



ELTE | IK

PROGRAMOZÁS

Gyakori programozási minta változatok

Horváth Győző, Pluhár Zsuzsa

Ismétlés



Feladatmegoldás lépései

1. Specifikáció

- a) Példa
- b) Bemenet, kimenet
 - i. egyszerű adat?
 - ii. több különböző? – rekord
 - iii. több azonos? – **tömb**
- c) Előfeltétel
- d) Utófeltétel

2. Algoritmus

- a) Adat \rightarrow változók
- b) Új halmazok \rightarrow típusok
- c) Beolvasás
- d) Feldolgozás
 - i. támpontok az uf-ben
 - ii. végrehajtható spec.
 - iii. és, vagy, \rightarrow , \forall , \exists
 - iv. **nevezetes minták**
- e) Kiírás

3. Kód

Megfeleltetések

Példa adat	Specifikáció halmaz	Algoritmus típus	Kód type
3	N	Egész	int
-3	Z	Egész	int
3,3	R	Valós	double
igaz	L	Logikai	bool
"alma"	S	Szöveg	string
"a"	K	Karakter	char
{név:"Győző", jegy: 5}	név:S x jegy:N	Rekord	struct
[3, 5, -6, 2]	Z[1..n]	Tömb	int[]

Analóg programozás – visszavezetés

- Visszavezetés
 - Konkrét feladat felírása
 - Összevetés a minta sablonjával
 - Különbségek felírása egy táblázatba
 - Különbségek alkalmazása a sablon algoritmusában
 - → Konkrét feladat algoritmus



Programozási minták

1. Összegzés
2. Megszámolás
3. Maximumkiválasztás
4. Feltételes maximumkeresés
5. Keresés
6. Eldöntés
7. Kiválasztás
8. Másolás
9. Kiválogatás

Most Common DUPLO Parts



Brick Architect

szummás, mindenes feladat



számlálós ciklus

létezikes feladat



feltételes ciklus

szummás, mindenes feladat



számlálós ciklus



Minimumkiválasztás



Minimumkiválasztás

példa – specifikáció

minimumkiválasztás

Feladat:

Adjuk meg egy adott $f:[e..u] \rightarrow H$ függvény értékei között a **legkisebb** elemet!

Specifikáció:

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

Ki: $\text{minind} \in \mathbb{Z}, \text{minért} \in H$

Ef: $e \leq u$

Uf: $\text{minind} \in [e..u]$ és

$\forall i \in [e..u]: (f(\text{minind}) \leq f(i))$ és
 $\text{minért} = f(\text{minind})$

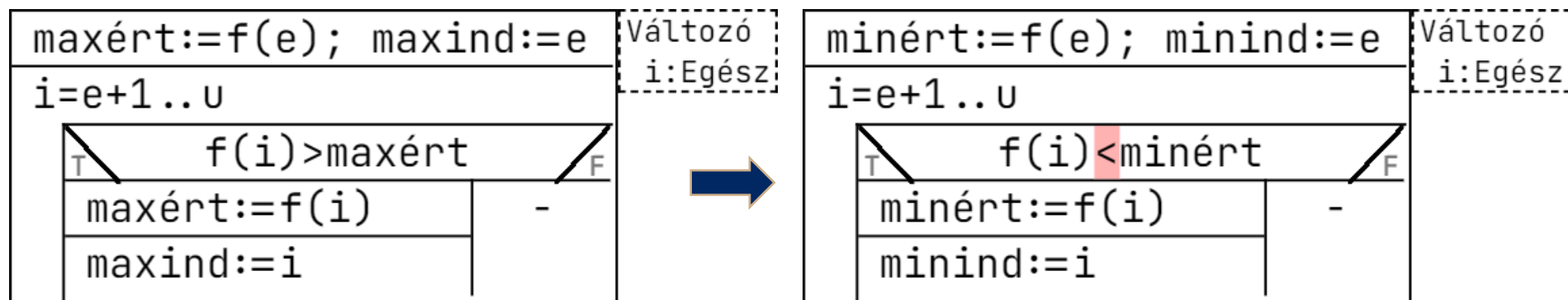
Uf: $(\text{minind}, \text{minért}) = \text{MIN}(i=e..u, f(i))$

egyetlen különbség a
maximumkiválasztáshoz
képest az elnevezések
mellett

Minimumkiválasztás algorithmus – analóg algoritmikus gondolkodással

Algoritmus:

