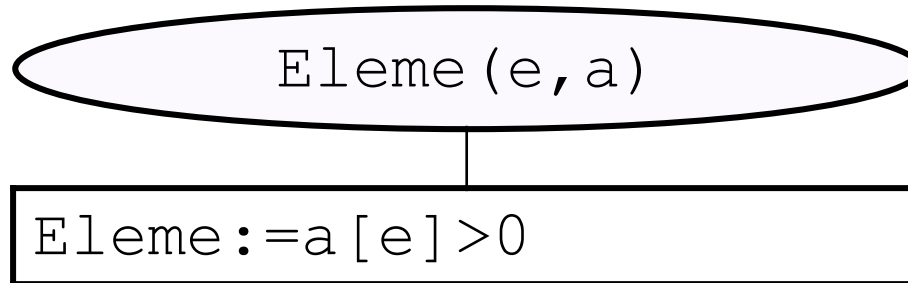


Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

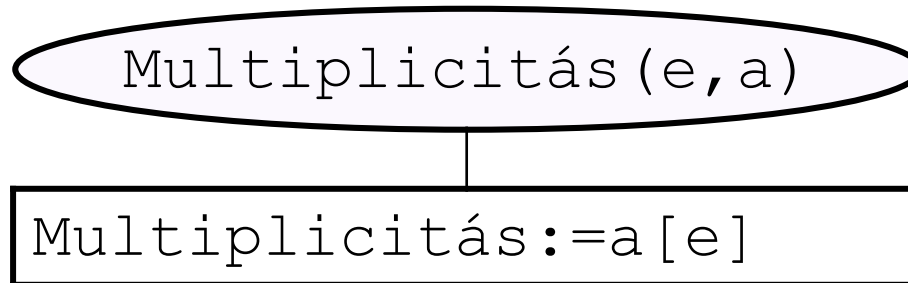


Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

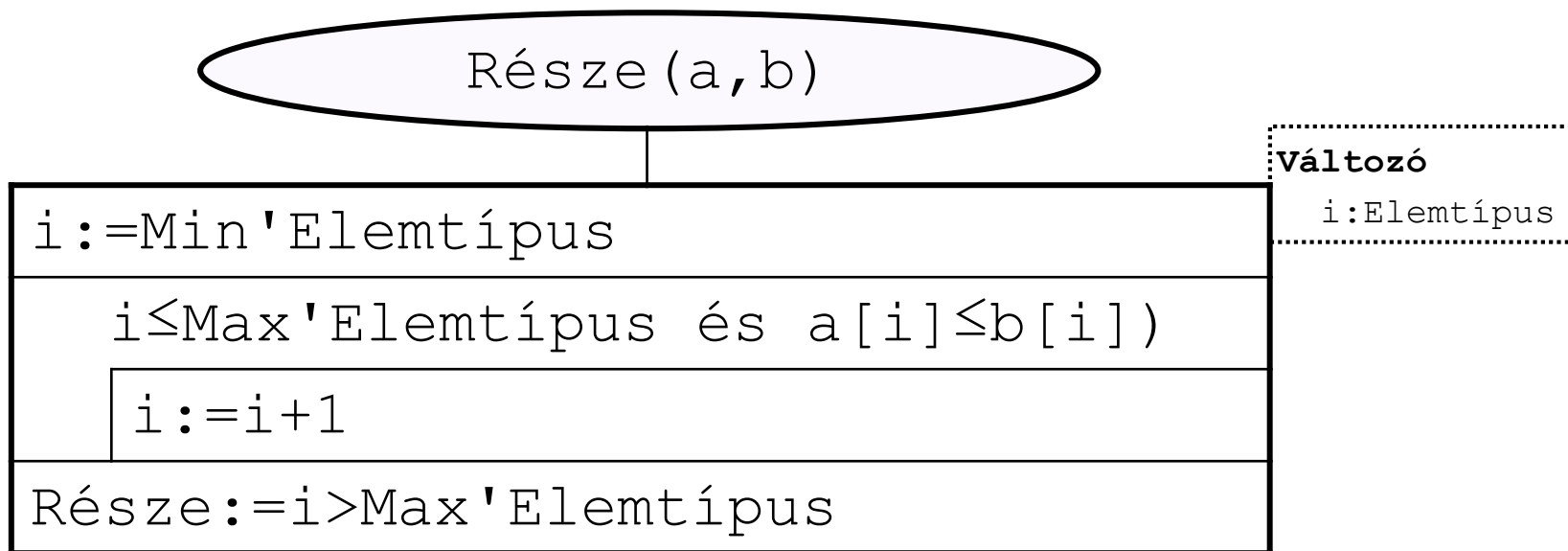


Műveletigény számítása:

Nem függ a multihalmaz elemszámától.



