2. táblás gyakorlat – specifikáció, összegzésre visszavezetés

Határozzuk meg természetes számokat tartalmazó tömb legnagyobb elemét. A feladatot összegzésre vezessük vissza.

Az összegzés alapalgoritmusa így szól: induljunk ki egy olyan elemből, amit bármivel „összeműveletelve” azt a bármit kapjuk (legalábbis balról), majd így „műveleteljük” össze sorban az összes elemet (balról – azaz elvileg az absztrakt tételben fontos, hogy $s \coloneqq s + f(i)$-t írunk és nem $s \coloneqq f(i) + s$-t).

Most mivel a legnagyobb elemet keressük, valamiféle összehasonlító művelet kéne a tételbeli „+” helyébe. Legyen ez a $max(a,b)$ művelet, ez ugyanúgy $H \times H \to H$ alakú, mint a $+$, amennyiben a H a természetes számok halmaza. $max(a,b)$ értéke legyen a, ha $a > b$, és b különben.

Ekkor a 0 egy olyan elem, hogy $max(0,b) = b$ lesz bármilyen b természetes számra, hiszen az „a” szerepében levő 0 nem tud nagyobb lenni egy ilyen számnál sem.

Megtaláltuk tehát a *művelet*et és a *nullelem*et is.

Már csak egy fontos dolog maradt. Általában az összegzésnél nem okoz gondot az üres intervallum kezelése. Üres összeg értéke 0, üres szorzaté 1, üres uniózásé üres halmaz, stb. Mindig a nullelem. De az nem igaz, hogy üres tömb legnagyobb eleme a 0 lenne. Üres tömbnek nincs legnagyobb eleme, nem is értelmes ezt a kérdést feltenni rá. Éppen ezért kikötjük az előfeltételben, hogy a tömb nem lehet üres. Ez mutatja – a nem igazán erre az esetre szánt – összegzés tétel relatív rugalmasságát. Ha a célnak megfelelőbb maximumkiválasztás tételt használtuk volna, ott már eleve – a tétel részeként – kikötöttük, hogy nem lehet üres az intervallum.

$$A = (t : \mathbb{N}^n, maximum : \mathbb{N})$$

$$ef = (t = t' \wedge n > 0)$$

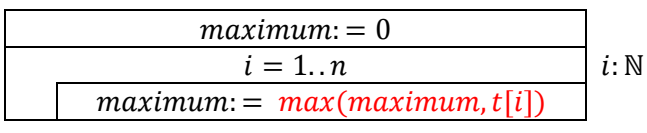
$$uf = \left(ef \wedge maximum = \operatorname*{MAX}_{i=1}^{n} (t[i]) \right),$$

ahol MAX a max művelet nagyoperátor változata.

Visszavezetés (összegzés):

H	$\sim$	$\mathbb{N}$
$[m..n]$	$\sim$	$[1..n]$
$f(i)$	$\sim$	$t[i]$
s	$\sim$	$maximum$
$+, 0$	$\sim$	$max, 0$

Struktogram:



Persze a max egy nem megengedett függvény, szóval definiáljuk, és helyettesítsük:

$$max: \mathbb{N} \times \mathbb{N} \to \mathbb{N}, \text{ és}$$

$$\forall i,j \in \mathbb{N}: max(i,j) = \begin{cases} i, & ha\ i > j \\ j, & kül. \end{cases}$$

$maximum := 0$		
$i = 1..n$		$i: \mathbb{N}$
	$maximum > t[i]$	
	$SKIP$	$maximum := t[i]$

Megjegyzések:

- a struktogramban a „$max(maximum, t[i])$” kifejezésben a két argumentum sorrendje nem mindegy! Nem véletlenül a $maximum$ van elöl. Ez az, amelyik kezdetben a 0 értéket kapja, ami *baloldali nullelem*, ezért kell neki baloldalon lennie, hogy $i = 1$ esetben megkaphassa a $maximum$ a $t[1]$ értékét garantáltan. Természetesen jelen esetben (is) a 0 kétoldali nullelem, de ha be akarjuk tartani a tételt, akkor fontos, hogy a tételbeli „s” (azaz a „$maximum$”) bal oldalon legyen.
- Viszont ezek után ennek a strukogramnak a kódolása során már nyilván alkalmazhatunk ekvivalens átalakításokat:

```
if(maximum>t[i])
{ }
else
{
  maximum = t[i];
}
```

helyett írhatjuk ezt:

```
if(maximum<=t[i])
{
  maximum = t[i];
}
else
{ }
```

Innen:

```
if(maximum<=t[i])
  maximum = t[i];
```

De akár itt és most ez is helyes (mert maximum kezdetben 0, és legalább egy elem van):

```
if(maximum<t[i])
  maximum = t[i];
```

Még egy helyes változat:

```
maximum = maximum<t[i] ? t[i] : maximum;
```

- A $max(a,b)$ ugyanolyan binér művelet, mint a $+$, csak épp nem infix alakban írtuk le. A $+$-t is leírhatnám így: $+(a,b)$ és a maxot is leírhatnám akár így: $a\ max\ b$.
- Ha a feladat minimumkeresés lenne, már nem oldhatnám meg ezzel a tétellel, mert a „min” művelethez már nem tudok baloldali nullelemet felmutatni. Nincs olyan természetes szám, ami garantáltan nem kisebb más természetes számoknál. Ha az előfeltételben kiköthetném, hogy a tömb elemei valamilyen rögzített k-nál nem nagyobbak, akkor ez a k lehetne a nullelem, és akkor már minimumkiválasztásra is alkalmazható lenne ez a tétel, természetesen ekkor is csak amellett a feltétel mellett, miszerint a tömb nem lehet üres.
- Akkor se lenne alkalmazható a tétel, ha a teljes egész (vagy akár racionális) számok körében kutakodnánk, vagy bármely olyan halmazban, ahol nincs legkisebb elem. Persze, hasonlóan, mint a minimumkeresésnél, itt is kinevezhetünk egy mesterséges minimumot, ami nem is olyan nagyon légbőlkapott ötlet, hiszen ha szeretnénk lekódolni is a megtervezett programot, úgyse használhatjuk az egész számok elvben végtelen számosságú halmazát, lesz mindig egy legkisebb elem, mert az ábrázolhatóság illetve a memória határt szab. Lehet ez a legkisebb ábrázolható elem ekkor a „nullelem”.