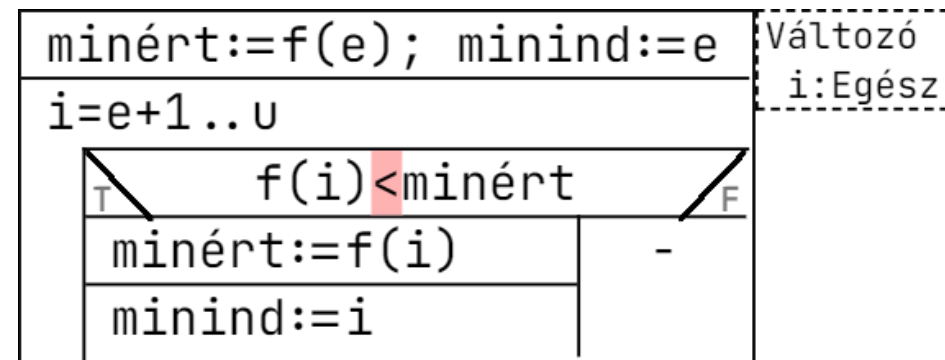
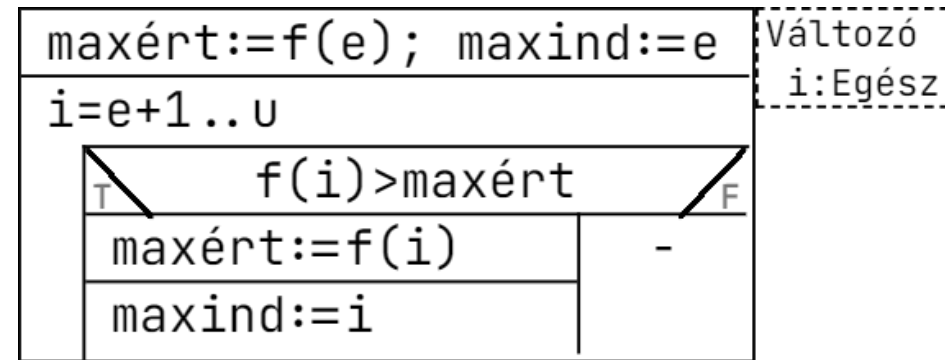
másik reláció használata



Minimumkiválasztás algoritmus – visszavezetéssel

Algoritmus:

- \leq „felüldefiniálása”
- eddig csak azt használtuk ki, hogy ez teljes rendezési reláció
- jelentése nem volt érdekes
- helyettesíthető bármilyen teljes rendezési relációval
- azaz $\leq(a,b)=a\geq b$



$a \leq b$ felírható függvényszerűen: $\leq(a,b)$
Az operátor jelentését definiálhatjuk felül

Minimumkiválasztás algoritmus – visszavezetéssel

Algoritmus:

- ötlet: alakítsuk át a feladatot!
- a függvényértékek **ellentettjei** között keressünk maximumot!
- a megtalált maximum a keresett minimum **ellentettje** lesz

Maximumkiválasztás

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

Ki: $\text{maxind} \in \mathbb{Z}, \text{maxért} \in \mathbb{H}$

Ef: $e \leq u$

Uf: $(\text{maxind}, \text{maxért}) =$
 $\text{MAX}(i=e..u, f(i))$

Visszavezetés:

$\text{maxind}, \text{maxért}$	\sim	$\text{minind}, -\text{minért}$
$f(i)$	\sim	$-f(i)$

