Programozási tételek specifikációja

Alapvető fogalmak és jelölések:

H halmaz	• tetszőleges (véges) halmaz a szokásos halmaz műveletekkel és konstanosokkal ($\in,\notin,\subset,\subseteq,\cup,\cap,\setminus,\emptyset$, Számosság)
L	• *logikai értékek* halmaza: {Igaz,Hamis} a szokásos műveletekkel ($\neg,\wedge,\vee,\Leftrightarrow,\Rightarrow$), kvantorokkal ($\forall,\exists$) és indivíduum változokkal, halmazokkal[1]
N	• *természetes számok* halmaza {0, 1, ...} a szokásos műveletekkel (+,-,*,Div,Mod)
Z	• *egész számok* halmaza {.., -1, 0, 1, ...} a szokásos műveletekkel (+,-,*,Div,Mod)
R	• *valós számok* halmaza a szokásos műveletekkel (+,-,*,/,^)
C	• *karakterek* halmaza {…, ’ ’,…, ’A’,…,’z’, …}
S	• *szövegek* halmaza
H-Sorozat	• H-beli elemek N-esek (N∈**N**), amelyen alapműveletként értelmezzük az *i. elem* ($i\in\{1..N\}$) kiválasztását: s_i:= az S sorozat i. eleme[2], valamint az *eleme* relációt: x∈S, ha $\exists i$: x=s_i, és a *sorozathossz* függvényt, ekkor Hossz(S)=N
H^N halmaz N∈**N**	• H-beli N-*elemű sorozatok* halmaza
H^* halmaz	• H-beli *véges sorozatok* halmaza, azaz $H^* = \cup_{(i=0..\infty)} H^i$
()∈H^* *üres sorozat*	• S=() ⇔ Hossz(S)
[x..y] *index-intervallum* *x,y*∈**N**	• [x..y]:={x,x+1,...,y-1,y}
I[x..y] *index-sorozat*	• I[x..y]:=(x,x+1,...,y-1,y)∈$\mathbf{N}^{y-x}$, ha y≥x I[x..y]:=(), egyébként
$\mathcal{F}$(H,S) *függvényhalmaz* *H-ból S-be képez*	• $\mathcal{F}$(H,S):={ f : f:H→S }
Rendezés($\le_H$) $\le_H \in \mathcal{F}$(HxH,**L**) *rendezési reláció (infix jelölésű bináris függvény)* *ha lehet:* $\le_H$ *helyett* $\le$	• ha ∀a∈H: a≤a (reflexív), ∀a,b,c∈H: a≤b és b≤c ⇒ a≤c (tranzitív), ∀a,b∈H: a≤b és b≤a ⇒ a=b (antiszimmetrikus) ∀a,b∈H: a≤b vagy b≤a (teljes)
H halmaz *RendezettHalmaz*$_\le$*(H)*	• ha ∀h,j∈H: h≤j vagy h≥j (ha H véges ⇒ ∃maximális, minimális eleme)
S∈H^* sorozat *RendezettSorozat*$_\le$*(S)*	• ha RendezettHalmaz$_\le$(H) és ∀*i*∈[1..N]: $s_i \le s_{i+1}$
S∈H^* sorozat *HalmazFölsorolás(S)*	• ha ∀*i,j*∈[1..N]: *i*≠*j* ⇒ $s_i \ne s_j$
X&Y *konkatenáció*ja X és Y sorozatoknak	• X&Y:=(x_1,...,x_N,y_1,...,y_M), ahol N=Hossz(X) és M=Hossz(Y)
X⊆Y X sorozat *részsorozata* Y sorozatnak	• ∀x∈X: ∃y∈Y: x=y és *i,j*∈[1..Hossz(X)]: *i*<*j*: x_i,x_j∈X ⇒ ∃*k,l*∈[1..Hossz(Y)]: *k*<*l*: (x_i=y_k és x_j=y_l)

[1] Pl. „**∃i∈H: T(i)**” logikai kifejezésben az i – individuumváltozó, a H – az individuumhalmaz. A kifejezést így értjük: „**létezik olyan H-beli i elem, amelyre teljesül a T predikátum**”.

[2] Vegyük észre a jelölési „finomságot”: a sorozatot nagybetűvel, de annak valamely elemét kisbetűvel fogjuk jelölni.

$R\backslash r$ sorozat $r \in H$, $R \in H^N$: HalmazFölsorolás(R)	• ha $r \in R \Rightarrow R\backslash r \in H^{N-1}$ és $R=R_1\&(r)\&R_2 \Rightarrow R\backslash r=R_1\&R_2$ ha $r \notin R \Rightarrow R\backslash r=R$
$\mathcal{P}(R)$ az $R \in H^N$ *Permutációhalmaza*	• $\mathcal{P}(R):=\{S \in H^N : N=1 \Rightarrow S=R,$ $N>1 \Rightarrow S=s\&S'$ és $s \in R$ és $S' \in \mathcal{P}(R\backslash s)\}$
X,Y a Z *felbontása* (X,Y,Z sorozatok)	• ha $Z=X\&Y$
$T_i:H \rightarrow \mathbb{L}$ $i \in [1..K]$ H-*partícionálása*	• $\forall x \in H: \vee_{(i=1..K)} T_i(x)=$Igaz és $(T_i(x) \wedge T_j(x)=$Hamis $(i \neq j))$

Definíció: Az $F:H^* \rightarrow S^*$ feladat *elemenként feldolgozható*, $\forall x \in H^*$ $\forall \underline{\underline{x}},\underline{x}$ x-felbontásra igaz, hogy $F(\underline{\underline{x}})\&F(\underline{x})=F(x)$.

