***Programozási tételek intervallumra***

## *1. Összegzés*

*Feladat*: Adott egy $f$:[$m$..$n$]→$H$ függvény. A $H$ halmaz elemein értelmezett egy asszociatív, baloldali nulla elemmel rendelkező művelet (nevezzük ezt összeadásnak és jelölje a +). Határozzuk meg a függvény intervallumon felvett értékeinek összegét!

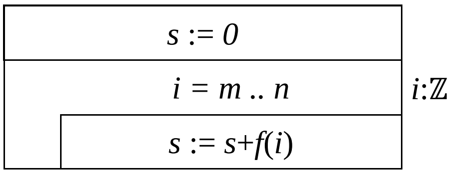
*Specifikáció*:

$A = (m{:}\mathbb{Z},\ n{:}\mathbb{Z}\ ,\ s{:}H)$

$Ef = (\ m=m' \wedge n=n')$

$Uf = (\ Ef \wedge s = \sum_{i=m..n} f(i))$

*Algoritmus*:




## *2. Számlálás*

*Feladat*: Adott egy $felt$:[$m$..$n$]→𝕃 feltétel. Határozzuk meg, hogy hányszor teljesül az intervallumon a feltétel, azaz hányszor veszi fel az igaz értéket!

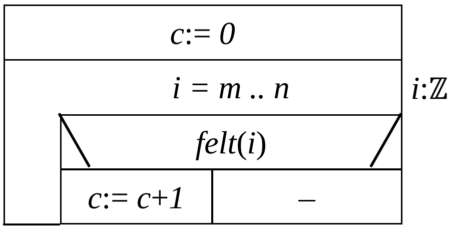
*Specifikáció*:

$A = (m{:}\mathbb{Z},\ n{:}\mathbb{Z},\ c{:}\ \mathbb{N})$

$Ef = (\ m=m' \wedge n=n')$

$Uf = (\ Ef \wedge c = \sum_{\substack{i=m..n \\ felt(i)}} 1\ )$

*Algoritmus*:




## *3. Maximum kiválasztás*

*Feladat*: Adott egy $f$:[$m$..$n$]→$H$ függvény. A $H$ halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, melyik a függvény legnagyobb értéke és adjuk meg az egyik olyan intervallumbeli elemet, ahol a függvény ezt az értéket felveszi!

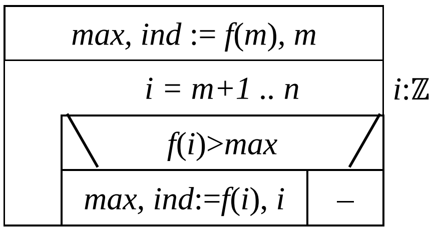
*Specifikáció*:

$A = (m{:}\mathbb{Z},\ n{:}\mathbb{Z},\ max{:}H,\ ind{:}\mathbb{Z})$

$Ef = (m=m' \wedge n=n' \wedge m \leq n\ )$

$Uf = (\ Ef \wedge\ max,ind = MAX_{i=m..n} f(i)\ )$

*Algoritmus*:



### *4. Feltételes maximumkeresés*

*Feladat*: Adott egy $f$:[$m..n$]→$H$ függvény és egy $felt$:[$m..n$]→$\mathbb{L}$ feltétel. A $H$ halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, melyik a függvény legnagyobb értéke azok között, amelyeket olyan intervallumbeli elemhez rendel, amelyek kielégíti a feltételt! Adjuk meg az egyik olyan intervallumbeli elemet, amelyre a feltétel teljesül és ahol a függvény ezt a maximális értéket felveszi!

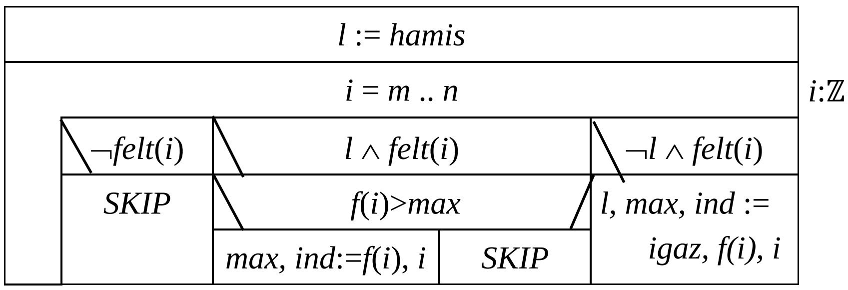
*Specifikáció*:

$A = (m:\mathbb{Z},\ n:\mathbb{Z},\ l:\mathbb{L},\ ind:\mathbb{Z},\ max:H)$

$Ef = (\ m=m' \wedge n=n'\ )$

$$Uf = (\ Ef \wedge (l,max,ind) = \underset{felt(i)}{MAX_{i=m..n}}\, f(i)$$

*Algoritmus*:




***Megjegyzés***: A fenti programozási tételek rugalmasságát mutatják az alábbiak.

1. Az indexet megadó eredményváltozó elhagyható, ha nincs rá szükség
   - maximum kereséseknél, lineáris keresésnél (eldöntés)
2. Minimum keresés
   - Az algoritmus szempontjából mindegy, hogy a „>” vagy a „<” relációt használja. (Specifikációban: MAX helyett MIN)
3. Legutolsó elem keresése
   - Maximum kereséseknél: *f(i)>max* helyett *f(i)≥max*

***5. Kiválasztás (szekvenciális vagy lineáris kiválasztás)***

*Feladat*: Adott egy *felt*:$\mathbb{Z}\rightarrow\mathbb{L}$ feltétel és egy *m* egész szám. A feltétel az *m*-nél nagyobb vagy egyenlő egész számokra van értelmezve, legalábbis az első olyan egész számig, ahol a feltétel igaz értéket vesz fel (teljesül). Ilyen egész szám biztosan van. Határozzuk meg az *m*-nél nagyobb vagy egyenlő legelső olyan egész számot, amelyre a feltétel teljesül!