Minimumkiválasztás

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

Ki: $\text{minind} \in \mathbb{Z}, \text{minért} \in \mathbb{H}$

Ef: $e \leq u$

Uf: $(\text{minind}, -\text{minért}) =$
 $\text{MAX}(i=e..u, -f(i))$

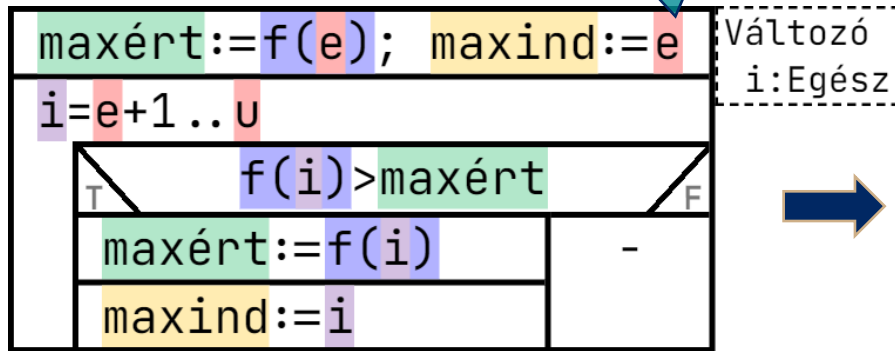
ötlet!

Minimumkiválasztás algoritmus – visszavezetéssel

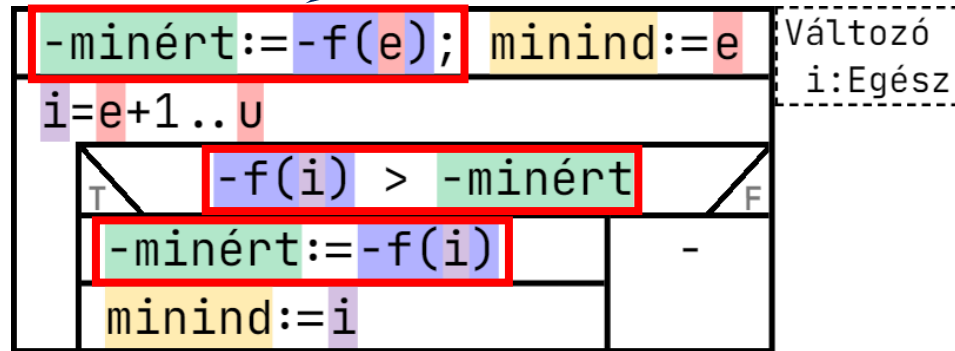
$\text{maxind}, \text{maxért} \sim \text{minind}, -\text{minért}$
 $f(i) \sim -f(i)$

maximumkiválasztás sablonja

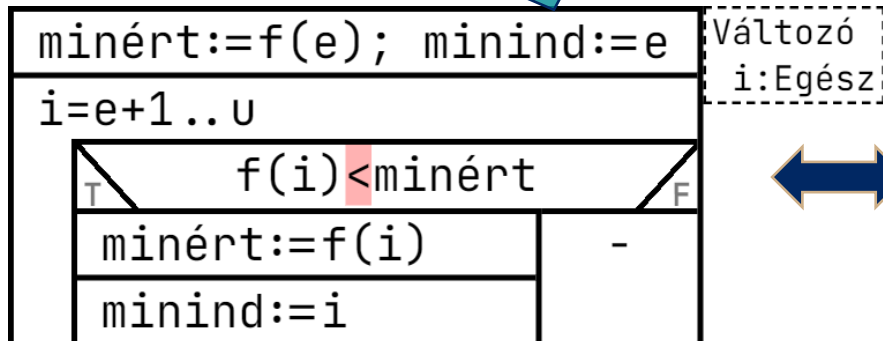
Algoritmus:



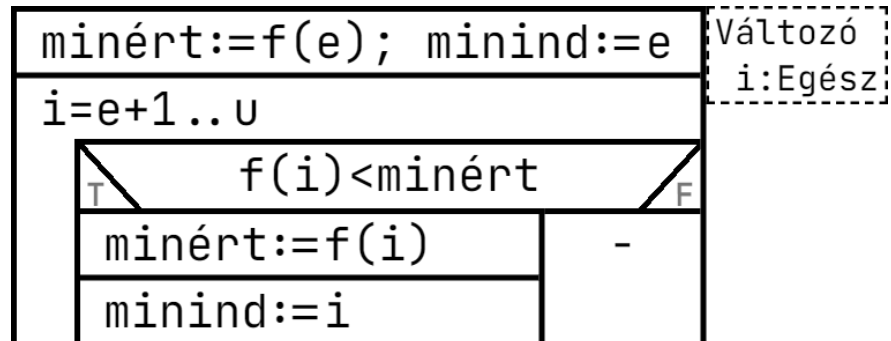
nem megengedett értékadás



algoritmikus gondolkodással



szorozzunk -1-gyel a szükséges helyeken!