Lemma: Ha az $F:H^* \rightarrow S^*$ feladat elemenként feldolgozható, akkor $\exists f:H \rightarrow S$ függvény, hogy $f(h_i)=s_i$ $i \in [1..N]$.

Tételek szerkezete: bemenet, kimenet, előfeltétel, utófeltétel

A függvényként (is) szerepeltetés célja, hogy már „ránézésre” is kitünjön, mely sémák (absztrakt algoritmusok) melyekkel kombinálódhatnak. Ez később nagy segítségünkre lesz a bonyolultabb feladatok megoldásánál.

Megállapodások:

- A tételeket mint *függvény*eket is megadjuk (a „fejsorban”, az ún. szignatúrában), amelynek argumentuma tartalmazza azokat a „kellékeket”, amelyek a szükséges bemenetei, ill. az értékét, amely halmazba képez. A tétel-függvény paramétereinek sorrendje megegyezik a ’*Be*’-részben szereplő adatok sorrendjével. A kimenet halmaza megegyezik a ’*Ki*’-részben felsorolt adatok értékhalmazának direktszorzatával. A sorozatokat gyakorta két adattal írjuk le: a hosszával és az elemeket tartalmazó adatszerkezettel. E jelöléssel hívjuk föl a figyelmet arra, hogy pusztán ilyen „szintaktikai” természetű dolog (mint a paraméterek, ill. eredményhalmazok száma és mibenléte) is segítségünkre lehet a tételek összeépítésénél. A tétel-függvény szignatúrájának és a ’*Be*’+’*Ki*’-résznek a kapcsolatát segíti értelmezni a 0.-ként említett tétel színezése.
- Szívesebben adunk meg egy sorozatot H^*-beliként (H^N helyett), mivel a specifikációban legkevésbé így köti meg a kezünket (nem utal a leírás *tömb*re vagy más *fixhosszúságú szerkezet*re). Ekkor viszont a sorozat és hossza –mint rendszerint megjelenő bemeneti információ– „kapcsolat” nélkül marad; ezt a hiányzó információt kell beilleszteni az előfeltételbe, felhasználva a sorozatokra előredefiniált *Hossz függvény*t. A „tétel-függvényben” éppen ezen megfontolásból nem tüntetjük föl a sorozatok elemszámát szimbolizáló paramétereket.

0. Másolás$\big(H^*,\mathcal{F}(H,S)\big):S^*$

Be: $N\in\mathbb{N},\ X\in H^*,\ f:H\to S$
Ki: $Y\in S^*$
Ef: $\mathrm{Hossz}(X)=N \wedge$ *a feladat elemenként feldolgozható az f függvénnyel*
Uf: $\mathrm{Hossz}(Y)=N \wedge \forall i\in[1..N]:\ y_i=f(x_i)$

1. Sorozatszámítás$\big(H^*,\mathcal{F}(H^*,H)\big):H$ lehetne még **Sorozatszámítás**$\big(H^*,\mathcal{F}(H^*,H_2)\big):H_2$

Be: $N\in\mathbb{N},\ X\in H^*,\ F:H^*\to H$
Ki: $S\in H$
Ef: $\mathrm{Hossz}(X)=N \wedge$
$(\exists f:H\times H\to H \wedge \exists f_0\in H):\ (\ F((x_1,..,x_N))=f(F(x_1,..,x_{N-1}),x_N)\ \wedge\ F(())=f_0\ \wedge\ \forall h\in H: f(f_0,h)=h\ )$
Uf: $S=F(X)$

1’. Sorozatszámítás$\big(H^*,H,\mathcal{F}(H\times H,H)\big):H$ lehetne még **Sorozatszámítás**$\big(H^*,S,\mathcal{F}(H_2\times H,S)\big):H_2$

Be: $N\in\mathbb{N},\ X\in H^*,\ f:H\times H\to H,\ f_0\in H$
Ki: $S\in H$
Ef: $\mathrm{Hossz}(X)=N \wedge \forall h\in H: f(f_0,h)=h$
Uf: $S=F(X)$
Def: $F:H^*\to H$
$F((x_1,..,x_N))=f(F(x_1,..,x_{N-1}),x_N))\ \wedge\ F(())=f_0\ )$

2. Eldöntés$\big(H^*,\mathcal{F}(H,L)\big):L$

Be: $N\in\mathbb{N},\ X\in H^*,\ T:H\to L$
Ki: $VAN\in L$
Ef: $\mathrm{Hossz}(X)=N$
Uf: $VAN\equiv \exists i\in[1..N]:T(x_i)$

