

8. táblás gyakorlat feladatai

Egy szekvenciális inputfájl vadászok eredményeit (vadász neve, vadászat dátuma, zsákmány fajtája, zsákmány súlya) tárolja a vadászok neve, azon belül a vadászat dátuma szerint rendezetten. Igaz-e, hogy minden vadász valamelyik vadászatán lőtt medvét, és hány olyan vadász van, aki minden vadászatán lőtt nyulat és a zsákmányainak összsúlya meghaladja a 250 kilogrammot?

Minta bemenet:

Élesszemű	2020.01.12	nyúl	2	Minden vadászaton lőtt
Élesszemű	2020.01.12	fácán	3	nyulat, zsákmányainak
Élesszemű	2020.01.22	medve	250	összsúlya több, mint 250,
Élesszemű	2020.01.22	nyúl	4	lőtt medvét.
Fürgelábú	2020.01.15	szarvas	260	Nem minden vadászatán
Fürgelábú	2020.02.17	nyúl	3	lőtt nyulat, nem lőtt medvét.
Szerencsés	2020.01.01	nyúl	16	Minden vadászaton lőtt
Szerencsés	2020.02.19	medve	200	nyulat, lőtt medvét, zsákmányainak
Szerencsés	2020.02.19	nyúl	10	összsúlya kevesebb, mint 250
...				

Specifikáció:

$A = (x:infile(Trófea), l: \mathbb{L}, c: \mathbb{N})$

Trófea = rec(név:S, dátum:S, fajta:S, súly:N)

$Ef = (x=x_0 \wedge x \neq (név,dátum))$

Ötlet:

Soroljuk fel, hogy az egyes vadászok lőttek-e medvét, lőttek-e minden vadászatukon nyulat, és mennyi a zsákmányaiknak az összsúlya.

Új specifikáció:

$A = (t:enor(Vadász), l: \mathbb{L}, c: \mathbb{N})$

Vadász = rec(medve: \mathbb{L} , nyúl: \mathbb{L} , összsúly: \mathbb{N})

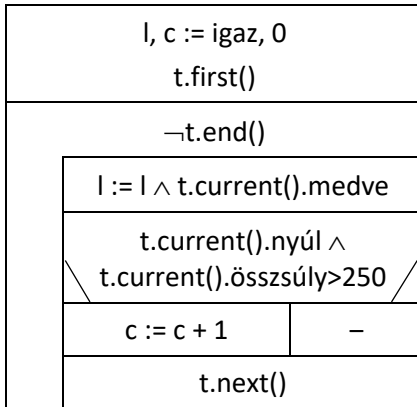
$Ef = (t=t_0)$

$Uf = (l = \bigwedge_{e \in t_0} e.medve \wedge c = \sum_{e \in t_0} 1)$
 $e.nyúl \wedge e.összsúly > 250$

Összeesélés és Számlálás közös felsorolón

t:enor(E)	~ t:enor(Vadász)
f(e)	~ e.medve
H, +, 0	~ \mathbb{L} , \wedge , igaz
felt(e)	~ $e.nyúl \wedge e.összsúly > 250$

Algoritmus:



Felsoroló:

Ötlet: A t:enor(Vadász) felsoroló megvalósításához szükség lesz egy olyan y:enor(Összesítés) felsorolóra, amely minden vadásznak az egyes vadászatokon elért összesített eredményeit sorolja fe a vadász neve és a vadász dátuma szerint növekedően rendezetten: lőtt-e medvét, lőtt-e nyulat, és mennyi volt ott a zsákmányai súlyának összege.

t:enor(Vadász)

Vadász = rec(medve:ℒ, nyúl:ℒ, összsúly:ℕ)

Vadász*	first()	next()	current() : Vadász	end() : ℒ
y : enor(Összesítés) akt : Vadász vége : ℒ	y.first() next()	lásd külön	return akt	return vége

Összesítés = rec(név: S, dátum:S, medve:ℒ, nyúl:ℒ, össz:ℕ)

next() művelet

$A = (y:enor(Összesítés), akt:Vadász, vége:ℒ)$

$Ef = (y = y' \wedge (y \text{ folyamatban van} \vee y \text{ befejeződött}) \wedge y \neq_{név})$