Minimumkiválasztás sablon

Feladat

Adott az egész számok egy $[e..u]$ intervalluma és egy $f:[e..u] \rightarrow H$ függvény. A H halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, hogy az f függvény **hol** veszi fel az $[e..u]$ nem üres intervallumon a legkisebb értéket, és mondjuk meg, **mekkora** ez a minimális érték!

Specifikáció

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

Ki: $\text{minind} \in \mathbb{Z}$, $\text{minért} \in H$

Ef: $e \leq u$

Uf: $\text{minind} \in [e..u]$ és

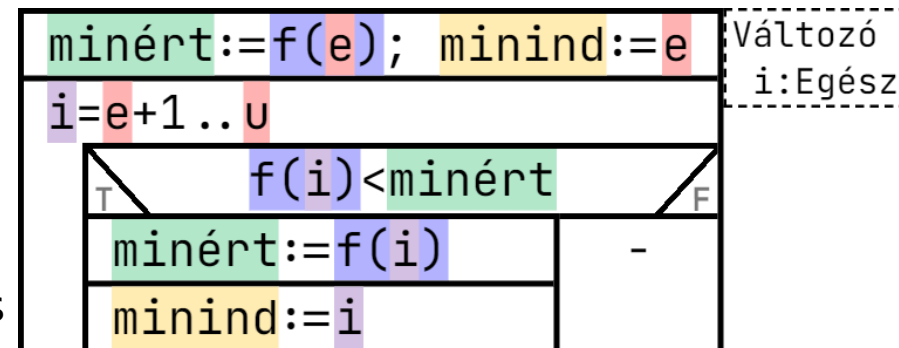
$\forall i \in [e..u]: (f(\text{minind}) \leq f(i))$ és

$\text{minért} = f(\text{minind})$

Rövidítve:

Uf: $(\text{minind}, \text{minért}) = \text{MIN}(i = e..u, f(i))$

Algoritmus



Hátulról keresés



Hátulról keresés példa

Feladat:

Keressük meg a legutolsó T tulajdonságú elemet!

Specifikáció:

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

Ki: $van \in \mathbb{L}$, $ind \in \mathbb{Z}$

Ef: -

Uf: $van = \exists i \in [e..u] : (T(i))$ és
 $van \rightarrow (ind \in [e..u] \text{ és } T(ind) \text{ és } \forall i \in [ind+1..u] : (\text{nem } T(i)))$

vö. az előlről keresésnél:
 $[e..ind-1]$

Hátulról keresés

algoritmus – algoritmikus gondolkodással

Algoritmus:

induljunk hátulról, lépegezzünk előre, amíg az intervallumon belül vagyunk és nem találtunk T tulajdonságú elemet

$ind := e$
$ind \leq u$ és nem $T(ind)$
$ind := ind + 1$
$van := ind \leq u$

$ind := u$
$ind \geq e$ és nem $T(ind)$
$ind := ind - 1$
$van := ind \geq e$

Hátulról keresés algoritmus – visszavezetéssel

Algoritmus:

- ötlet: alakítsuk át a feladatot!
- az $[e..u]$ intervallumot tükrözzük az origóra, és az így kapott **$[-u..-e]$ intervallumban keressük** az elsőt adott tulajdonságú elemet
- a talált ind helyett a **$-ind$ lesz az eredmény**

Előlről keresés sablonja

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

Ki: $van \in L, ind \in \mathbb{Z}$

Ef: -

Uf: $(van, ind) = \text{KERES}(i=e..u, T(i))$

Hátulról keresés

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

Ki: $van \in L, ind \in \mathbb{Z}$

Ef: -

Uf: $(van, -ind) = \text{KERES}(i=-u..-e, T(-i))$

ötlet!