3. Kiválasztás$\big(H^*,\mathcal{F}(H,L)\big):N$

Be: $N\in\mathbb{N},\ X\in H^*,\ T:H\to L$
Ki: $SORSZ\in\mathbb{N}$
Ef: $\mathrm{Hossz}(X)=N \wedge \exists i\in[1..N]: T(x_i)$
Uf: $SORSZ\in[1..N]\ \wedge\ T(x_{SORSZ})$

4. (Lineáris) keresés$\big(H^*, \mathcal{F}(H,L)\big)$:$L \cup L \times \mathbb{N}$

Be: $N \in \mathbb{N}$, $X \in H^*$, $T: H \to L$
Ki: $VAN \in \mathbb{L}$, $SORSZ \in \mathbb{N}$
Ef: Hossz(X)=N
Uf: $VAN \equiv \exists i \in [1..N] : T(x_i) \;\wedge\; VAN \Rightarrow SORSZ \in [1..N]$ és $T(x_{SORSZ})$

5. Megszámolás$\big(H^*, \mathcal{F}(H,L)\big)$:$\mathbb{N}$

Be: $N \in \mathbb{N}$, $X \in H^*$, $T: H \to L$
Ki: $DB \in \mathbb{N}$
Ef: Hossz(X)=N
Uf: $DB = \sum_{\substack{i=1 \\ T(x_i)}}^{N} 1$

6a. Maximum-kiválasztás$\big(H^*, \mathcal{F}(H \times H,L)\big)$:$\mathbb{N}$ (index)

Be: $N \in \mathbb{N}$, $X \in H^*$, $\leq \in \mathcal{F}(H \times H,L)$
Ki: $MAXI \in \mathbb{N}$
Ef: Hossz(X)=N $\wedge$ N≥1 $\wedge$ Rendezés(≤) $\wedge$ $RendezettHalmaz_{\leq}(H)$
Uf: $MAXI \in [1..N] \;\wedge\; \forall i \in [1..N]: x_{MAXI} \geq x_i$

6b. Maximum-kiválasztás$\big(H^*, \mathcal{F}(H \times H,L)\big)$:H

Be: $N \in \mathbb{N}$, $X \in H^*$, $\leq \in \mathcal{F}(H \times H,L)$
Ki: $MAX \in H$
Ef: Hossz(X)=N $\wedge$ N≥1 $\wedge$ Rendezés(≤) $\wedge$ $RendezettHalmaz_{\leq}(H)$
Uf: $MAX \in X \;\wedge\; \forall i \in [1..N]: MAX \geq x_i$

7a. Kiválogatás$\big(H^*, \mathcal{F}(H,L)\big)$:$\mathbb{N}^*$ (index-sorozat)

Be: $N \in \mathbb{N}$, $X \in H^*$, $T: H \to L$
Ki: $DB \in \mathbb{N}$, $Y \in \mathbb{N}^*$
Ef: Hossz(X)=N
Uf_1: $DB = \sum_{\substack{i=1 \\ T(x_i)}}^{N} 1 \;\wedge\; Y \in [1..N]^{DB} \;\wedge\; (\, T(x_{y_i}) \;\; i \in [1..DB] \,) \wedge$ HalmazFölsorolás(Y)
Uf_2: $DB = \sum_{\substack{i=1 \\ T(x_i)}}^{N} 1 \;\wedge\; Y \in [1..N]^{DB} \;\wedge\; (\, T(x_{y_i}) \;\; i \in [1..DB] \,) \wedge Y \subseteq (1,2...,N)$

7b. Kiválogatás$\big(H^*, \mathcal{F}(H,L)\big)$:$H^*$ (értéksorozat)

Be: $N \in \mathbb{N}$, $X \in H^*$, $T: H \to L$
Ki: $DB \in \mathbb{N}$, $Y \in H^*$
Ef: Hossz(X)=N
Uf: $DB = \sum_{\substack{i=1 \\ T(x_i)}}^{N} 1 \;\wedge\; Y \in H^{DB} \;\wedge Y \subseteq X \;\wedge\; T(y_i) \;\; i \in [1..DB]$

8. BelsőRendezés$\big(H^*, \mathcal{F}(H \times H,L)\big)$:$H^*$

Be: $N \in \mathbb{N}$, $X \in H^*$, $\leq \in \mathcal{F}(H \times H,L)$
Ki: $X' \in H^*$ (X új értéke)
Ef: Hossz(X)=N $\wedge$ Rendezés(≤)
Uf: $X' \in \mathcal{P}(X) \;\wedge\;$ $RendezettSorozat_{\leq}(X')$

9. IndexesRendezés$\left(H^*, \mathcal{F}(H\times H,L)\right)$:$N^*$ (index-sorozat)

Be: $N\in\mathbb{N}$, $X\in H^*$, $\leq\in\mathcal{F}(H\times H,L)$

Ki: $F\in\mathbb{N}^*$ (felsoroló vektor)

Ef: Hossz(X)=N $\wedge$ Rendezés($\leq$)

Uf: X=X' $\wedge$ $F\in\mathcal{P}((1,2,\ldots,N))$ $\wedge$ RendezettSorozat$_{\leq}$(X°F)

(azaz $\forall i,j\in[1..N]$: $(i<j)$ $x_{f_i}\leq x_{f_j} \Rightarrow f_i\leq f_j$)

második változat:

Be: $N\in\mathbb{N}$, $X\in H^*$, $\leq\in\mathcal{F}(H\times H,L)$

Ki: $S\in\mathbb{N}^*$ (sorrend-vektor)

Ef: Hossz(X)=N $\wedge$ Rendezés($\leq$)

Uf: X=X' $\wedge$ $S\in\mathcal{P}((1,2,\ldots,N))$ $\wedge$ $\forall i,j\in[1..N]$: $(i<j)$ $x_i\leq x_j \Rightarrow s_i\leq s_j$

10. Szétválogatás$\left(H^*, \mathcal{F}_i(H,L)\ (i=1..K)\right)$:$(H^*)^K$

Be: $N\in\mathbb{N}$, $X\in H^*$, $K\in\mathbb{N}$, T_i:H→L (i=1..K)

Ki: $DB_i\in\mathbb{N}$, $Y_i\in H^*$ (i=1..K)

Ef: Hossz(X)=N $\wedge$ K≥2 $\wedge$ T_i (i=1..K) H partícionálása

Uf: $DB_i=\sum_{\substack{j=1\\T_i(x_j)}}^{N}1$ $\wedge$ $Y_i\in H^{DB_i}$ $\wedge$ $Y_i\subseteq X$ $\wedge$ $\forall j\in[1..DB_i]$: $T_i(y_{i_j})$ (i=1..K)

Megjegyzések:

1. „Alapvariációja”: indexsorozatokkal történő szétválogatás

Szétválogatás$\left(H^*, \mathcal{F}_i(H,L)\ (i=1..K)\right)$:$(N^*)^K$

2. Kétfelé szétválogatás: (T_1:=T, T_2:= nem-T)

Szétválogatás$\left(H^*, \mathcal{F}(H,L)\right)$:$(H^*)^2$

a. Külön, két outputsorozatba

b. Helyben: „elválasztó indextől” balra a T-, jobbra a nem-T-tulajdonságúakat

11. Metszet$\left(H^*, H^*\right)$:H^*

Be: $N\in\mathbb{N}$, $X\in H^*$, $M\in\mathbb{N}$, $Y\in H^*$

Ki: $DB\in\mathbb{N}$, $Z\in H^*$ [pontosabban $Z\in H^{DB}$, DB≤min(N,M)][3]

Ef: Hossz(X)=N $\wedge$ HalmazFölsorolás(X) $\wedge$ Hossz(Y)=M $\wedge$ HalmazFölsorolás(Y)

Uf: $DB=\sum_{\substack{i=1\\x_i\in Y}}^{N}1$ $\wedge$ $Z\in H^{DB}$ $\wedge$ HalmazFölsorolás(Z) $\wedge$

($z_i\in X$ $\wedge$ $z_i\in Y$ $i\in[1..DB]$)

12. Egyesítés$\left(H^*, H^*\right)$:H^*

Be: $N\in\mathbb{N}$, $X\in H^*$, $M\in\mathbb{N}$, $Y\in H^*$

Ki: $DB\in\mathbb{N}$, $Z\in H^*$ [pontosabban $Z\in H^{DB}$, DB≤N+M]

Ef: Hossz(X)=N $\wedge$ HalmazFölsorolás(X) $\wedge$ Hossz(Y)=M $\wedge$ HalmazFölsorolás(Y)

Uf: $DB=N+\sum_{\substack{j=1\\y_j\notin X}}^{M}1$ $\wedge$ $Z\in H^{DB}$ $\wedge$ HalmazFölsorolás(Z) $\wedge$

($z_i\in X$ $\vee$ $z_i\in Y$ $i\in[1..DB]$)

[3] Ez a megjegyzés a deklaráció szempontjából lehet fontos.

13. Összefuttatás$\left(H^*,H^*,\mathcal{F}(H\times H,L)\right)$:$H^*$

Be: $N\in\mathbb{N}$, $X\in H^*$, $M\in\mathbb{N}$, $Y\in H^*$, $\leq\in\mathcal{F}(HxH,L)$

Ki: $DB\in\mathbb{N}$, $Z\in H^*$ [pontosabban $Z\in H^{DB}$, $DB\leq N+M$]

Ef: Rendezés(≤) ∧ Hossz(X)=N ∧ HalmazFölsorolás(X) ∧
Hossz(Y)=M ∧ HalmazFölsorolás(Y) ∧
RendezettSorozat$_{\leq}$(X) ∧ RendezettSorozat$_{\leq}$(Y)

Uf: $DB = N + \sum_{\substack{j=1 \\ y_j\notin X}}^{M} 1$ ∧ $Z\in H^{DB}$ ∧ HalmazFölsorolás(Z) ∧ RendezettSorozat$_{\leq}$(Z) ∧
($z_i\in X \vee z_i\in Y$ $i\in[1..DB]$)

Keresések

Új fogalmak:

$L\in H^*$ *lehetségesek sorozata*	– az eredeti X egy alkalmasan választott rész-sorozata: $L\subseteq X$
$\lVert.\rVert$: $H^*\to\mathbb{N}$ *hossz*	– sorozat hossza (elemszáma)
σ: $H^*\to H^*$ *szűkítés*	– $\forall L\in H^*$: $\sigma(L)=L_1$ $\lVert L\rVert>\lVert L_1\rVert$
ε: $H^*\to\mathbb{N}$ *elemkiválasztás*	– $\forall L\in H^*$: $\varepsilon(L)\in[1..L]$

Ezen fogalmakkal az *általános keresés tétel megfogalmazása*:

```
L:=X
Ciklus amíg ‖L‖>0 és nem T(X(ε(L)))
  L:=σ(L)
Ciklus vége
Van:=‖L‖>0
Ha Van akkor Sorsz:=ε(L)
```

Nézzük a *lineáris keresés tétel újrafogalmazás*át!