Multihalmaz típus

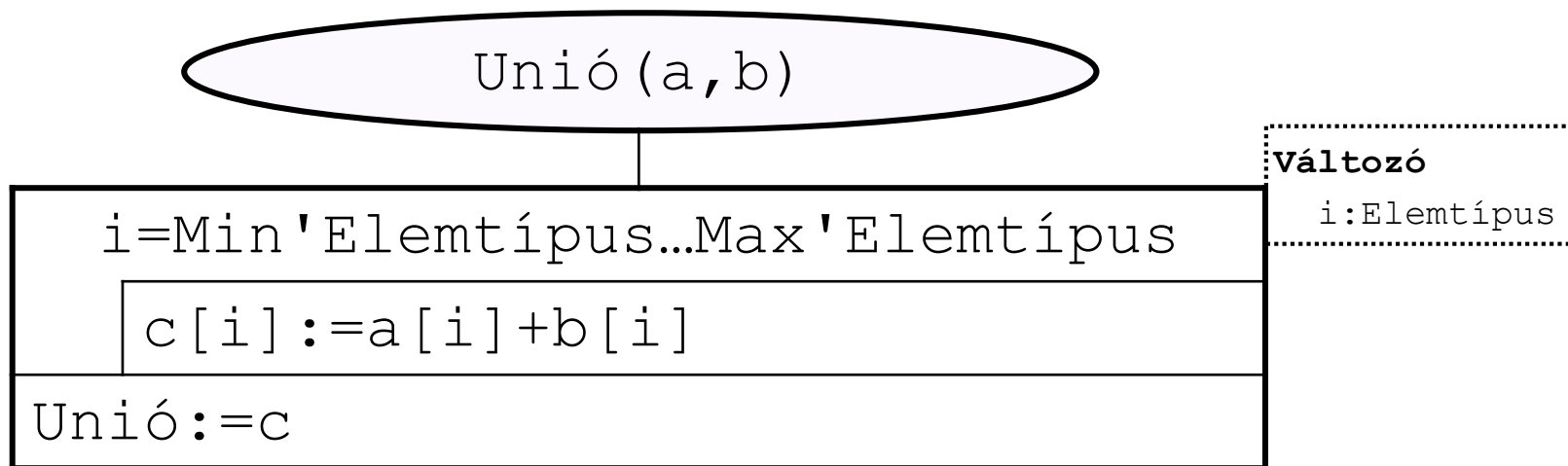


Műveletigény számítása:

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



Multihalmaz típus

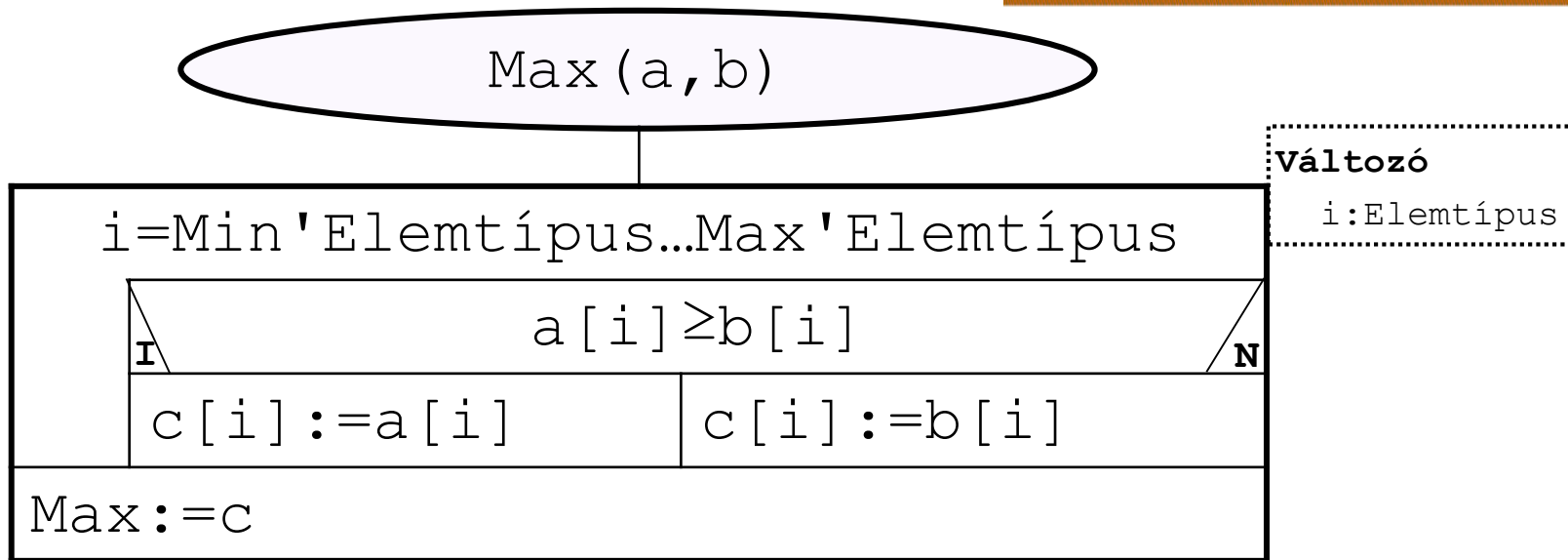


Műveletigény számítása:

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



Multihalmaz típus

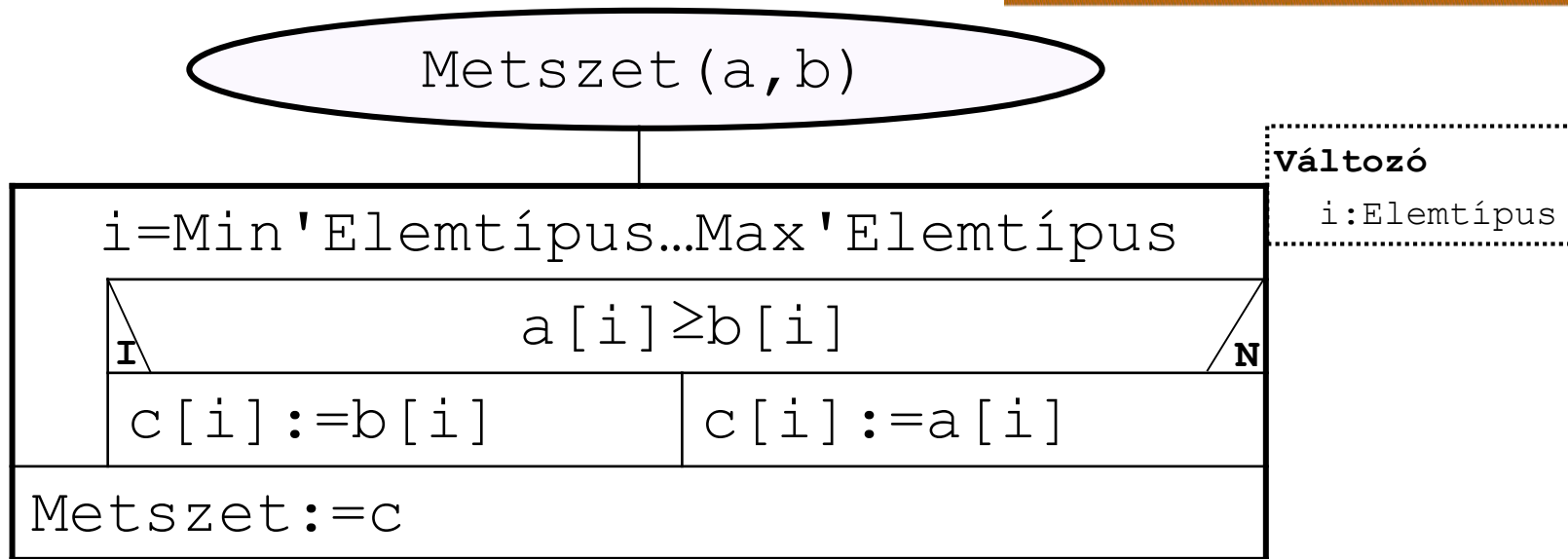


Műveletigény számítása:

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



Multihalmaz típus



Műveletigény számítása:

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



Áttekintés



- Programozási tételek általánosítása
 - Összegzés
 - Megszámolás
 - Maximum-kiválasztás
 - Feltételes Maximum-keresés
 - Kiválasztás
 - Keresés
- Multihalmaz típus
 - Multihalmaz típus elemek felsorolásával
 - Multihalmaz típus darabszám vektorral