Visszavezetés:

ind	\sim	$-ind$
$e..u$	\sim	$-u..-e$
$T(i)$	\sim	$T(-i)$

Hátulról keresés algoritmus – visszavezetéssel

keresés sablonja

```
ind := e
ind ≤ u és nem T(ind)
  ind := ind + 1
van := ind ≤ u
```

nem megengedett értékadás

$ind \sim -ind$
 $e..u \sim -u..-e$
 $T(i) \sim T(-i)$

algoritmikus gondolkodással

```
ind := u
ind ≥ e és nem T(ind)
  ind := ind - 1
van := ind ≥ e
```

$e..u \sim -u..-e$

```
ind := -u
ind ≤ -e és nem T(ind)
  ind := ind + 1
van := ind ≤ -e
```

$ind \sim -ind$
 $T(i) \sim T(-i)$

```
-ind := -u
-ind ≤ -e és nem T(ind)
  -ind := -ind + 1
van := -ind ≤ -e
```

szorozzunk -1-gyel a szükséges helyeken!

```
ind := u
ind ≥ e és nem T(ind)
  ind := ind - 1
van := ind ≥ e
```

Mind eldöntés



Mind eldöntés példa

Feladat:

Adjuk meg, hogy egy $[e..u]$ intervallum minden eleme rendelkezik-e egy adott T tulajdonsággal!

Specifikáció:

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

Ki: $\text{mind} \in \mathbb{L}$

Ef: -

Uf: $\text{mind} = \forall i \in [e..u] : (T(i))$

Rövidítve:

Uf: $\text{mind} = \mathbf{MIND}(i=e..u, T(i))$



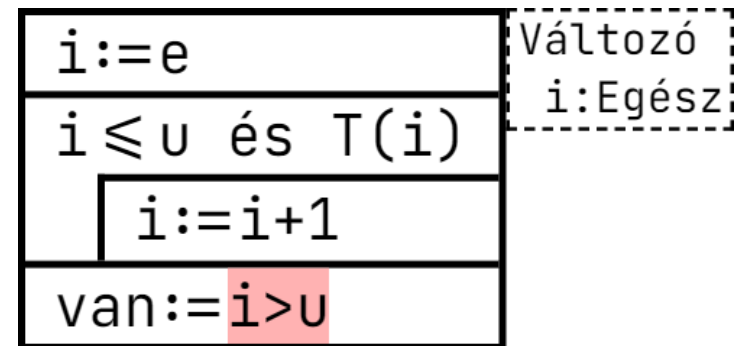
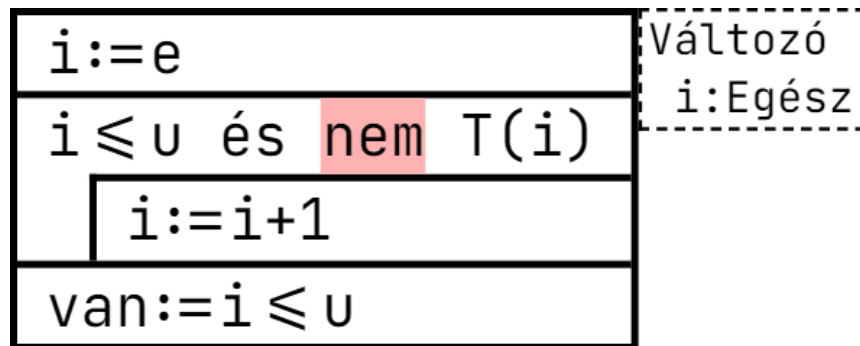
eldöntésben: $\exists i \in [e..u] : (T(i))$

Mind eldöntés

algoritmus – algoritmikus gondolkodással

Algoritmus:

Addig lépkedjünk előre, amíg jót találunk. Ha lelépünk az intervallumról, akkor mindegyik jó volt.



Mind eldöntés algorithmus – visszavezetéssel

Algorithmus:

- ötlet: alakítsuk át a feladatot!
- **van-e** olyan elem, ami **nem rendelkezik a T tulajdonsággal**?
- a **válasz negáltja** lesz a keresett eredmény

Van-e olyan?

Be: $e \in Z, u \in Z$

Ki: $\text{van} \in L$

Ef: -

Uf: $\text{van} = \text{VAN}(i=e..u, T(i))$

Visszavezetés:

van	\sim	nem mind
$T(i)$	\sim	$\text{nem } T(i)$

Mind olyan-e?