L:={x_i: $i\in$[e..N]} (elejétől végig, e: ahol tartunk=1..N+1)

L azonosítható, sőt helyettesíthető is az első elemének indexével: L:=L(e)→e

ε(L(e)):=x_e (első elem)

σ(L(e)):= {x_i $i\in$[e+1..N]} ⊆ {x_i $i\in$[e..N]}

‖L(e)‖:=N-e+1

```
L:=X        [i:=1, 1.-nél kezdődik a ... sorozat]
Ciklus amíg ‖L‖>0  [i≤N, azaz még van hol keresni]  és
            nem T(X(ε(L)))  [T(X(i)), azaz az i. olyan-e]
  L:=σ(L(ε(L)))  [i:=i+1, a következőnél kezdődik a ... sorozat]
Ciklus vége
Van:=‖L‖>0  [i≤N, azaz lenne még hol keresni, de megvan]
Ha Van akkor SORSZ:=ε(L).sorsz  [SORSZ:=e]
```

A közismert keresésekben az alábbi értelmezésű függvényeket szoktuk használni:

1. $\varepsilon_1((x_i,...,x_j)):=i$ (első)
2. $\varepsilon_u((x_i,...,x_j)):=j$ (utolsó)
3. $\varepsilon_k((x_i,...,x_j)):=(i+j)\ DIV\ 2$ (középső)
4. $\sigma_1((x_i,...,x_j)):=(x_{i+1},...,x_j)$ (első elem nélkül)
5. $\sigma_u((x_i,...,x_j)):=(x_i,...,x_{j-1})$ (utolsó elem nélkül)

6'. $\sigma_{k<}((x_i,...,x_j)):=(x_l,...,x_m)\subseteq(x_i,...,x_j)$, ahol $l=i$ és $m=(i+j\ DIV\ 2)-1$

6''. $\sigma_{k>}((x_i,...,x_j)):=(x_l,...,x_m)\subseteq(x_i,...,x_j)$, ahol $l=(i+j\ DIV\ 2)+1$ és $m=j$

7. $\sigma_<((x_i,...,x_j)):=(x'_l,...,x'_k)\subseteq(x_i,...,x_j)$ $\forall x'_m<x_i$ $m\in[l..k]$ (elsőnél kisebbek)

14. LineárisKeresésRendezetben$\big(H^*, H, \mathcal{F}(H\times H, L)\big)$:$L\cup L\times N$

Be: $N\in\mathbb{N}$, $X\in H^*$, $y\in H$, T_y:$H\to L$ ($T_y(h)\equiv h=y\ \forall h\in H$), $\leq\in \mathcal{F}(H\times H,L)$
Ki: $VAN\in L$, $SORSZ\in\mathbb{N}$
Ef: Hossz(X)=N $\wedge$ Rendezés($\leq$) $\wedge$ RendezettSorozat$_{\leq}$(X)
Uf: $VAN \equiv \exists i\in[1..N] : x_i=y\ \wedge\ VAN \Rightarrow SORSZ\in[1..N] \wedge x_{SORSZ}=y$
Absztrakt algoritmus:
$L:=\{x_i: i\in[e..N]\}$ (elejétől végig, e: ahol tartunk=1..N+1)

Az e a sorozat elejét jelöli, amelyben még lehet a keresett. Figyelem: az L sorozat még annak előtte, hogy az N. eleméhez érnénk, véget érhet! (Ha x>y bekövetkezett.)

L azonosítható, sőt helyettesíthető is az első elemének indexével: $L:=L(e)\to e$

$\varepsilon(L(e)):=\varepsilon_1(L(e)):=e$ (a mindenkori első elem)
$\sigma(L(e)):=\{x_i: i\in[e+1..N]\}$, ha $x_e\leq y$ (1)
$\sigma(L(e)):=\{x_i: i\in[N+1..N]\}$, ha $x_e>y$ (2)
(1) és (2) $\subseteq \{x_i: i\in[e..N]\}$
$L(e):=N-e+1$ ($L(e)>0 \Leftrightarrow N-e+1>0 \Leftrightarrow N\geq e$)

```
e:=1
Ciklus amíg N≥e  és  X(e)≠y
  Elágazás
    X(e)<y esetén  e:=e+1
    X(e)>y esetén  e:=N+1
  Elágazás vége
Ciklus vége
Van:=N≥e
Ha Van akkor SORSZ:=e
```

Vegyük észre:
1. a ciklusból kilépünk, ha 'nem N≥e vagy X(e)=y' teljesül;