$Uf = (vége = y'.end() \wedge (\neg vége \rightarrow (akt.medve, y) = \bigvee_{e \in (y'.current(), y')}^{e.név=y'.current().név} e.medve$
 $\wedge (akt.nyúl, y) = \bigwedge_{e \in (y'.current(), y')}^{e.név=y'.current().név} e.nyúl$
 $\wedge (akt.összsúly, y) = \sum_{e \in (y'.current(), y')}^{e.név=y'.current().név} e.össz))$

Három összegzés (közös felsorolóval)

t:enor(E) ~ y:enor(Összesítés)
 first() nélkül
 amíg: y.current().név=akt.név

f(e) ~ (y.current().medve,
 y.current().nyúl,
 y.current().össz)

s ~ akt

H, +, 0 ~ (ℒ×ℒ×ℕ), (∨, ∧, +), (hamis, igaz, 0)

vége := y.end()	
¬vége	
név, akt.medve, akt.nyúl, akt.összsúly := y.current().név, hamis, igaz, 0	-
¬y.end() ∧ y.current().név=név	
akt.medve := akt.medve ∨ y.current().medve	
akt.nyúl := akt.nyúl ∧ y.current().nyúl	
akt.összsúly := akt.összsúly + y.current().össz	
y.next()	

Felsoroló:

y:enor(Összesítés) Összesítés = rec(név: S, dátum:S, medve:L, nyúl:L, össz:N)

Összesítés*	first()	next()	current() : Összesítés	end() : L
x : infile(Trófea) dx : Trófea sx : Status akt : Összesítés vége : L	sx,dx,x:read next()	lásd külön	return akt	return vége

Trófea = rec(név:S, dátum:S, fajta:S, súly:N)

next() művelet

A = (x:infile(Trófea), dx:Trófea, sx:Status, akt:Összesítés, vége:L)

Ef = (x = x' ∧ x'^(név,dátum) ∧ dx = dx' ∧ sx = sx')

Uf = (vége = (sx'=abnorm) ∧ (¬vége → akt.név=dx'.név ∧ akt.dátum=dx'.dátum ∧

dx,név=akt.név ∧ dx,dátum=akt.dátum

∧ (akt.medve, (sx,dx,x)) = $\bigvee_{dx \in (dx',x')}$ dx.fajta="medve" ∧

dx,név=akt.név ∧ dx,dátum=akt.dátum

∧ (akt.nyúl, (sx,dx,x)) = $\bigvee_{dx \in (dx',x')}$ dx.fajta="nyúl" ∧

dx,név=akt.név ∧ dx,dátum=akt.dátum

∧ (akt.össz, (sx,dx,x)) = $\sum_{dx \in (dx',x')}$ dx.súly))

Három összegzés (két vagyolás és egy összeadás)

közös felsorolóval

t:enor(E)	~ x:infile(Trófea) sx,dx,x:read, first() nélkül, amíg: dx.név=akt.név ∧ dx.dátum=akt.dátum
f(e)	~ (dx.fajta="medve", dx.fajta="nyúl", dx.súly)
s	~ akt.medve, akt.nyúl, akt.össz
H, +, 0	~ (L×L×N), (∨,∨,+), (hamis,hamis,0)

vége := sx=abnorm	
¬vége	
akt := (dx.név, dx.dátum, hamis, hamis, 0)	-
$sx=norm \wedge dx.név=akt.név$ $\wedge dx.dátum=akt.dátum$	
akt.medve := akt.medve \vee dx.fajta="medve"	
akt.nyúl := akt.nyúl \vee dx.fajta="nyúl"	
akt.össz := akt.össz + dx.súly	
sx, dx, x:read	

Mátrixok feldolgozása:

Nevezetes bejárások (felsorolók)

A mátrixok feldolgozhatók sor-, vagy oszlop folytonosan. Ezt a két bejárást tekinthetjük nevezetes felsorolásnak.

Legyen A egy $n \times m$ -es valós mátrix (n sora és m oszlopa van). Sorfolytonos felsorolója indexpárokat állít elő, amelyekkel a mátrix elemek elérhetők:

Sorfolytonos felsoroló:

t:enor($\mathbb{N} \times \mathbb{N}$) sorfolytonos felsorolása az indexpároknak

($\mathbb{N} \times \mathbb{N}$)*	first()	next()	current() : ($\mathbb{N} \times \mathbb{N}$)	end() : \mathbb{L}
i: \mathbb{N} j: \mathbb{N} n, m : \mathbb{N}	i, j := 1, 1	if j < m then j := j + 1 else i := i + 1 j := 1 endif	return (i, j)	return (i > n)