*Specifikáció*:

$A = (m:\mathbb{Z}, i:\mathbb{Z})$

$Ef = ( m=m' \wedge \exists k\geq m:felt(k) )$

$Uf = ( Ef \wedge i = SELECT_{i\geq m} felt(i))$

*Algoritmus*:

| |
|---|
| *i*:= *m* |
| $\neg$*felt*(*i*) |
| *i* := *i*+*1* |

***6. Keresés (szekvenciális vagy lineáris keresés)***

*Feladat*: Adott egy *felt*:[*m*..*n*]$\rightarrow\mathbb{L}$ feltétel. Határozzuk meg az intervallum első olyan elemét, amelyre teljesül a feltétel! (Ez az ún. pesszimista lineáris keresés.)

*Specifikáció*:

$A = (m:\mathbb{Z}, n:\mathbb{Z}, l:\mathbb{L}, ind:\mathbb{Z})$

$Ef =( m=m' \wedge n=n' )$

$Uf = (Ef \wedge l, ind = SEARCH_{i=m..n} felt(i))$

*Algoritmus*:

| | |
|---|---|
| *l*, *i*:= *hamis*, *m* | *i*:$\mathbb{Z}$ |
| $\neg$*l* $\wedge$ *i*$\leq$*n* | |
| *l*, *ind* := *felt*(*i*), *i* | |
| *i* := *i*+*1* | |

Ez az algoritmus használható az eldöntés típusú feladatok megoldásához is, csak ilyenkor mind a specifikációból, mind a programból elhagyhatjuk az ind változót és az azzal kapcsolatos részeket. Feladat: Van-e olyan eleme az intervallumnak, amelyre teljesül a feltétel?

Az optimista lineáris keresés azt vizsgálja, hogy teljesül-e az intervallumnak minden elemére a feltétel, és megadja az első olyan indexet, amelyre nem. Ennek egyszerűbb változata az optimista eldöntés, amikor a specifikációból és a programból elhagyhatjuk az ind változót és az azzal kapcsolatos részeket.

*Specifikáció*:

$A = (m:\mathbb{Z}, n:\mathbb{Z}, l:\mathbb{L})$

$Ef =( m=m' \wedge n=n' )$

$Uf = (Ef \wedge \mathrm{l} = \forall SEARCH_{i=m..n} felt(i))$

*Algoritmus*:

| | |
|---|---|
| *l*, *i*:= *igaz*, *m* | *i*:$\mathbb{Z}$ |
| *l* $\wedge$ *i*$\leq$*n* | |
| *l* := *felt*(*i*) | |
| *i* := *i*+*1* | |

***Megjegyzés***: Az első adott tulajdonságú elem keresése helyett az utolsót is kereshetjük, ha i:=i+1 helyett az *i:=i-1*-et használjuk.

### *7. Logaritmikus keresés*

*Feladat*: Adott az egész számok egy [$m..n$] intervalluma és egy $f:\mathbb{Z}\to H$ függvény, amelyik az [$m..n$] intervallumon monoton növekvő. (A $H$ halmaz elemei között értelmezett egy rendezési reláció.) Keressünk meg a függvény [$m..n$] intervallumon felvett értékei között egy adott értéket!

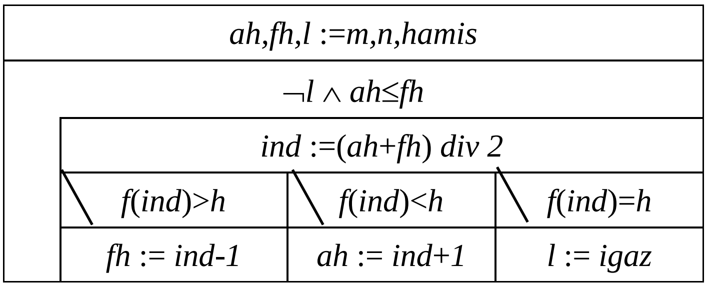
*Specifikáció*:

$A = (m:\mathbb{Z}, n:\mathbb{Z}, h:H, l:\mathbb{L}, ind:\mathbb{Z})$

$Ef = ( m=m' \wedge n=n' \wedge h=h' \wedge \forall j\in[m..n-1]: f(j) \leq f(j+1) )$

$Uf = ( Ef \wedge l=( \exists j\in[m..n]:f(j)=h ) \wedge l\to( ind\in[m..n] \wedge f(ind)=h ) )$

*Algoritmus*:




$ah, fh:\mathbb{Z}$