N<e $\Leftrightarrow$ találunk nagyobbat: X(e)>y (l. σ definícióját)
vagy egyszerűen a sorozat végére érünk: N+1=e

2. Így a ciklusfeltételre következő igaz:

N≥e és X(e)≠y nem (X(e)>y vagy N+1=e) és X(e)≠y $\Leftrightarrow$
X(e)≤y és N+1>e és X(e)≠y $\Leftrightarrow$ N+1>e és X(e)<y $\Leftrightarrow$ N≥e és X(e)<y

3. Ezen átalakítás után a ciklusmagban fölöslegessé vált az elágazás, hiszen az első ág triviálisan igaz, a másik meg sohasem igaz.
4. Ezzel viszont a kilépés okát már nem jelzi egyértelműen az 'N≥e' feltétel (hiszen az L sorozat üresre állítását nem végzi el senki): egybemosódott a sikeres és az értelmetlen keresést meggátoló kilépés. A sikerességet jelenti az 'N≥e és X(e)=y' feltétel teljesülése. Vagyis:

```
e:=1
Ciklus amíg N≥e  és  X(e)<y
  e:=e+1
Ciklus vége
Van:=N≥e és X(e)=y
Ha Van akkor SORSZ:=e
```

5. Egyszerűsödik az algoritmus, ha sikerül egyszerűsíteni e szétválás érzékelése. A trükk mindössze annyi, hogy az utolsó elem előtt kilépünk, s ennek vizsgálatára redukáljuk a problémát:

```
e:=1
Ciklus amíg N>e  és  X(e)<y
  e:=e+1
Ciklus vége
Van:=X(e)=y
Ha Van akkor SORSZ:=e
```

15. LogaritmikusKeresés$\big(H^*,H,\mathcal{F}(H\times H,L)\big)$:$L\cup L\times\mathbb{N}$

Be: $N\in\mathbb{N}$, $X\in H^*$, $y\in H$, $T_y:H\to L$ ($T_y(h)\equiv h=y$ $\forall h\in H$), $\le\in\mathcal{F}(H\times H,L)$

Ki: $VAN\in L$, $SORSZ\in\mathbb{N}$

Ef: Hossz(X)=N $\wedge$ Rendezés(≤) és RendezettSorozat$_{\le}$(X) $\wedge$
$\exists$elem:$H^N\times\mathbb{N}\to H$ szelekciós függvény: elem(X,i)=x_i

Uf: $VAN\equiv\exists i\in[1..N]: x_i=y$ $\wedge$ $VAN\Rightarrow SORSZ\in[1..N]\wedge x_{SORSZ}=y$

Absztrakt algoritmus:

L:={x_i: $i\in$[e..v]} (a mindenkori eleje, vége, kezdetben: (1,N))

L azonosítható, sőt helyettesíthető is az első és az utolsó elemének indexével:
L:=L(e,v) $\to$ (e,v)

ε(L(e,v)):=ε_k(L(e,v)):=(e+v) DIV 2 (középső elem)

σ(L(e,v)):=$\sigma_{k\le}$(L(e,v)), ha $x_{L(e,v)}$<y (1)

σ(L(e,v)):=$\sigma_{k\ge}$(L(e,v)), ha $x_{L(e,v)}$>y (2)

$\|L(e,v)\|$:=v-e+1 ($\|L(e,v)\|>0 \Leftrightarrow v-e+1>0 \Leftrightarrow v\ge e$)

```
(e,v):=(1,N)
Ciklus amíg v≥e és X((e+v) DIV 2)≠y
  Elágazás
    X((e+v) DIV 2))>y esetén (e,v):=(e,((e+v) DIV 2)-1)
    X((e+v) DIV 2))<y esetén (e,v):=(((e+v) DIV 2)+1,v)
  Elágazás vége
Ciklus vége
Van:=v≥e
Ha Van akkor SORSZ:=(e+v) DIV 2
```

Az egyszerűbb leírhatóság érdekében tároljuk az ε(L(e,v))-dik (=(e+v) DIV 2) elem indexrészét a k változóban, e leválasztás viszont a külön kiszámolást jelenti! Másrészt az 'X(k)≠y' feltétel kétirányúra egyszerűsíti a ciklusmagot. Így az alábbi algoritmus marad:

```
(e,v):=(1,N); k:=(e+v) DIV 2
Ciklus amíg v≥e és X(k)≠y
  Ha X(k)>y akkor  v:=k-1
          különben e:=k+1
  k:=(e+v) DIV 2
Ciklus vége
Van:=v≥e
Ha Van akkor SORSZ:=k
```

16. VisszalépésesKeresés$\big((H_1^*\times H_2^*\times...),\mathcal{F}(H_1\times H_2\times...,L)\big)$:$L\cup L\times\mathbb{N}^*$

Be: $K\in\mathbb{N}$, $M\in\mathbb{N}^*$ (*K hosszú méretsorozat*),
$Y_i\in H_i^*$ ($\forall i\in[1..K]$) (*M_i elemszámú lehetőségsorozatok*),
l: $\bigcup_{i=1}^{K}\times_{j=1}^{i} H_j\to L$ (*lehetőségek összeférése*),

Ki: $VAN\in L$, $X\in\mathbb{N}^*$ (*az összeférő lehetőségek indexeinek sorozata*)