Be: $e \in Z, u \in Z$

Ki: $\text{mind} \in L$

Ef: -

Uf: $\text{mind} = \text{VAN}(i=e..u, \text{nem } T(i))$

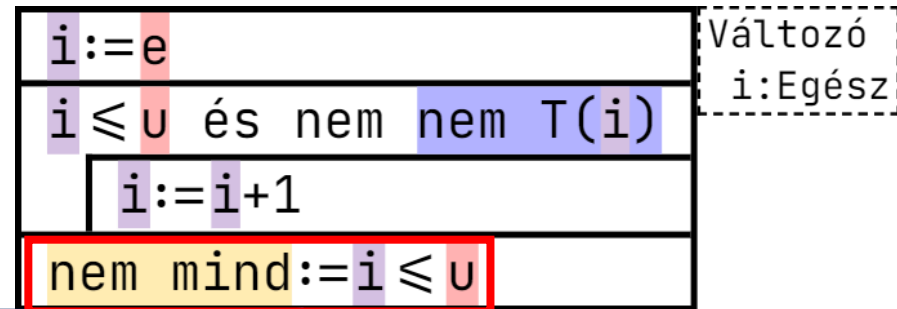
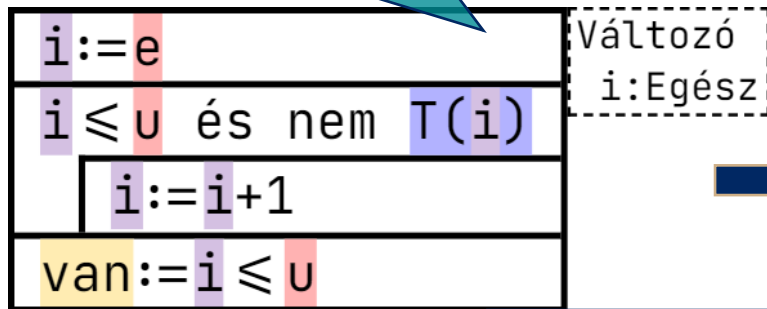
Uf: $\text{nem mind} = \text{VAN}(i=e..u, \text{nem } T(i))$

ötlet!

Mind eldöntés algoritmus – visszavezetéssel

van	~	nem mind
$T(i)$	~	nem $T(i)$

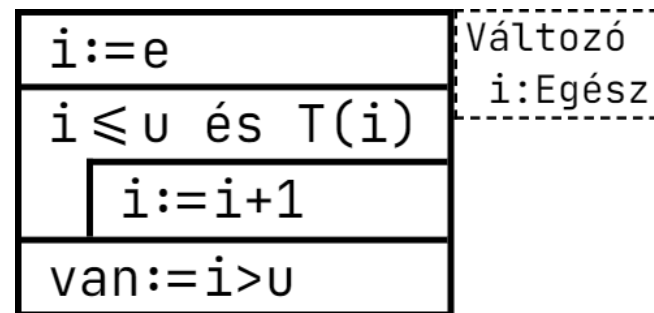
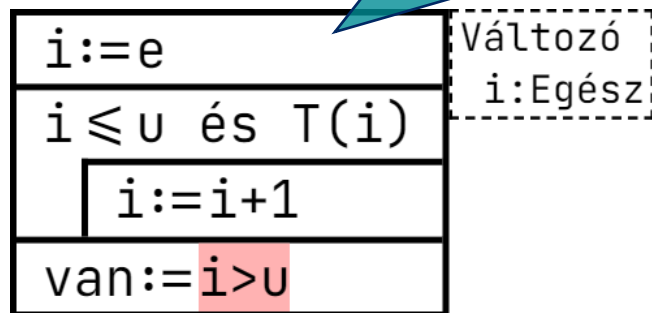
eldöntés sablonja



nem megengedett értékadás

negálunk a szükséges helyeken

algoritmikus gondolkodással



Mind eldöntés (vagy optimista eldöntés) sablon

Feladat

Adott az egész számok egy $[e..u]$ intervalluma és egy $T:[e..u] \rightarrow \text{Logikai feltétel}$. Határozzuk meg, hogy az $[e..u]$ intervallumnak **mindegyik** eleme olyan-e, amely kielégíti a T feltételt!