Feladat: Határozzuk a mátrix (egyik) legnagyobb elemét:

A = (X: $\mathbb{R}^{n \times m}$, ind, jnd: \mathbb{N})

Ef = (X=X' \wedge n>0 \wedge m>0)

Uf = (ind, jnd = $\mathbf{MAX}_{i=1, j=1}^{n, m} X'[i, j]$)

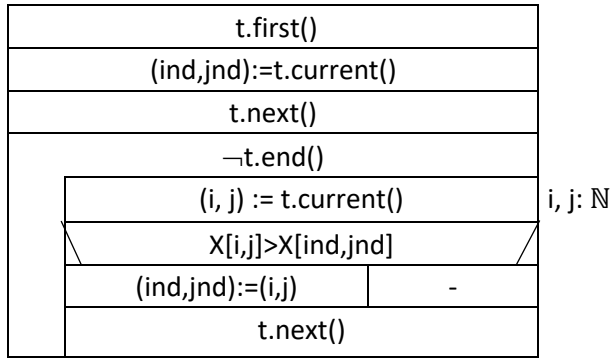
Maximum kiválasztás

t:enor(E) ~ indexpárok sorfolytonos felsorolója

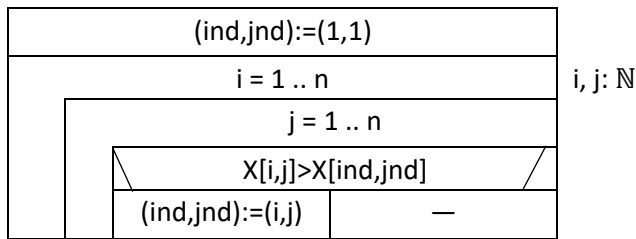
f(e) ~ X[i, j]

H, > ~ $\mathbb{R}, >$

Algoritmus:



Szokásosabb formában:



Feladat: Mennyi egy négyzetes mátrix sakktabla szerinti fehér mezőire eső elemeinek az összege?

I. megoldás: nevezetes felsorolóval

Specifikáció:

$$A = (a: \mathbb{R}^{n \times n}, s: \mathbb{R})$$

$$Ef = (a = a')$$

$$Uf = (Ef \wedge s = \sum_{\substack{i,j=1,1 \\ 2 \mid i+j}}^{n,n} a[i,j])$$

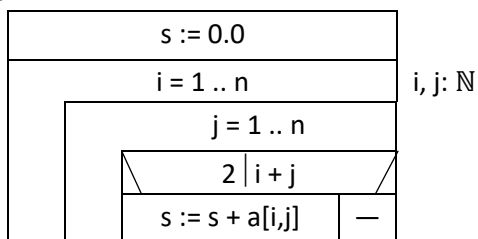
Összegzés

$$t: \text{enor}(\mathbb{R}) \quad \sim \text{indexpárok klasszikus felsorolója}$$

$$f(e) \quad \sim a[i,j] \text{ ha } 2 \mid i+j$$

$$H, +, 0 \quad \sim \mathbb{R}, +, 0.0$$

Algoritmus:



Mátrix, egyedi felsoroló készítés

Specifikáció:

$$A = (a: \mathbb{R}^{n \times n}, t: \text{enor}(\mathbb{N} \times \mathbb{N}), s: \mathbb{R})$$

$$Ef = (a = a' \wedge t = t')$$

$$Uf = (s = \sum_{(i,j) \in t'} a'[i,j])$$

Összegzés

t:enor(E) ~ indexpárok sakktáblás felsorolója
 f(e) ~ a[i,j]
 H, +, 0 ~ $\mathbb{R}, +, 0.0$

Algoritmus:

s := 0.0	
t.first()	
¬t.end()	
(i, j) := t.current()	i, j: \mathbb{N}
s := s + a[i,j]	
t.next()	

Felsoroló

t:enor($\mathbb{N} \times \mathbb{N}$) sakktáblaszerű felsorolása az indexpároknak

$(\mathbb{N} \times \mathbb{N})^*$	first()	next()	current() : $\mathbb{N} \times \mathbb{N}$	end() : \mathbb{L}
n : \mathbb{N} i, j : \mathbb{N}	i, j := 1, 1	if j ≤ n-2 then j:=j+2 else i := i+1 if i páros then j := 2 else j := 1 endif endif	return (i, j)	return (i>n)