Ef: $K\ge2$ $\wedge$ Hossz(M)=K $\wedge$
HalmazFölsorolás(Y_i) $\forall i\in[1..K]$ $\wedge$
$\forall j\in[1..K]$, $\forall i\in[1..j]$ $l(h_1,...,h_j)\Rightarrow l(h_1,...,h_i)$

Uf: $VAN\equiv\exists Z\in Y_1\times Y_2\times...\times Y_K$: l(Z) $\wedge$
$VAN\Rightarrow y^{(i)}_{x_i}\in Y_i$ és $l(y^{(1)}_{x_1},...,y^{(k)}_{x_k})$

Absztrakt algoritmus:

L: A keresett megoldás a $\times_{i=1}^{K} Y_i$ K-dimenziós tér egy pontja, amely kielégíti az l-t. Ennek megadását az X-tömb végzi az Y_i-beli komponensek *indexei*nek felsorolásával. A megoldás keresése során e teret kell

ügyesen bejárnunk, lehetőleg úgy, hogy ne kelljen mind az M_1*...*M_K „pontjának” vizsgálatát elvégeznünk. Ezt úgy tesszük, hogy szisztematikusan szűkítjük a teret egyre szűkülő alterek uniójára, azáltal, hogy újabb és újabb komponensét kötjük meg a leendő megoldásnak. Kezdetben csak az 1. dimenzióban lesz kötött érték. Ekkor tehát a térnek azt a K-1 dimenziós alterét szemeltük ki további vizsgálatra, amelynek 1. komponense éppen a rögzített érték. A vizsgálatra szoruló pontok halmaza még a teljes tér. Amikor kiderül a komponensek konkretizálása során, hogy a „koncentrált” altér nem tartalmazhatja a megoldást, akkor a teljes alteret elvetve vesszük a „szomszédos” (valamelyik soron következő, azaz a megfelelő Y_i-beli későbbi elem által meghatározott) alteret.
Ezek után az L a következő indexsorozatok *halmaz*a: az (1,1, ..., 1), első elemek sorozatától az $(m_1,m_2,...,m_K)$ utolsó sorozatig „tart”. (Az $m_i=\|Y_i\|$.) (Formális okok miatt egészítsük ki a fenti teret olyan elemekkel is, amelyek bármely koordinátája a 0 is lehet.) Ezt a halmazt tesszük pontok *sorozat*ává az ε és σ függvények segítségével.
A gondolatmenetből nyilvánvaló, hogy a pontfelsorolás alapja az L-beli sorozatok *kezdőszelet*ei lesznek. Az aktuális L azonosítására mód van a „kiinduló” sorozat nem triviális (azaz rögzített) elemeket tartalmazó kezdőszelete által:

$L:=L(x_1,x_2,...x_i)$ (az x_i-k az *indexek*et jelentik),

elvárjuk, hogy

$L(x_1,x_2,...x_i)=L(x_1,x_2,...x_i,0,...,0)$.

Vegyük észre az L alábbi tulajdonságait!
1. Az összes lehetőség L halmaza: L(1,1,...,1).
2. Monoton csökkenés:

a. $L(x_1,...,x_i)=L(x_1,...,x_{i+1})\quad \forall x_{i+1}\in[1..m_{i+1}]$,
azaz „előrehaladáskor” nem csökken a vizsgálatban szereplők száma;

b. $L(x_1,...,x_i)\supset L(x_1,...,x'_{i-1})\quad \forall x'_{i-1}\in(x_{i-1}..m_{i-1}]$.
azaz „visszalépéskor” éppen annyival csökken, ahány „dimenziós” alteret sikerült kizárni a további vizsgálatból, pontosabban éppen az

$\{(x_1,...,x_{i-1},\mathbf{x'_i,...,x'_K}) : \forall j\in(i..K]\ \forall \mathbf{x'_j}\in[1..m_j]\}$

alteret, ami elemszáma $\prod_{j=i}^{K} m_j$;

c. $L(x_1,...,x_i)\supset L(x_1,...,x'_i)\quad \forall x'\in(x_i..m_i]$,
azaz többszörös visszalépés után az „átlépett” alterekbeli elemekkel csökkenthető a vizsgálat;

d. a b. és a c.-ből következik, hogy a „legkisebb” $L(x_1,...,x_i)$ az L(), amely az $L(x_{m_1})$-ből való visszalépés után adódik. Ebben 0 darab elem van.

Az „L-mérték” nyomon követését végzi a $\|.\|$ *függvény*, így elkerülhetetlen ennek matematikai megragadása. Nyilvánvaló lehetőség lenne az előbbiekre építve a még ki nem szórt pontok számával mérni. Ennél lényegesen egyszerűbb mód is van. A d. figyelembe vételével az L üressé válását észlelhetjük az éppen a lerögzített elemek számának figyelésével: amikor az 0-ra csökken, akkor ürült ki a tér is.