Specifikáció

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

Ki: $\text{mind} \in \mathbb{L}$

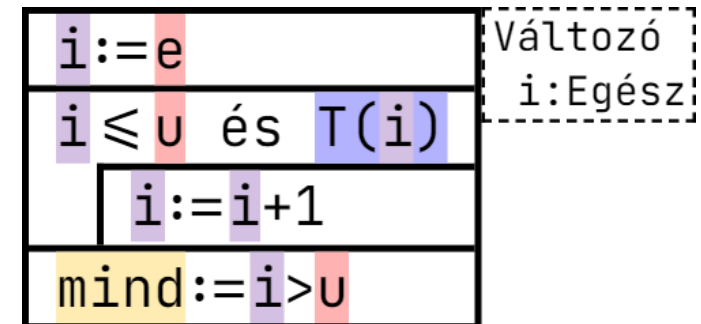
Ef: -

Uf: $\text{mind} = \forall i \in [e..u] : (T(i))$

Rövidítve:

Uf: $\text{mind} = \text{MIND}(i=e..u, T(i))$

Algoritmus



Kiegészítés: futóindex mint sablonrész



Példa – legmelegebb vasárnap

viSSzavezetés

Hétfői naptól kezdve mértük a déli hőmérsékletet.
Add meg a legmelegebb vasárnapot!

Feladatsablon

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

Ki: $\text{maxind} \in \mathbb{Z}$, $\text{maxért} \in \mathbb{H}$

Ef: $e \leq u$

Uf: $(\text{maxind}, \text{maxért}) =$
 $\text{MAX}(i = e..u, f(i))$

ViSSzavezetés:

Algoritmus:

Legmelegebb vasárnap

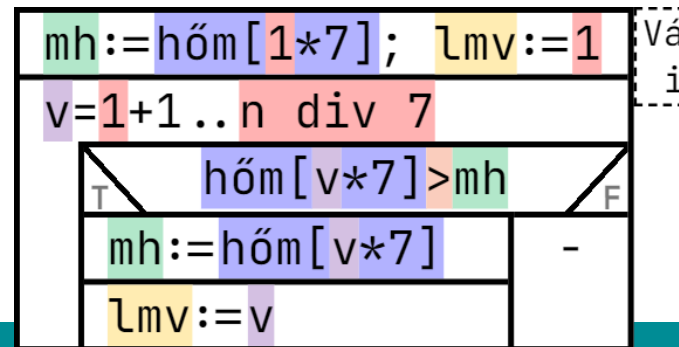
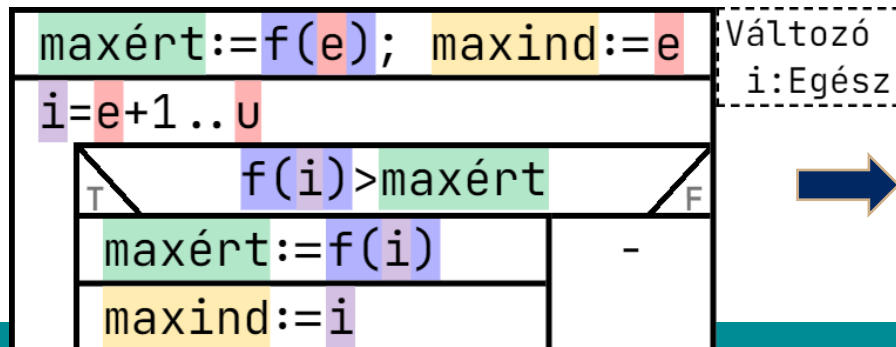
Be: $n \in \mathbb{N}$, $\text{hőm} \in \mathbb{R}[1..n]$

Ki: $\text{lmv} \in \mathbb{N}$, $\text{mh} \in \mathbb{R}$

Ef: $n \geq 7$

Uf: $(\text{lmv}, \text{mh}) =$
 $\text{MAX}(v = 1..n \text{ div } 7, \text{hőm}[v*7])$

$\text{maxind}, \text{maxért}$	\sim	lmv, mh
i	\sim	v
$e..u$	\sim	$1..n \text{ div } 7$
$f(i)$	\sim	$\text{hőm}[v*7]$



i	hőm
1	23,3
2	20,1
3	
4	
5	
6	
e 7	23,3
8	
9	
10	
11	
12	
13	
e+1 14	12,7
15	
16	
17	
18	
19	
20	
e+2 21	25,6
22	
23	

Összefoglalás



Programozási minták

1. Összegzés
2. Megszámolás
3. Maximumkiválasztás
 - a. Minimumkiválasztás
4. Feltételes maximumkeresés
5. Keresés
6. Eldöntés
 - a. Mind eldöntés
7. Kiválasztás
8. Másolás
9. Kiválogatás
 - a. Értékek