Az ε értelmezése következik:

$$\varepsilon\big(L(x_1,...,x_i)\big):=(x_1,...x_i,0,...,0)$$

Nézzük a σ meghatározását!

$$\sigma\big(L(x_1,...,x_i)\big):=\begin{cases} L(x_1,...,x_i,x_{i+1}) & ,i\le K\wedge l(x_1,...,x_i,x_i) \\ \sigma(L(x_1,...,x'_{i-1},0)) & ,i\ge 1\wedge\neg l(x_1,...,x_i,x_i) \\ & \wedge\exists^* j\in[1..i):\exists_* z_j\in(x_j..m_j]:l(x_1,...,z_j) \end{cases}$$

Magyarázatok, jelölések, megállapodások és megállapítások:

1. $\exists^* x'_v$ jelentése: $\exists x'_v\in(x_v..m_v]$ és az *első olyan* ..., illetve ha $\neg\exists x'_v$ korábbi értéke (azaz $x'_v=0$), akkor $x'_v\in[1..m_v]$ és *olyan*.
2. Természetesen a második ágra csak az első ág feltételének nem teljesülése esetén kerülhet sor.
3. Az i jelölje a vizsgált megoldássorozat nem triviális kezdőszeletének hosszát! Ekkor
$L>0 \Leftrightarrow i\ge 1$.
4. A keresett elemig akkor jutunk, amikor $i>K$.
5. Nincs meg a keresett elem (azaz L=**0**), ha $i<1$.
6. (Mint nyilvánvaló:) az L-t az *i* és az X *együtt* határozzák meg!
Így pl. $L:=X \Leftrightarrow i:=1: X:=0$

Az algoritmus:

```
Típus Keresett=Rekord(van:Logikai,melyik:Egész)

i:=1 [i. kezdőszeletnél tartunk]
X(1..N):=0 [még nincs rögzített]
Ciklus amíg i≥1 és i≤K
  ker:=JóElem(i)  [az i.-ig lehetséges-e a kiválasztás?]
  Elágazás
   I≤K és ker.van      esetén X(i):=ker.melyik; i:+1
         i≥1           esetén X(i):=0; i:-1
  Elágazás vége
Ciklus vége
Van:=i≥1
[X-ben található a megoldás, ha létezik]
```

Mivel a ciklusban csak i≥1 és i≤K esetben vagyunk, s csak ekkor történhet az i növelése, vagy csökkentése, ezért a kétirányú elágazás helyett írható Ha-típusú elágazás is, egyszerűbb feltétellel. Így kapjuk:

```
i:=1 [i. kezdőszeletnél tartunk]
X(1..N):=0 [még nincs rögzített]
Ciklus amíg i≥1 és i≤K
  ker:=JóElem(i)  [az i.-ig lehetséges-e a kiválasztás?]
  Ha ker.van akkor   X(i):=ker.melyik; i:+1
          különben  X(i):=0; i:-1
Ciklus vége
Van:=i≥1
[X-ben található a megoldás, ha létezik]
```

```
Függvény JóElem(i:Egész): Keresett
                        [i.-ig nem triviális sorozat megoldás-e?]
  j:=X(i)+1
  Ciklus amíg j≤M(i) és nem Lehetséges(i,j)
    j:+1
  Ciklus vége
  JóElem.van:=j≤M(i)
  Ha JóElem.van akkor JóElem.melyik:=j
Függvény vége.

Függvény Lehetséges(i,j:Egész): Logikai
                        [i. komponens összefér-e az eddigiekkel?]
  Lehetséges:=l(Y(1)(X(1)),...,Y(i-1)(X(i-1)), Y(i)(j))
Függvény vége.
```

A visszalépéses keresés (backtrack) elvének beépítése más tételekbe

Mindössze annyi az észreveendő, hogy

1. L>0	i≥1
2. T(ε(L))	i≤K
3. σ(L,ε(L))	keresett:=JóElem(i)

```
Ha  keresett.van  akkor  X(i):=keresett.melyik;  i:+1
                  különben X(i):=0;  i:-1
```

Így pl. a megszámolás tétel variánsa a következő lesz. Annyi előzetes átalakításra szükség van, hogy a megszámolás tétel ’számlálásos ciklusát’ ’amíg’-osra kell átírni.

```
Db:=0
i:=1;  X(1..K):=0
```

```
Ciklus amíg i≥1
  Ha  i>K  akkor  Db:+1;  i:-1
                      [visszalép a K. komponenshez]
  keresett:=JóElem(i)
  Ha  keresett.van  akkor  X(i):=keresett.melyik;  i:+1
                 különben X(i):=0;  i:-1
Ciklus vége
```

... vagy a maximum-kiválasztás ... (Itt feltételezzük, hogy ∃ ≤ rendezés az L téren, ami X megoldásokat összevethetővé tesz. Az egyszerűség kedvéért föltesszük, hogy az L legkisebb eleme az (0,...,0).)

```
i:=1; X(1..K):=0
Max:=X  [az eddigi „legnagyobb”, „legjobb” megoldás]
Ciklus amíg i≥1
  Ha i>K akkor
    Ha Max<X akkor  Max:=X
    i:-1  [visszalép a K. komponenshez]
  Elágazás vége
  keresett:=JóElem(i)
  Ha  keresett.van  akkor  X(i):=keresett.melyik: i:+1
                 különben X(i):=0 : i:-1
Ciklus vége
Van:=i≥1
[Max-ban található a megoldás, ha létezik]
```